Technische Universität München
TUM School of Engineering and Design

# Optimal Control of Flexible Multibody Systems using Adjoint Sensitivities

**Daniel Lichtenecker**

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**

genehmigten Dissertation.

**Vorsitz:** Prof. Dr.-Ing. Florian Holzapfel

**Prüfende der Dissertation:**

1. TUM-IAS Fellow Dr. Karin Nachbagauer
2. Prof. dr.ir. Daniel J. Rixen
3. Prof. Ahmed A. Shabana, Ph.D.

Die Dissertation wurde am 01.07.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 26.09.2024 angenommen.

## Acknowledgement

This thesis results from my research at the Chair of Applied Mechanics at the Technical University of Munich. The work has been supported by the Hans Fischer Fellowship Program of the Institute for Advanced Study (IAS) of the Technical University of Munich. I am deeply grateful for the opportunity to be a TUM-IAS funded doctoral candidate in the IAS Focus Group "Multibody Systems". This research group consists of excellent researchers and inspiring personalities, including Prof. Michel Géradin, Prof. Karin Nachbagauer, and Dr. Valentin Sonneville.

First and foremost, I would like to express my sincere gratitude to my doctoral advisor, Karin Nachbagauer, for her invaluable advice and continuous support throughout my entire scientific career. Dear Karin, your approach to teaching and working has fascinated me since the first day I attended your applied mathematics class. With your passion and enthusiasm for mathematics, you have inspired me to start a PhD program. You have introduced me to the world of science, and beyond that, I have learned a lot from you on a personal level. Thank you for numerous inspiring discussions on our common research interest, for your invaluable feedback on my papers and this thesis, for giving me the freedom to shape my scientific profile and your support of every idea I had during my PhD studies, and for the conversations that went far well beyond university life.

I would like to express my gratitude to the members of the examination committee. In particular, I am very thankful that Prof. Florian Holzapfel (Technical University of Munich) has agreed to chair the exam. Additionally, I am deeply grateful to Prof. Daniel Rixen (Technical University of Munich) and Prof. Ahmed A. Shabana (University of Illinois at Chicago) for their interest in my research, valuable feedback, and willingness to review my work. In addition to the members of the examination committee, I would like to express my gratitude to Prof. Peter Betsch (Karlsruhe Institute of Technology) for his valuable feedback on my thesis.

Special thanks to Prof. Masayoshi Tomizuka for hosting my research stay at the Mechanical Systems Control Lab of the University of California, Berkeley. The unique opportunity to work as a visiting scholar under his supervision was undoubtedly one of the highlights of my PhD studies. The research stay was a professional enrichment and an invaluable personal experience. I am grateful for the warm hospitality of Prof. Masayoshi Tomizuka and all the members of the Mechanical Systems Control Lab team. Their support has greatly enriched my time in Berkeley and allowed me to explore the academic and cultural environment.

I would like to thank all the members of the Chair of Applied Mechanics with whom I had the pleasure of working during my time at the university. Special thanks go to Prof. Daniel Rixen for the professional conversations and the freedom he gives to his PhD students. His openness to explore different areas of research is truly inspiring, and the positive working atmosphere at the Chair is mainly due to his cooperative working style. I have had the privilege of working with many exceptionally talented and motivated colleagues at the Chair.

I appreciate our professional and personal conversations and friendships that extended far beyond our time at the university. That is what matters at the end of the day. I am proud to be a part of the Chair, and this time will stay in my best memory.

Finally, but most importantly, I would like to express my deepest gratitude to my family, particularly my mother, Gertraud, and my wonderful wife, Katharina, for their never-ending support. Dear Katharina, you are my sunshine, my inspiration, my entire world. Without your tremendous understanding and encouragement, it would have been impossible for me to complete this thesis. You were the first to believe in my plan to leave our life in Austria behind and move to Munich so that I could pursue a career in research. You encouraged us to take the unique opportunity to move to Munich despite your previous job, and for that, I will be forever grateful. Thank you for managing our social life, for doing everything in your power to make me feel comfortable, for giving me the freedom to develop my scientific and professional career, for understanding all the late nights and early mornings, and for traveling around the world with me. But most of all, thank you for being my best friend. I owe you everything. The future is ours.

Ismaning, May 14, 2024 *Daniel*

## Abstract

This thesis covers the topic of adjoint-based sensitivity analysis and its application to optimal control problems of flexible multibody systems. The new scientific contribution in this research area consists of an analytical and, thus, accurate approach for the efficient computation of first-order sensitivities, which is crucial for solving complex and large-scale optimization problems.

The adjoint method is of major relevance in this thesis for developing efficient and accurate computational methods in sensitivity analysis. As the theoretical foundation relevant to this thesis, the general concept of the adjoint method is reviewed in depth. Additionally, the theoretical aspects of solving optimal control problems are provided. Therein, the necessary first-order optimality conditions of unbounded and bounded optimization problems are discussed, and shooting and collocation methods are addressed as essential representatives of gradient-based solution methods for the optimization problem. Furthermore relevant to this thesis are the theoretical aspects of flexible multibody systems, focusing on the absolute nodal coordinate formulation. The absolute nodal coordinate formulation is used in this thesis to describe the dynamical behavior of deformable structures.

The optimal control problem of such flexible multibody systems under equality and inequality constraints is solved by iteratively minimizing the cost functional in this thesis. The iterative solution procedure of gradient-based optimization methods requires the computation of sensitivities concerning the cost functional and the constraints in each iteration. The computation of sensitivities is of fundamental importance in optimization since the sensitivities are used to compute the search direction to minimize the cost functional while simultaneously satisfying the equality and inequality constraints of the optimization problem. An efficient and accurate approach to sensitivity analysis is particularly advantageous for large-scale optimization problems. To address the requirements of efficiency and accuracy in sensitivity analysis, the general concept of the adjoint theory is exploited in this thesis to derive approaches to sensitivity analysis concerning various optimal control problems.

In this thesis, optimal control problems are treated using direct optimization methods. Optimal controls are computed for rest-to-rest problems of mechanical systems using both rigid and flexible bodies to model the mechanical components. Initial and final state constraints of the mechanical system are specified, and equality and inequality constraints are considered during the motion of the mechanical system. For example, the time-optimal control problem of a flexible two-arm robot and the energy optimal control problem of a nonlinear spring pendulum are analyzed. These examples demonstrate the theoretical concepts presented in this thesis and highlight the computational benefits.

In summary, this thesis contributes to optimal control of flexible multibody systems. The adjoint-based sensitivities presented in this thesis significantly reduce the computational time to find an optimal solution in sophisticated optimization problems, such as the combined optimal control and structural optimization problem within a single optimization framework. The results of this thesis contribute to the ongoing development of methods for optimal control problems of flexible multibody systems and enable new possibilities for future applications, e.g., in soft robotics.

## Zusammenfassung

Diese Dissertation befasst sich mit dem Thema der adjungierten Sensitivitätsanalyse und ihrer Anwendung auf Optimalsteuerungsprobleme flexibler Mehrkörpersysteme. Der neue wissenschaftliche Beitrag zu diesem Forschungsgebiet besteht in einem analytischen und damit genauen Ansatz zur effizienten Berechnung von Sensitivitäten erster Ordnung, die für die Lösung komplexer Optimierungsprobleme entscheidend sind.

Die adjungierte Methode ist in dieser Dissertation von zentraler Bedeutung für die Entwicklung von effizienten und genauen Berechnungsmethoden zur Sensitivitätsanalyse. Als theoretische Grundlage für diese Dissertation wird das allgemeine Konzept der adjungierten Methode vorgestellt. Darüber hinaus werden die theoretischen Aspekte zur Lösung von Optimalsteuerungsproblemen dargestellt. Darin werden die notwendigen Optimalitätsbedingungen erster Ordnung sowohl für unbeschränkte als auch für beschränkte Optimierungsprobleme untersucht, und Schießverfahren und Kollokationsverfahren als wichtige Vertreter gradientenbasierter Lösungsverfahren für Optimalsteuerungsprobleme werden diskutiert. Weiters sind die theoretischen Grundlagen der Modellierung flexibler Mehrkörpersysteme für diese Dissertation relevant, wobei der Schwerpunkt auf der sogenannten absolute nodal coordinate formulation liegt. Dieser Modellierungsansatz wird zur Beschreibung des dynamischen Verhaltens von deformierbaren Strukturen verwendet.

Optimalsteuerungsprobleme solcher flexibler Mehrkörpersysteme unter Gleichheits- und Ungleichungsbeschränkungen werden in dieser Dissertation durch eine iterative Minimierung des Kostenfunktionals gelöst. Der iterative Lösungsprozess gradientenbasierter Optimierungsverfahren erfordert die Berechnung der Sensitivitäten des Kostenfunktionals und der Beschränkungen in jedem Iterationsschritt. Die Berechnung von Sensitivitäten ist in der Optimierung von fundamentaler Bedeutung, da auf Basis der Sensitivitäten die Suchrichtung zur Minimierung des Kostenfunktionals unter Einhaltung der Gleichheits- und Ungleichungsbeschränkungen des Optimierungsproblems berechnet wird. Ein effizienter und analytisch hergeleiteter Ansatz zur Sensitivitätsanalyse ist insbesondere bei großen Optimierungsproblemen von entscheidender Bedeutung. Um den Anforderungen an Effizienz und Genauigkeit der Sensitivitätsanalyse gerecht zu werden, wird in dieser Dissertation das allgemeine Konzept der adjungierten Methode verwendet, um Ansätze zur Sensitivitätsanalyse für unterschiedlichste Optimalsteuerungsprobleme zu formulieren.

In dieser Dissertation werden Optimalsteuerungsprobleme mit direkten Optimierungsverfahren gelöst. Optimale Steuerungen werden für Punkt-zu-Punkt-Probleme mechanischer Systeme berechnet, wobei sowohl starre als auch flexible Körper zur Modellierung der mechanischen Bauteile verwendet werden. Anfangs- und Endzustandsbedingungen des mechanischen Systems werden spezifiziert und Gleichheits- und Ungleichungsbeschränkungen während der Bewegung des mechanischen Systems berücksichtigt. So wird zum Beispiel die zeitoptimale Bewegung eines flexiblen zweiarmigen Roboters untersucht und das energieoptimale Steuerungsproblem eines nichtlinearen Federpendels analysiert. Diese Beispiele veranschaulichen unter anderem die theoretischen Konzepte der wissenschaftlichen Beiträge dieser Dissertation und zeigen deren Vorteile in der Anwendung auf.

Zusammenfassend leistet diese Dissertation einen bedeutenden Beitrag zur optimalen Steuerung flexibler Mehrkörpersysteme. Die vorgestellten Berechnungsmethoden mittels adjungierter Sensitivitätsanalyse reduzieren die benötigte Rechenzeit für eine optimale Lösung des Optimierungsproblems signifikant. Dies ermöglicht es, komplexe Optimierungsprobleme wie das kombinierte Steuerungs- und Strukturoptimierungsproblem in einer gemeinsamen Optimierung effizient zu lösen. Die Ergebnisse dieser Dissertation tragen zur Weiterentwicklung von Methoden zur optimalen Steuerung flexibler Mehrkörpersysteme bei und eröffnen neue Möglichkeiten für zukünftige Anwendungen, z. B. im Bereich der Soft-Robotik.

# Contents


**1 Introduction** **1**
- 1.1 Context and Motivation . . . 2
- 1.2 Objective . . . 3
- 1.3 Outline . . . 4

**2 General Concept of the Adjoint Theory** **7**
- 2.1 Historical Perspective . . . 7
- 2.2 Linear Duality . . . 8
  - 2.2.1 Static Problems . . . 8
  - 2.2.2 Dynamic Problems . . . 9
- 2.3 The Adjoint Method for Sensitivity Analysis . . . 10
  - 2.3.1 Static Problems . . . 10
  - 2.3.2 Dynamic Problems . . . 13
- 2.4 Computational Aspects . . . 15
  - 2.4.1 Duality of Sensitivities . . . 16
  - 2.4.2 Backward Time Integration . . . 16
  - 2.4.3 Memory Efficient Implementation . . . 18
  - 2.4.4 Procedure of Adjoint Sensitivities . . . 18

**3 Fundamentals of Optimal Control** **19**
- 3.1 Problem Formulation . . . 21
- 3.2 Indirect Methods . . . 21
  - 3.2.1 Unbounded Optimal Control Problems . . . 21
  - 3.2.2 Bounded Optimal Control Problems . . . 27
  - 3.2.3 Solution Approaches . . . 31
- 3.3 Direct Methods . . . 33
  - 3.3.1 Direct Single Shooting . . . 34
  - 3.3.2 Direct Multiple Shooting . . . 36
  - 3.3.3 Direct Collocation . . . 37
  - 3.3.4 Nonlinear Programming Problem Formulation . . . 40
  - 3.3.5 Optimality Conditions . . . 41
  - 3.3.6 A Basic Sequential Quadratic Programming Method . . . 42
  - 3.3.7 A Basic Interior Point Method . . . 43
- 3.4 Sensitivity Analysis . . . 44
  - 3.4.1 Finite-Difference Method . . . 45
  - 3.4.2 Direct Differentiation Method . . . 46
  - 3.4.3 Adjoint Variable Method . . . 46

**4 Large Deformation Problems in Flexible Multibody Systems** **49**
- 4.1 Overview . . . 49
- 4.2 Absolute Nodal Coordinate Formulation . . . 50

4.2.1 Kinematics . . . . . 51
4.2.2 Equations of Motion . . . . . 53
4.3 Element Assembly . . . . . 58

**5 Scientific Contributions** **61**
5.1 Outline of Publications . . . . . 61
5.2 Summary of Publications . . . . . 61
5.2.1 Publication I . . . . . 62
5.2.2 Publication II . . . . . 63
5.2.3 Publication III . . . . . 64

**6 Closure** **67**
6.1 Conclusion and Discussion . . . . . 67
6.2 Outlook . . . . . 68

**A Notation** **71**
A.1 Vectors and Matrices . . . . . 71
A.2 Nabla Operator . . . . . 71

**B Publications** **73**
B.1 Publication I . . . . . 73
B.2 Publication II . . . . . 96
B.3 Publication III . . . . . 109

**Bibliography** **139**

Chapter 1

# Introduction

Enhancing mechanical systems is crucial to meet the increasing demands of modern industry and future technological progress. The rapid development in mechanical engineering, such as in robotics, aerospace, or vehicle dynamics, simultaneously requires efficient methods to optimally perform the desired tasks of a mechanical system. Engineering and scientific communities have faced increasing challenges in developing sophisticated simulation models to solve complex and challenging problems in mechanical engineering. Therefore, there is a growing need for robust and efficient algorithms in research and industry to meet current and future challenges.

Faced with these challenges, the optimization of mechanical systems plays an essential role in addressing complex and challenging problems across various scientific disciplines. For example, a permanent issue in mechanical engineering is the reduction of energy consumption to enable the development of sustainable and efficient solutions. A common approach to reduce energy consumption is using a lightweight design resulting from a structural optimization. Material savings reduce the actuator requirements to drive a mechanical system. In addition to structural optimization, the reduction of energy can be addressed directly by optimizing the actuation for an energy optimal control of a given mechanical system. Typically, structural optimization and optimal control are considered independently in a sequential process. However, coupling both optimization strategies is promising to obtain the best possible mechanical structure concerning an optimal control problem (OCP).

To formulate such optimization strategies, the use of mathematical models is required to predict the physical behavior of the given mechanical system. For example, lightweight structures and components made of soft material can undergo high deformation during motion. Therefore, proper formulations are required to capture the flexibility of the components accurately. Depending on the properties of the mechanical system, different formulations of flexible multibody dynamics are used for modeling. In addition to modeling, computing the system response of the mathematical model plays a critical role in the efficiency and accuracy of the solution. Due to the complexity of the models, numerical simulation methods are usually used to compute the evolution of the system response.

The performance of mechanical systems can be improved by finding an optimal design of mechanical components and/or an optimal control to drive the mechanical system toward its desired goal. To this end, expertise in modeling, simulation, and numerical optimization is required. This thesis mainly focuses on the optimal control of deformable mechanical systems. However, the scientific contributions of this thesis can also be used in optimal design, e.g., to address the challenging optimization problem of coupling structural optimization and optimal control.

## 1.1 Context and Motivation

Optimal control of flexible components such as soft robotic systems is of major concern, especially when using lightweight soft materials [141]. Designing the shape of soft robots and the computation of a high-fidelity control is an emerging research and development area [143]. Such advanced robotic systems are becoming important due to their inherent compliance, strong adaptability, and ability to effectively operate in unstructured environments like medical or bioinspired applications [42]. Their compliance and flexibility often lead to highly nonlinear dynamics, which complicates the design and control of such robots. Accurate formulations are essential to account for the effects of deformations and stresses on the motion control of soft robots. Capturing the effects of deformation can be considered by the concepts of flexible multibody dynamics. Mechanical models formulated with flexible bodies are usually underactuated systems, and the OCP becomes more complicated compared to fully actuated systems [135].

In general, optimal control aims to determine time-dependent control functions to drive a mechanical system toward its desired goal. In a mathematical context, optimal control regards optimizing the input of a mechanical system to minimize a cost functional while satisfying physical constraints, e.g., the rest-to-rest motion of a soft robot. Therein, the dynamical behavior of the underlying mechanical system is described by using differential equations. To address the complexity of OCPs, various numerical methods have been developed to meet the specific requirements of different OCP formulations. However, the focus of this thesis is to use so-called direct methods for solving different OCPs. Direct methods formulate a nonlinear programming (NLP) problem that can be addressed using gradient-based optimization algorithms, e.g., the sequential quadratic programming (SQP) method or the interior point (IP) method. These algorithms use first and second-order gradients to determine a feasible solution to the optimization problem. The computation of gradients is usually referred to as sensitivity analysis and is essential in the iterative solution process of optimization problems. Methods for computing sensitivities are generally divided into numerical and analytical approaches.

In Gufler et al. [74], a comprehensive literature review presents various gradient-based optimization methods in the design optimization of flexible multibody systems. The review paper discusses the objectives in the design optimization of flexible multibody dynamics and reviews concepts and applications in this research area. More than 160 publications in the bibliography provide a comprehensive overview of optimization algorithms, types and formulations, and sensitivity analyses. Due to the increasing complexity of optimization problems, scientists have paid much attention to developing sophisticated analytical approaches for sensitivity analysis. Analytical sensitivity analysis includes the direct differentiation method and the adjoint variable method, two principal approaches for computing sensitivities. The direct differentiation method [81] facilitates implementation via straightforward differentiation of the differential equations, cost functional, and (in)equality constraints with respect to the optimization variables. Thus, the direct differentiation method yields a high computational effort for large-scale optimization problems. A computationally efficient alternative to the direct differentiation method is given by the adjoint variable method. This method avoids computing the differentiation of the differential equations with respect to the optimization variables by introducing adjoint variables. Therefore, the adjoint variable method is best suited for large-scale optimization problems.

The use of the adjoint variable method for computing gradients in optimization problems has a long history in optimal control theory [103]. Pioneering contributions to the adjoint variable method are given by Kelley [91] and Bryson and Ho [30]. The relevance of adjoint-based gradients has increased significantly in various engineering disciplines such as design

optimization [114] and multibody dynamics [38, 51], especially as these applications become more complex. Therefore, the efficient computation of gradients has become crucial in sensitivity analysis. For example, applications of sensitivity analysis can be found in a broad spectrum of fields, including optimization, parameter identification, data assimilation, optimal control, uncertainty analysis, and experimental design. In the emerging research area of neural networks, the ability of solvers to effectively compute gradients for high-dimensional machine learning models is particularly advantageous [128]. Johnston and Patel [88] have found that adjoint methods are essential for the efficient gradient computation of functionals in both control theory and machine learning. Recent publications address the adjoint method in multibody dynamics, e.g., in a feedback-feedforward control system that computes input signals and the corresponding trajectories [106]. Schneider and Betsch [134] propose a novel approach that preserves the variational structure of problems through the strategic choice of a mechanical Hamiltonian and integration of constraints. Furthermore, Held and Seifried [84] presented an approach to demonstrate the use of the adjoint variable method for multibody systems initially modeled by differential-algebraic equations but solved in minimal coordinates using a QR decomposition.

In this thesis, the general concept of the adjoint variable method is used to compute first-order sensitivities of OCPs since the adjoint method is probably the most efficient method to compute sensitivities [37, 119]. Thus, the scientific contributions of this thesis contribute to the efficient and accurate computation of sensitivities and, therefore, play a significant role in future challenges of OCPs of flexible multibody systems.

## 1.2 Objective

The main goal of this thesis is to develop efficient and accurate approaches to compute first-order sensitivities in OCPs of flexible multibody systems. The thesis contributes to the following ongoing research areas in optimal control:

- **Time-optimal control problems**
  An important and challenging objective in optimal control is to identify the minimum required time to manipulate a mechanical system toward its desired goal. This problem becomes even more challenging when the mechanical system consists of flexible components. It is crucial to accurately capture the deformations and vibrations of these flexible structures by proper flexible multibody formulations to ensure that the optimization constraints during the manipulation and at the final time are satisfied. The compliance of the flexible structures and physical limitations of the actuator during the motion restrict the shortest possible time. The challenging problem requires advanced approaches to efficiently integrate the dynamic behavior of flexible components into the time-optimal control problem.

- **Sensitivity analysis of constraints in dynamics**
  Efficient and accurate computation of sensitivities is crucial in the iterative process of solving complex and large-scale optimization problems. In the past, considerable effort has been dedicated to developing approaches for sensitivity analysis of the cost functional. However, it is equally important to develop efficient methods for analyzing the sensitivities of constraints within optimization problems. Efficient and accurate computation of constraint sensitivities is essential to address challenging optimization problems, such as the simultaneous consideration of an optimal control and design problem in soft robotics.

Recent research activities in both research areas show a growing interest in integrating flexible multibody systems in OCPs. The interdisciplinary nature of optimal control concerning flexible components is challenging and requires the development of robust methods to improve the convergence to a local minimum of the optimization problem.

## 1.3 Outline

This thesis is organized into six chapters and two appendices. Following this introduction chapter, Chapters 2–4 provide basic and fundamental theoretical principles relevant to the understanding of the scientific contributions of this thesis. Chapter 5 summarizes the underlying publications of this thesis, and reprints of the publications are given in Appendix B. The content of the subsequent chapters and appendices is outlined as follows:

- **Chapter 2** establishes the mathematical foundation of the adjoint theory relevant to this thesis. Therein, a brief overview of the historical perspective of the adjoint theory is given, and the adjoint theory is introduced in an abstract context of linear algebra. In particular, the relation of dual and primal formulations in statics and dynamics is discussed. Furthermore, the terminology of duality is discussed in the sensitivity analysis of static and dynamic systems. Computational aspects of adjoint-based sensitivities for dynamical systems conclude this chapter.

- **Chapter 3** discusses the theoretical aspects of OCPs. An overview of numerical solution methods is given, focusing on direct and indirect methods. In particular, the necessary first-order optimality conditions of unbounded and bounded problems are derived in detail for indirect methods. Furthermore, shooting and collocation methods are reviewed as important representatives of direct methods. Using direct methods yields an NLP problem, which is usually solved by gradient-based optimization methods, as discussed in this chapter. Finally, the chapter concludes with a discussion of commonly used approaches in sensitivity analysis.

- **Chapter 4** provides an introduction to flexible multibody systems with an emphasis on the absolute nodal coordinate formulation. Therein, the kinematics and kinetics of a well-established beam element are discussed, which yields the equations of motion for a beam element. In addition, a constrained flexible multibody system is assembled using the equations of motion from a single beam element.

- **Chapter 5** summarizes the scientific contributions of this thesis based on three peer-reviewed publications. The scientific contributions focus on deriving advanced formulations in sensitivity analysis using the adjoint variable method, highlighting their efficiency and applicability to large-scale optimization problems.

- **Chapter 6** gives a brief conclusion and discussion of the main contributions of this thesis. Moreover, recommended future research directions are highlighted.

- **Appendix A** includes some relevant notations of linear algebra used in this thesis.

- **Appendix B** presents the peer-reviewed publications of this thesis as reprints.

As an illustrative example of the above outline, Fig. 1.1 visualizes the core chapters of this thesis within the framework of an OCP of a robotic system. This example aims to compute an optimal control to drive the robotic system modeled with two flexible bodies concerning a rest-to-rest maneuver. In this thesis, the modeling of flexible bodies relies on established

approaches widely used and tested within the multibody dynamics research community, as highlighted in Chapter 4. For an efficient computation the OCP, the methods discussed in Chapter 3 are used in the solution process along with adjoint-based sensitivities, which are part of the scientific contributions of this thesis. The efficient and accurate sensitivity analysis is based on the general concept of the adjoint theory as addressed in Chapter 2.

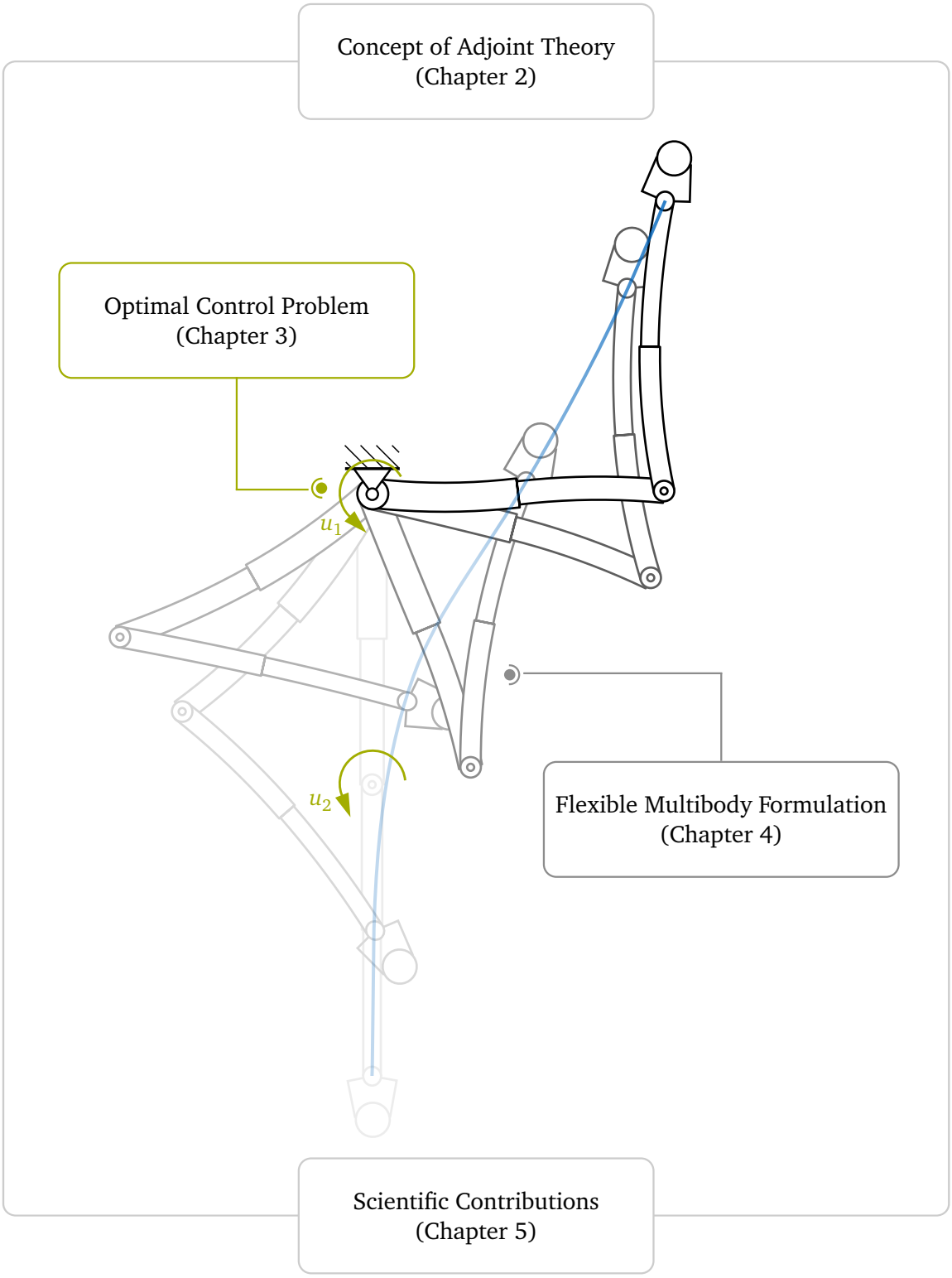



**Figure 1.1:** Visualization of the outline of this thesis based on the motion of a flexible robotic system

Chapter 2

# General Concept of the Adjoint Theory

This chapter establishes the mathematical foundation of the adjoint theory. The general concept of the adjoint theory is used in this thesis to develop adjoint-based sensitivities for accurate and efficient computation of first-order gradients in optimization problems.

The use of adjoint equations has a long history in computational science, engineering, and mathematics. A brief overview of the historical perspective of the adjoint theory is given in Section 2.1. The historical notes provide fundamental contributions in aerodynamics and optimal control, which are the foundation of current developments using the adjoint theory. In the general concept, the adjoint theory is based on the substitution of variables and utilizes the terminology of duality discussed in Section 2.2. Therein, the adjoint theory is introduced in an abstract context of linear algebra. The adjoint theory is first discussed in static problems for ease of understanding, followed by dynamic problems governed by first-order differential equations. An important application of the adjoint theory is the sensitivity analysis for dynamical systems, which is discussed in Section 2.3. Computational aspects of adjoint-based sensitivities for dynamical systems in Section 2.4 conclude this chapter.

The scope of this chapter is to provide sufficient details to understand the general concept of the adjoint theory. The adjoint theory is used in the scientific contributions of the author's publications in [99, 101, 102] to develop adjoint-based sensitivities of OCPs. It is not an aim to provide a complete treatment of the adjoint theory in optimal control. Instead, interested readers are referred to excellent textbooks, e.g., on general adjoint theory [108], on adjoint theory in optimal control [30, 91, 103]. The theoretical aspects of optimal control presented in the next sections mainly follow the work of these textbooks.

## 2.1 Historical Perspective

The origin of the adjoint theory can be traced back to the fundamental work of Joseph-Louis Lagrange, who introduced the Lagrange identity to define the adjoint operator. In the 20th century, the general concept of the adjoint theory has been established in a wide range of scientific fields. Researchers worked on pioneering theoretical developments using the adjoint theory within their area of research. In the field of optimal control, Kelley [91] and Bryson and Denham [28] independently developed a gradient-based method to treat OCPs using the adjoint theory. Lions [103] introduced the adjoint theory in OCPs for systems governed by partial differential equations. Another fundamental work of utilizing the adjoint theory in optimal control is the textbook of Bryson and Ho [30]. This book covers various optimization problems for dynamical systems and is a key reference for optimal control. Adjoint-based methods also have a long history in theoretical formulations of fluid mechanics. The adjoint theory was first used in this area by Pironneau [122] for drag minimization problems for Stokes flow. The pioneering work of Pironneau was later extended by Jameson [87] to perform aerodynamic shape optimization. Based on developments in optimal control and

fluid mechanics, researchers in the field of structural mechanics adopted the adjoint theory to develop new methods in structural optimization [3, 80]. Researchers identified that using adjoint methods has a particular potential in sensitivity analysis [4, 107]. Bestle and Eberhard [12] developed a sensitivity analysis approach for optimizing multibody systems governed by first-order differential equations. This approach has been extended by Bestle and Seybold [13] for systems governed by differential-algebraic equations (DAE). Further significant developments, including the adjoint theory in multibody dynamics, are provided in the papers [38, 51, 113]. The aforementioned fundamental contributions are the basis of current developments using the adjoint theory.

## 2.2 Linear Duality

The adjoint method can be interpreted as a special case of linear duality to compute an output [65]. The computation of an output can be viewed from the two perspectives of a *primal* or *dual* problem definition. In the general concept, the adjoint method is based on the substitution of variables and utilizes the terminology of duality [111]. This section introduces the adjoint method in an abstract context of linear algebra. The terminology of duality for static problems is discussed first for ease of understanding. The concept is then extended to dynamic problems governed by first-order differential equations.

### 2.2.1 Static Problems

Suppose that a computation procedure requires the numerical evaluation of the time-invariant matrix

$$\mathbf{Y} = \mathbf{F} + \mathbf{G}^{\mathrm{T}}\mathbf{B} \in \mathbb{R}^{N_{\mathrm{j}} \times N_{\mathrm{z}}}, \tag{2.1}$$

where the matrices $\mathbf{F} \in \mathbb{R}^{N_{\mathrm{j}} \times N_{\mathrm{z}}}$ and $\mathbf{G} \in \mathbb{R}^{N_{\mathrm{n}} \times N_{\mathrm{j}}}$ are given. The matrix $\mathbf{B} \in \mathbb{R}^{N_{\mathrm{n}} \times N_{\mathrm{z}}}$ is the unknown of the linear system

$$\mathbf{A}\mathbf{B} = \mathbf{C}, \tag{2.2}$$

in which the matrices $\mathbf{A} \in \mathbb{R}^{N_{\mathrm{n}} \times N_{\mathrm{n}}}$ and $\mathbf{C} \in \mathbb{R}^{N_{\mathrm{n}} \times N_{\mathrm{z}}}$ are given. The straightforward approach to compute $\mathbf{Y}$ is to first solve (2.2) for $\mathbf{B}$, and then use the solution of $\mathbf{B}$ in (2.1) to compute $\mathbf{Y}$. An alternative to this approach is to introduce a matrix $\mathbf{P} \in \mathbb{R}^{N_{\mathrm{n}} \times N_{\mathrm{j}}}$, the so-called *dual* or *adjoint* matrix including adjoint variables, and replace (2.1) with

$$\tilde{\mathbf{Y}} = \mathbf{F} + \mathbf{P}^{\mathrm{T}}\mathbf{C}, \tag{2.3}$$

where the adjoint matrix $\mathbf{P}$ is the unknown of the linear system

$$\mathbf{A}^{\mathrm{T}}\mathbf{P} = \mathbf{G}. \tag{2.4}$$

The equations in (2.3) and (2.4) are referred to the *dual* problem corresponding to the *primal* problem formulation in (2.1) and (2.2). The dual and primal problems are defined in such a way that the matrices $\tilde{\mathbf{Y}}$ and $\mathbf{Y}$ are equal. The equivalence of the primal and the dual problem can be shown by substituting (2.2) and (2.4) into (2.3), which results in

$$\tilde{\mathbf{Y}} = \mathbf{F} + \mathbf{P}^{\mathrm{T}}\mathbf{C} = \mathbf{F} + \mathbf{P}^{\mathrm{T}}\mathbf{A}\mathbf{B} = \mathbf{F} + \left(\mathbf{A}^{\mathrm{T}}\mathbf{P}\right)^{\mathrm{T}}\mathbf{B} = \mathbf{F} + \mathbf{G}^{\mathrm{T}}\mathbf{B} = \mathbf{Y}. \tag{2.5}$$

The equation in (2.5) states that $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ are equivalent if the primal system is solved for $\mathbf{B}$ and the dual system is solved for $\mathbf{P}$, respectively. Both perspectives require the solution of linear systems to compute the matrices $\mathbf{B}$ and $\mathbf{P}$. Note that these matrices have the same number of rows $N_\mathrm{n}$, but the number of columns $N_\mathrm{z}$ for $\mathbf{B}$ and $N_\mathrm{j}$ for $\mathbf{P}$ is different. Thus, one has to solve either $N_\mathrm{z}$ different primal computations for $\mathbf{B}$ or $N_\mathrm{j}$ different dual computations for $\mathbf{P}$. This fact implies the computational benefits of using the dual approach when $N_\mathrm{j} < N_\mathrm{z}$. Using the dual approach is especially efficient when $N_\mathrm{j} \ll N_\mathrm{z}$. Moreover, the number of columns $N_\mathrm{z}$ of the matrix $\mathbf{Y}$ determines the number of columns for $\mathbf{B}$, which is not the case for $\mathbf{P}$. This characteristic is the most powerful aspect of linear duality, where linear algebra is exploited to efficiently compute $\mathbf{Y}$ [111].

### 2.2.2 Dynamic Problems

Similar to the static case where all variables are time-invariant, the terminology of linear duality can be exploited for time-dependent problems. Suppose that a computation procedure requires the numerical evaluation of the matrix

$$\mathbf{Y} = \mathbf{D}(t_\mathrm{f})\mathbf{B}(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{G}(t)^\mathsf{T}\mathbf{B}(t)\Big]\,\mathrm{d}t \in \mathbb{R}^{N_\mathrm{j}\times N_\mathrm{z}}, \tag{2.6}$$

where the matrices $\mathbf{D} \in \mathbb{R}^{N_\mathrm{j}\times N_\mathrm{n}}$, $\mathbf{F} \in \mathbb{R}^{N_\mathrm{j}\times N_\mathrm{z}}$, and $\mathbf{G} \in \mathbb{R}^{N_\mathrm{n}\times N_\mathrm{j}}$ are given. The matrix $\mathbf{B} \in \mathbb{R}^{N_\mathrm{n}\times N_\mathrm{z}}$ is the unknown of the linear matrix differential equation

$$\dot{\mathbf{B}}(t) = \mathbf{A}(t)\mathbf{B}(t) + \mathbf{C}(t) \quad \text{with} \quad \mathbf{B}(t_0) = \mathbf{0}, \tag{2.7}$$

in which the matrices $\mathbf{A} \in \mathbb{R}^{N_\mathrm{n}\times N_\mathrm{n}}$ and $\mathbf{C} \in \mathbb{R}^{N_\mathrm{n}\times N_\mathrm{z}}$ are given. The straightforward approach to evaluate $\mathbf{Y}$ is to first solve (2.7) for $\mathbf{B}$, and then use the solution of $\mathbf{B}$ in (2.6) to compute $\mathbf{Y}$. An alternative to this approach is to introduce a time-dependent matrix $\mathbf{P} \in \mathbb{R}^{N_\mathrm{n}\times N_\mathrm{j}}$, the so-called *dual* or *adjoint* matrix including adjoint variables, and replace (2.6) with

$$\tilde{\mathbf{Y}} = \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{P}(t)^\mathsf{T}\mathbf{C}(t)\Big]\,\mathrm{d}t, \tag{2.8}$$

where the adjoint matrix $\mathbf{P}$ is the unknown of the linear matrix differential equation

$$\dot{\mathbf{P}}(t) = -\mathbf{A}(t)^\mathsf{T}\mathbf{P}(t) - \mathbf{G}(t) \quad \text{with} \quad \mathbf{P}(t_\mathrm{f}) = \mathbf{D}(t_\mathrm{f})^\mathsf{T}. \tag{2.9}$$

The linear matrix differential equation constitutes a terminal-value problem that has to be solved backward in time due to the final condition [148]. The equations in (2.8) and (2.9) are referred to the *dual* problem corresponding to the *primal* problem formulation in (2.6) and (2.7). The dual problem corresponding to the primal problem is defined in such a way that the matrices $\tilde{\mathbf{Y}}$ and $\mathbf{Y}$ are equal. Substituting (2.7) and (2.9) into (2.8) yields

$$\begin{aligned}
\tilde{\mathbf{Y}} &= \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{P}(t)^\mathsf{T}\mathbf{C}(t)\Big]\,\mathrm{d}t \\
&= \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{P}(t)^\mathsf{T}\Big(\dot{\mathbf{B}}(t) - \mathbf{A}(t)\mathbf{B}(t)\Big)\Big]\,\mathrm{d}t \\
&= \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{P}(t)^\mathsf{T}\dot{\mathbf{B}}(t) - \Big(\mathbf{A}(t)^\mathsf{T}\mathbf{P}(t)\Big)^\mathsf{T}\mathbf{B}(t)\Big]\,\mathrm{d}t \\
&= \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{P}(t)^\mathsf{T}\dot{\mathbf{B}}(t) + \Big(\mathbf{G}(t) + \dot{\mathbf{P}}(t)\Big)^\mathsf{T}\mathbf{B}(t)\Big]\,\mathrm{d}t.
\end{aligned} \tag{2.10}$$

The integral of the term $\mathbf{P}(t)^\mathsf{T}\dot{\mathbf{B}}(t)$ is computed by applying integration by parts

$$\int_{t_0}^{t_\mathrm{f}} \mathbf{P}(t)^\mathsf{T}\dot{\mathbf{B}}(t)\,\mathrm{d}t = -\int_{t_0}^{t_\mathrm{f}} \dot{\mathbf{P}}(t)^\mathsf{T}\mathbf{B}(t)\,\mathrm{d}t + \underbrace{\mathbf{P}(t_\mathrm{f})^\mathsf{T}\mathbf{B}(t_\mathrm{f})}_{\mathbf{D}(t_\mathrm{f})\mathbf{B}(t_\mathrm{f})} - \underbrace{\mathbf{P}(t_0)^\mathsf{T}\mathbf{B}(t_0)}_{\mathbf{0}}, \tag{2.11}$$

where the boundary term at $t = t_0$ vanishes due to the initial conditions of the primal system (2.7). The equivalence of the primal and the dual problem can be shown by substituting (2.11) into (2.10), which results in

$$\begin{aligned}\tilde{\mathbf{Y}} =& \mathbf{D}(t_\mathrm{f})\mathbf{B}(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) - \dot{\mathbf{P}}(t)^\mathsf{T}\mathbf{B}(t) + \big(\mathbf{G}(t) + \dot{\mathbf{P}}(t)\big)^\mathsf{T}\mathbf{B}(t)\Big]\,\mathrm{d}t \\ =& \mathbf{D}(t_\mathrm{f})\mathbf{B}(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{G}(t)^\mathsf{T}\mathbf{B}(t)\Big]\,\mathrm{d}t = \mathbf{Y}.\end{aligned} \tag{2.12}$$

The equation in (2.12) states that $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ are equivalent if the primal matrix differential equation is solved for $\mathbf{B}$ and the dual matrix differential equation is solved for $\mathbf{P}$, respectively. Both perspectives require the solution of linear matrix differential equations to compute the matrices $\mathbf{B}$ and $\mathbf{P}$. Note that these matrices have the same number of rows $N_\mathrm{n}$, but the number of columns $N_\mathrm{z}$ for $\mathbf{B}$ and $N_\mathrm{j}$ for $\mathbf{P}$ is different. Thus, one has to solve either $N_\mathrm{z}$ different primal computations for $\mathbf{B}$ or $N_\mathrm{j}$ different dual computations for $\mathbf{P}$. This fact implies the computational benefits of using the dual approach when $N_\mathrm{j} < N_\mathrm{z}$. Using the dual approach is especially efficient when $N_\mathrm{j} \ll N_\mathrm{z}$. Moreover, the number of columns $N_\mathrm{z}$ of the matrix $\mathbf{Y}$ determines the number of columns for $\mathbf{B}$, which is not the case for $\mathbf{P}$. This characteristic is the most powerful aspect of linear duality [111].

## 2.3 The Adjoint Method for Sensitivity Analysis

This section exploits the concept of linear duality applied to local sensitivity analysis. Local sensitivities describe how the values of a function or functional $\mathbf{J} \in \mathbb{R}^{N_\mathrm{j}}$ changes to a small perturbation of the input $\mathbf{z} \in \mathbb{R}^{N_\mathrm{z}}$, i.e.,

$$\nabla_\mathbf{z}\mathbf{J}^\mathsf{T} = \frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{z}}\bigg|_{\mathbf{z}=\mathbf{z}_k}, \tag{2.13}$$

evaluated at the local point $\mathbf{z} = \mathbf{z}_k$. Various approaches can be used to compute the sensitivities in (2.13). However, this section focuses on the primal-dual approach to highlight the advantages of the *dual* approach. The dual or adjoint method is probably one of the most efficient approaches in sensitivity analysis since the number of underlying adjoint variables does not depend on the dimension of the input $\mathbf{z}$. The subsequent two sections discuss the primal-dual approach for sensitivities concerning the function and functional $\mathbf{J}$ in static and dynamic problems, respectively.

### 2.3.1 Static Problems

For static (time-invariant) problems, suppose that the function $\mathbf{J}$ is formulated with

$$\mathbf{J} = \mathbf{L}(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \xi(\mathbf{z})), \tag{2.14}$$

where $\mathbf{L} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \times \mathbb{R}^{N_\mathrm{l}} \to \mathbb{R}^{N_\mathrm{j}}$ specifies the function $\mathbf{J}$, $\mathbf{x} : \mathbb{R}^{N_\mathrm{z}} \to \mathbb{R}^{N_\mathrm{n}}$ is the vector of state variables, $\mathbf{u} : \mathbb{R}^{N_\mathrm{z}} \to \mathbb{R}^{N_\mathrm{m}}$ is the control, and $\xi : \mathbb{R}^{N_\mathrm{z}} \to \mathbb{R}^{N_\mathrm{l}}$ is a set of parameters. The state variables $\mathbf{x}$ are computed by the static equilibrium equations

$$\mathbf{x}(\mathbf{z}) = \mathbf{f}(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \xi(\mathbf{z})), \tag{2.15}$$

where $\mathbf{f} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \times \mathbb{R}^{N_\mathrm{l}} \to \mathbb{R}^{N_\mathrm{n}}$ is the right-hand side vector. For example, the right-hand side vector can represent the elongation of a nonlinear spring due to an acting force. The scope of this section is to compute the sensitivities of the function $\mathbf{J}$ in (2.14) with respect to the input $\mathbf{z}$. The total derivative of $\mathbf{J}$ is defined by

$$\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{z}} = \frac{\partial \mathbf{L}}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}. \tag{2.16}$$

Equation (2.16) requires the total derivative of the equilibrium equations (2.15) with respect to the input $\mathbf{z}$. Taking the total derivative of the equilibrium equations with respect to $\mathbf{z}$ yields

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}. \tag{2.17}$$

Note that the linear equations in (2.16) and (2.17) have the same characteristics as the primal problem introduced in equations (2.1) and (2.2). This can be seen by defining

$$\mathbf{A} \coloneqq \mathbf{I} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \quad \mathbf{B} \coloneqq \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}, \quad \mathbf{C} \coloneqq \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}, \quad \mathbf{F} \coloneqq \frac{\partial \mathbf{L}}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}, \quad \mathbf{G} \coloneqq \left( \frac{\partial \mathbf{L}}{\partial \mathbf{x}} \right)^\mathsf{T}, \tag{2.18}$$

where $\mathbf{I}$ is the identity matrix, and reformulating the equations (2.16) and (2.17) by using (2.18). Consequently, the total derivative in (2.16) is transformed into the standard form

$$\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{z}} = \mathbf{F} + \mathbf{G}^\mathsf{T}\mathbf{B}, \tag{2.19}$$

and the total derivative in (2.17) is also transformed into the standard form

$$\mathbf{A}\mathbf{B} = \mathbf{C}. \tag{2.20}$$

The straightforward approach to compute the sensitivities in (2.19) is to first solve (2.20) for $\mathbf{B}$, then use the solution to obtain the sensitivities. This procedure corresponds to the primal approach described in Section 2.2.1. However, it has to be mentioned that the dimension of the solution space for the linear system in (2.20) depends on the number of state variables $N_\mathrm{n}$ and the number of input variables $N_\mathrm{z}$. The dimension of the solution space changes linearly when the number of inputs $N_\mathrm{z}$ changes. Thus, the computational effort to compute (2.19) depends significantly on the computational effort to solve (2.20) for $\mathbf{B}$, especially when the number of inputs $N_\mathrm{z}$ is tremendous. The standard forms in (2.19) and (2.20) correspond to the primal approach formulated in Section 2.2.1. Consequently, the associated dual approach can be derived from the terminology of duality.

#### Duality Viewpoint

As shown in Section 2.2.1, the computation of the sensitivities in (2.19) can be transformed into its dual counterpart by introducing the dual system and the adjoint matrix $\mathbf{P}$ leading to

$$\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{z}} = \mathbf{F} + \mathbf{P}^\mathsf{T}\mathbf{C}, \tag{2.21}$$

where the adjoint matrix is computed according to the introduced adjoint equations in (2.4) with

$$\mathbf{A}^{\mathsf{T}}\mathbf{P} = \mathbf{G}. \tag{2.22}$$

The term $\mathbf{G}^{\mathsf{T}}\mathbf{B} \equiv \mathbf{P}^{\mathsf{T}}\mathbf{C}$ can be computed either by the primal approach, solving $\mathbf{AB} = \mathbf{C}$ for $\mathbf{B}$, or by the adjoint approach, solving $\mathbf{A}^{\mathsf{T}}\mathbf{P} = \mathbf{G}$ for $\mathbf{P}$. For a single input variable $N_z = 1$ there would be no benefit in using the adjoint approach, but for multiple input variables $N_z \gg 1$, the adjoint approach is computationally much more efficient. Moreover, the number of input variables $N_z$ does not influence the adjoint matrix and, therefore, also not the number of adjoint variables.

#### Lagrangian Viewpoint

The dual approach presented above uses the terminology of duality discussed in Section 2.2.1. An alternative description to derive the dual approach arises using the terminology of Lagrange multipliers [65]. Most textbooks and papers on adjoint methods are based on a Lagrangian viewpoint. In this alternative description, the function $\mathbf{J}$ is augmented by the equilibrium equations (2.15) and adjoint variables leading to

$$\bar{\mathbf{J}} = \mathbf{L}(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \xi(\mathbf{z})) + \mathbf{P}^{\mathsf{T}}\big[\mathbf{f}(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \xi(\mathbf{z})) - \mathbf{x}(\mathbf{z})\big]. \tag{2.23}$$

Note that the augmented function $\bar{\mathbf{J}}$ coincides with $\mathbf{J}$ for any choice of the adjoint variables in the case where the equilibrium equations (2.15) are satisfied. The sensitivities of the augmented function $\bar{\mathbf{J}}$ are given by

$$\frac{\mathrm{d}\bar{\mathbf{J}}}{\mathrm{d}\mathbf{z}} = \frac{\partial \mathbf{L}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \xi}\frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}} + \left[\frac{\partial \mathbf{L}}{\partial \mathbf{x}} - \mathbf{P}^{\mathsf{T}}\left(\mathbf{I} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\right]\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \mathbf{P}^{\mathsf{T}}\left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \xi}\frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}\right]. \tag{2.24}$$

To avoid the direct computation of the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ by solving (2.17), the adjoint matrix $\mathbf{P}$ may now be defined such that the rectangular bracket multiplied by the state sensitivities becomes zero, i.e., the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ do not have to be calculated directly as the entire term vanishes. To this end, the adjoint equations are defined by

$$\underbrace{\left(\mathbf{I} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{\mathsf{T}}}_{\mathbf{A}}\mathbf{P} = \underbrace{\left(\frac{\partial \mathbf{L}}{\partial \mathbf{x}}\right)^{\mathsf{T}}}_{\mathbf{G}}. \tag{2.25}$$

Using the definitions in (2.18), it is obvious that the adjoint equations in (2.25) derived by using the terminology of Lagrange multipliers are exactly the same as derived by using the terminology of duality. Moreover, if the adjoint equations in (2.25) are satisfied, then the total derivative in (2.24) reduces to

$$\frac{\mathrm{d}\bar{\mathbf{J}}}{\mathrm{d}\mathbf{z}} = \underbrace{\frac{\partial \mathbf{L}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \xi}\frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}}_{\mathbf{F}} + \mathbf{P}^{\mathsf{T}}\Bigg[\underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \xi}\frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}}_{\mathbf{C}}\Bigg]. \tag{2.26}$$

Note that the total derivative of the augmented function in (2.26) is exactly the same as derived by using the terminology of duality.

### 2.3.2 Dynamic Problems

For dynamic (time-dependent) problems, suppose that the functional $\mathbf{J}$ is formulated with

$$\mathbf{J} = \mathbf{E}(\mathbf{x}(\mathbf{z}, t_\mathrm{f}), t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \mathbf{L}(\mathbf{x}(\mathbf{z}, t), \mathbf{u}(\mathbf{z}, t), \boldsymbol{\xi}(\mathbf{z}))\,\mathrm{d}t, \tag{2.27}$$

where the state variables $\mathbf{x} : \mathbb{R}^{N_\mathrm{z}} \times \mathbb{R} \to \mathbb{R}^{N_\mathrm{n}}$ and the control $\mathbf{u} : \mathbb{R}^{N_\mathrm{z}} \times \mathbb{R} \to \mathbb{R}^{N_\mathrm{m}}$ are time-dependent, and an additional term $\mathbf{E} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R} \to \mathbb{R}^{N_\mathrm{j}}$ has been introduced. The state variables $\mathbf{x}$ are computed by the first-order differential equations

$$\dot{\mathbf{x}}(\mathbf{z}, t) = \mathbf{f}(\mathbf{x}(\mathbf{z}, t), \mathbf{u}(\mathbf{z}, t), \boldsymbol{\xi}(\mathbf{z})) \quad \text{with} \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \tag{2.28}$$

where $\mathbf{f} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \times \mathbb{R}^{N_\mathrm{l}} \to \mathbb{R}^{N_\mathrm{n}}$ is the right-hand side vector and $\bar{\mathbf{x}}_0$ are given initial state variables. For example, the right-hand side vector can represent the state vector of a nonlinear spring pendulum. The scope of this section is to compute the sensitivities of the functional $\mathbf{J}$ in (2.27) with respect to the input $\mathbf{z}$. The total derivative of $\mathbf{J}$ is defined by

$$\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{z}} = \frac{\partial \mathbf{E}}{\partial \mathbf{x}}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \int_{t_0}^{t_\mathrm{f}} \left( \frac{\partial \mathbf{L}}{\partial \mathbf{x}}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}} \right) \mathrm{d}t. \tag{2.29}$$

Similar to the static problem discussed above, the sensitivities (2.29) require the total derivative of the state variables $\mathbf{x}$ with respect to the input $\mathbf{z}$. Taking the total derivative of the state equations with respect to $\mathbf{z}$ yields

$$\frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}\mathbf{z}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}}. \tag{2.30}$$

Equation (2.30) is a system of linear differential equations that can be solved using classical time integration methods. The state variables are specified at the initial time by $\bar{\mathbf{x}}_0$, and therefore, changing the input $\mathbf{z}$ does not influence the state variables $\mathbf{x}$ at the initial time. Thus, the initial conditions to solve (2.30) are defined by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}(t_0) = \mathbf{0}. \tag{2.31}$$

Note that the equations in (2.29) and (2.30) have the same characteristics as the primal problem introduced in equations (2.6) and (2.7). This can be seen by defining

$$\begin{aligned}
&\mathbf{A}(t) := \frac{\partial \mathbf{f}}{\partial \mathbf{x}}, && \mathbf{B}(t) := \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}, && \dot{\mathbf{B}}(t) := \frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}\mathbf{z}},\\
&\mathbf{C}(t) := \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}}, && \mathbf{D}(t) := \frac{\partial \mathbf{E}}{\partial \mathbf{x}}, &&\\
&\mathbf{F}(t) := \frac{\partial \mathbf{L}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}}, && \mathbf{G}(t) := \left(\frac{\partial \mathbf{L}}{\partial \mathbf{x}}\right)^{\mathsf{T}}, &&
\end{aligned} \tag{2.32}$$

and reformulating the equations in (2.29) and (2.30) by using (2.32). Consequently, the total derivative in (2.29) is transformed into the standard form

$$\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{z}} = \mathbf{D}(t_\mathrm{f})\mathbf{B}(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \left[\mathbf{F}(t) + \mathbf{G}(t)^{\mathsf{T}}\mathbf{B}(t)\right] \mathrm{d}t, \tag{2.33}$$

and the total derivative in (2.30) is also transformed into the standard form

$$\dot{\mathbf{B}}(t) = \mathbf{A}(t)\mathbf{B}(t) + \mathbf{C}(t) \quad \text{with} \quad \mathbf{B}(t_0) = \mathbf{0}. \tag{2.34}$$

The straightforward approach to compute the sensitivities in (2.33) is to first solve (2.34) for $\mathbf{B}$, then use the solution to obtain the sensitivities. This procedure corresponds to the primal approach described in Section 2.2.2. However, it has to be mentioned that the dimension of the solution space for the linear matrix differential equation in (2.34) depends on the number of state variables $N_\mathrm{n}$ and the number of input variables $N_\mathrm{z}$. The dimension of the solution space changes linearly when the number of inputs $N_\mathrm{z}$ changes. Thus, the computational effort to compute (2.33) depends significantly on the computational effort to solve (2.34) for $\mathbf{B}$, especially when the number of inputs $N_\mathrm{z}$ is tremendous. The standard forms in (2.33) and (2.34) correspond to the primal approach formulated in Section 2.2.2. Consequently, the associated dual approach can be derived from the terminology of duality.

#### Duality Viewpoint

As shown in Section 2.2.2, the computation of the sensitivities in (2.33) can be transformed into its dual counterpart by introducing the dual system and the adjoint matrix $\mathbf{P}$ leading to

$$\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{z}} = \int_{t_0}^{t_\mathrm{f}} \Big[\mathbf{F}(t) + \mathbf{P}(t)^\mathsf{T}\mathbf{C}(t)\Big]\,\mathrm{d}t, \tag{2.35}$$

where the adjoint variables are computed according to the introduced adjoint system in (2.9) with

$$\dot{\mathbf{P}}(t) = -\mathbf{A}(t)^\mathsf{T}\mathbf{P}(t) - \mathbf{G}(t) \quad \text{with} \quad \mathbf{P}(t_\mathrm{f}) = \mathbf{D}(t_\mathrm{f})^\mathsf{T}. \tag{2.36}$$

The linear matrix differential equation in (2.36) constitutes a terminal-value problem that has to be solved backward in time. Thus, the history of all state variables is required first to evaluate $\mathbf{A}$ and $\mathbf{G}$ backward in time starting from $t_\mathrm{f}$ towards $t_0$. The adjoint variables can, therefore, not be computed simultaneously with the state variables [154]. However, the time-dependent term $\mathbf{G}^\mathsf{T}\mathbf{B} \equiv \mathbf{P}^\mathsf{T}\mathbf{C}$ can be computed either by the primal approach, solving $\dot{\mathbf{B}} = \mathbf{A}\mathbf{B} + \mathbf{C}$ for $\mathbf{B}$, or by the adjoint approach, solving $\dot{\mathbf{P}} = -\mathbf{A}^\mathsf{T}\mathbf{P} - \mathbf{G}$ for $\mathbf{P}$. For a single input variable $N_\mathrm{z} = 1$ there would be no benefit in using the adjoint approach, but for multiple input variables $N_\mathrm{z} \gg 1$, the adjoint approach is computationally much more efficient. Moreover, the number of input variables $N_\mathrm{z}$ does not influence the adjoint matrix and, therefore, also not the number of adjoint variables.

#### Lagrangian Viewpoint

The dual approach presented above uses the terminology of duality discussed in Section 2.2.2. An alternative description to derive the dual approach arises using the terminology of Lagrange multipliers [65]. In this alternative description, the functional $\mathbf{J}$ is augmented by the first-order differential equations (2.28) and adjoint variables leading to

$$\bar{\mathbf{J}} = \mathbf{E}(\mathbf{x}(\mathbf{z}, t_\mathrm{f}), t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \left[\mathbf{L}(\mathbf{x}(\mathbf{z}, t), \mathbf{u}(\mathbf{z}, t), \xi(\mathbf{z})) + \mathbf{P}^\mathsf{T}\left(\mathbf{f}(\mathbf{x}(\mathbf{z}, t), \mathbf{u}(\mathbf{z}, t), \xi(\mathbf{z})) - \dot{\mathbf{x}}(\mathbf{z}, t)\right)\right] \mathrm{d}t. \tag{2.37}$$

Note that the augmented functional $\bar{\mathbf{J}}$ coincides with $\mathbf{J}$ for any choice of the adjoint variables in the case where the state equations (2.28) are satisfied. The sensitivities of the augmented functional $\bar{\mathbf{J}}$ are given by

$$\begin{aligned}\frac{\mathrm{d}\bar{\mathbf{J}}}{\mathrm{d}\mathbf{z}} = &\frac{\partial \mathbf{E}}{\partial \mathbf{x}}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \int_{t_0}^{t_\mathrm{f}} \left[\left(\mathbf{P}^\mathsf{T}\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \frac{\partial \mathbf{L}}{\partial \mathbf{x}}\right)\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \xi}\frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}}\right.\\ &\left. + \mathbf{P}^\mathsf{T}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \xi}\frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}} - \frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}\mathbf{z}}\right)\right] \mathrm{d}t,\end{aligned} \tag{2.38}$$

where the term $\frac{\partial \mathbf{E}}{\partial \mathbf{x}}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}$ is evaluated at the final time $t = t_\mathrm{f}$. The integral of the term $\mathbf{P}^\mathsf{T}\frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}\mathbf{z}}$ is computed by applying integration by parts

$$-\int_{t_0}^{t_\mathrm{f}} \mathbf{P}^\mathsf{T}\frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}\mathbf{z}}\,\mathrm{d}t = \int_{t_0}^{t_\mathrm{f}} \dot{\mathbf{P}}^\mathsf{T}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}\,\mathrm{d}t - \mathbf{P}(t_\mathrm{f})^\mathsf{T}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}(t_\mathrm{f}) + \underbrace{\mathbf{P}(t_0)^\mathsf{T}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}(t_0)}_{\mathbf{0}}, \tag{2.39}$$

where the boundary term at $t = t_0$ vanishes due to the initial conditions (2.31). Substituting (2.39) in (2.38) yields to

$$\begin{aligned}\frac{\mathrm{d}\bar{J}}{\mathrm{d}\mathbf{z}} =& \left(\frac{\partial \mathbf{E}}{\partial \mathbf{x}} - \mathbf{P}^\mathsf{T}\right)\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \int_{t_0}^{t_\mathrm{f}}\Bigg[\left(\dot{\mathbf{P}}^\mathsf{T} + \mathbf{P}^\mathsf{T}\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \frac{\partial \mathbf{L}}{\partial \mathbf{x}}\right)\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} \\ &+ \frac{\partial \mathbf{L}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}} + \mathbf{P}^\mathsf{T}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}}\right)\Bigg]\mathrm{d}t.\end{aligned} \tag{2.40}$$

To avoid the direct computation of the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ by solving (2.30), the adjoint matrix $\mathbf{P}$ may now be defined such that the round brackets multiplied by the state sensitivities become zero, i.e., the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ do not have to be calculated directly as the entire term vanishes. To this end, the adjoint equations are defined by

$$\dot{\mathbf{P}} = -\underbrace{\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^\mathsf{T}}_{\mathbf{A}}\mathbf{P} - \underbrace{\left(\frac{\partial \mathbf{L}}{\partial \mathbf{x}}\right)^\mathsf{T}}_{\mathbf{G}} \quad \text{with} \quad \mathbf{P}(t_\mathrm{f}) = \underbrace{\left(\frac{\partial \mathbf{E}}{\partial \mathbf{x}}\right)^\mathsf{T}}_{\mathbf{D}}. \tag{2.41}$$

Using the definitions in (2.32), it is obvious that the adjoint equations in (2.41) derived by using the terminology of Lagrange multipliers are exactly the same as those derived by using the terminology of duality. Moreover, if the adjoint equations in (2.41) are satisfied, then the total derivative in (2.40) reduces to

$$\frac{\mathrm{d}\bar{J}}{\mathrm{d}\mathbf{z}} = \int_{t_0}^{t_\mathrm{f}}\Bigg[\underbrace{\frac{\partial \mathbf{L}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{L}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}}}_{\mathbf{F}} + \mathbf{P}^\mathsf{T}\underbrace{\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}}\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{z}}\right)}_{\mathbf{C}}\Bigg]\mathrm{d}t. \tag{2.42}$$

Note that the total derivative of the augmented functional in (2.42) is exactly the same as derived by using the terminology of duality.

## 2.4 Computational Aspects

This section discusses computational aspects relevant to adjoint-based sensitivity analysis. High-fidelity and efficient sensitivity analysis is crucial, e.g., if the computed sensitivities are used in a gradient-based optimization approach. In particular, this section aims to highlight computational aspects to compute sensitivities $\mathrm{d}J/\mathrm{d}\mathbf{z}$ of the scalar functional

$$J = \int_{t_0}^{t_\mathrm{f}} L(\mathbf{x}(\mathbf{z},t),\mathbf{u}(\mathbf{z},t))\,\mathrm{d}t, \tag{2.43}$$

where the state variables are computed by the first-order differential equations

$$\dot{\mathbf{x}}(\mathbf{z},t) = \mathbf{f}(\mathbf{x}(\mathbf{z},t),\mathbf{u}(\mathbf{z},t)) \quad \text{with} \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0. \tag{2.44}$$

### 2.4.1 Duality of Sensitivities

Adjoint methods can be interpreted as a special case of linear duality, and their core is based on the substitution of variables [111]. The adjoint method is often used in sensitivity analysis to compute first-order derivatives. Adjoint sensitivities are based on the underlying adjoint variables, whose dimension is independent of the number of input variables $N_z$. Instead, the dimension of the adjoint variables is proportional to the dimension of the function of interest $N_j$. However, the dimension of the function of interest is usually smaller than the number of input variables. For such problems, the adjoint method is one of the most efficient methods to evaluate sensitivities [37].

Figure 2.1 visualizes the dimensions of primal ($a$) and dual ($b$) sensitivities as well as the dimensions of primal ($c$) and dual ($d$) differential equations for two different sets of input variables. Increasing the number of input variables $\mathbf{z}$ from $N_z$ to $N_z + i$, with $i \geq 1 \in \mathbb{N}$, changes the dimensions of the derivatives with respect to the input $\mathbf{z}$, while the dimensions of the derivatives with respect to the state variables $\mathbf{x}$ are independent of the number of input variables $N_z$; see Fig. 2.1 for a graphical interpretation of the dimensions. Note that the primal ($a$) and dual ($b$) sensitivities require the solution of linear differential equations for $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ and $\mathbf{p}$, respectively. As aforementioned, the dimension of the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ changes with the number of the input variables $N_z$, while the dimension of the adjoint variables $\mathbf{p}$ are independent of the number of input variables $N_z$. This can also be seen in Fig. 2.1 for the dimensions of the linear differential equations associated with the primal ($c$) and dual ($d$) approach.

### 2.4.2 Backward Time Integration

The computation of the adjoint matrix $\mathbf{P}$ requires solving differential equations backward in time, starting from the final time $t = t_f$ towards the initial time $t = t_0$, i.e., the interval of integration $(t_f, t_0)$ is decreasing. The terminal-value problem in (2.9) can be transformed into a classical initial-value problem by using a time transformation [148]. Following Nachbagauer et al. [113], the time transformation can be performed by introducing a new time domain $\tau \in [t_0, t_f] : \mathbb{R} \to \mathbb{R}$ defined by

$$\tau = t_f - t. \tag{2.45}$$

Using the time transformation rule with an increasing interval of integration for $t$ constructs a decreasing $\tau$. Hence, the new time domain $\tau$ can be used to compute the adjoint variables backward in time with an increasing interval of integration for $t$. The transformation requires a reformulation of the linear matrix differential equation in (2.9) by the new time domain. Special attention has to be given to the total time derivative of a function $(\cdot)$ depending on the new time domain, i.e., $(\cdot)(\tau) = (\cdot)(t_f - t)$, which is defined by

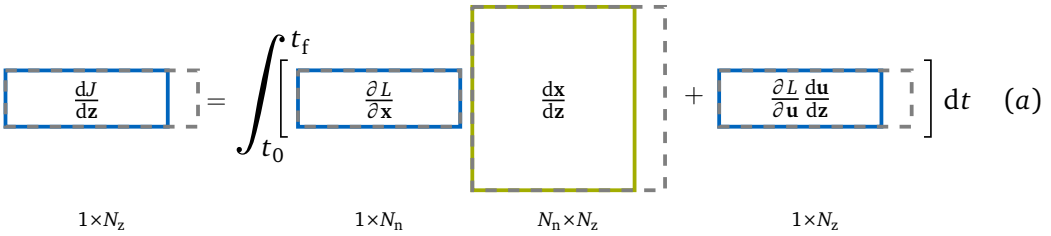
$$\frac{\mathrm{d}(\cdot)(\tau)}{\mathrm{d}t} = \frac{\mathrm{d}(\cdot)(\tau)}{\mathrm{d}\tau}\frac{\mathrm{d}\tau}{\mathrm{d}t} = -\frac{\mathrm{d}(\cdot)(\tau)}{\mathrm{d}\tau}. \tag{2.46}$$

By using the transformation rule (2.45) and the total time derivative (2.46), the linear matrix differential equation in (2.9) can be reformulated as a function of the new time domain

$$\frac{\mathrm{d}\mathbf{P}(\tau)}{\mathrm{d}\tau} = \mathbf{A}(\tau)^{\mathsf{T}}\mathbf{P}(\tau) + \mathbf{G}(\tau) \quad \text{with} \quad \mathbf{P}(t_f) = \mathbf{D}(t_f)^{\mathsf{T}}. \tag{2.47}$$

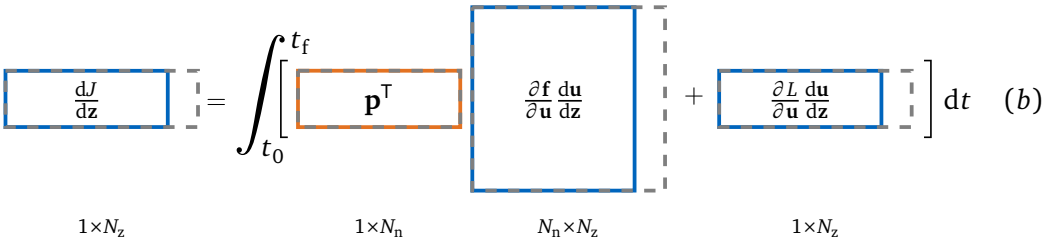
The reformulated linear matrix differential equation is an initial-value problem that can be solved by any suitable third-party library for solving differential equations.

*primal* sensitivities



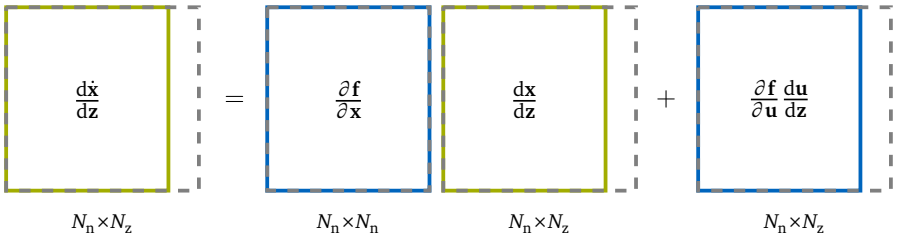

(a)

*dual* or *adjoint* sensitivities



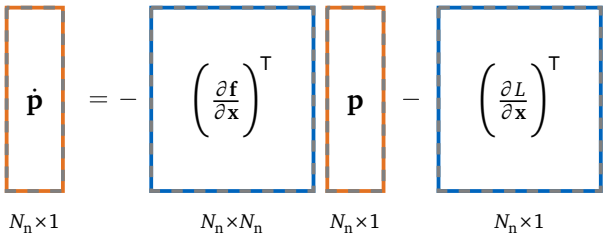

(b)

*primal* matrix differential equation




(c)

*dual* or *adjoint* matrix differential equation




(d)

—— dimensions with $N_\mathrm{z}$ input variables - - - dimensions with $N_\mathrm{z}+i$ input variables

**Figure 2.1:** Graphical interpretation of the dimensions in the sensitivity analysis in dynamics: $(a)$ and $(b)$ represent the sensitivities associated with the *primal* and *dual* approach, respectively; $(c)$ and $(d)$ represent the linear differential equations associated with the *primal* and *dual* approach, respectively

### 2.4.3 Memory Efficient Implementation

To fully utilize the advantages of the adjoint method, it is crucial to implement it memory-efficiently. In the last few decades, various approaches have been developed to implement the adjoint method in a memory-efficient way. The memory requirement of the adjoint method can be demanding since the entire history of the state variables for all $t \in [t_0, t_f]$ is required to compute the adjoint variables. This can be a demanding storage requirement, especially for large-scale problems. To overcome this memory overhead, Griewank [72] introduced the concept of checkpointing. The key idea of the checkpointing method is to store the state variables $\mathbf{x}$ only at specified checkpoints instead of the entire history. A comprehensive analysis of the checkpointing method is provided in [73]. Checkpointing is also used in a recent paper by Gholami et al. [64], which presents a memory-efficient adjoint-based neural ordinary differential equation framework that provides unconditionally accurate gradients. A memory-efficient approach in neural networks is using a symplectic adjoint method as proposed in [110], where a symplectic integrator solves the adjoint system with appropriate checkpoints. Another approach to reducing the memory footprint for implementations based on automatic differentiation can be realized using the operator overloading-based adjoint approach presented in [48]. In addition, a comprehensive benchmark in [92] presents various adjoint implementations focusing on efficiency, accuracy, and low-cost implementation.

### 2.4.4 Procedure of Adjoint Sensitivities

Using adjoint sensitivities in dynamics can be summarized with the following four major steps:

1. **Computation of partial derivatives**
   The sensitivity analysis requires the computation of partial derivatives, e.g., $\partial\mathbf{f}/\partial\mathbf{x}$ or $\partial\mathbf{f}/\partial\mathbf{u}$. Partial derivatives can be computed mainly by four different methods: finite-difference method, complex-step method, automatic differentiation, and symbolic differentiation [92].

2. **Forward time integration**
   The evolution of the state variables $\mathbf{x}$ is computed forward in time using a classical time integration method to solve first-order differential equations, e.g., using a Runge–Kutta time-integration scheme [32]. Explicit or implicit time integration methods can be used depending on the characteristics of the differential equations.

3. **Backward time integration**
   The evolution of the adjoint matrix $\mathbf{P}$ is computed backward in time starting from the final time $t = t_f$ towards the initial time $t = t_0$. Since the linear matrix differential equation depends on the state variables $\mathbf{x}$, the forward time integration must be finished before the backward time integration can be performed.

4. **Computation of sensitivities**
   The final step in the procedure to obtain abjoint-based sensitivities requires to compute an integral over time. The integral can be solved using classical numerical integration methods, such as a quadrature method [41].

Chapter 3

# Fundamentals of Optimal Control

Optimal control aims to determine inputs that drive a dynamical system towards its desired goal. In a mathematical context, optimal control regards optimizing a dynamical system to minimize a cost functional while satisfying physical constraints, e.g., the rest-to-rest motion of a soft robot. The dynamical system under consideration may be described as nonlinear first-order differential equations

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\xi}) \quad \text{with} \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^{N_\mathrm{n}}$ is the vector of state variables, $\mathbf{u} \in \mathbb{R}^{N_\mathrm{m}}$ is the control, and $\boldsymbol{\xi} \in \mathbb{R}^{N_\mathrm{l}}$ is a set of parameters. The ordinary differential equation (ODE) in (3.1) together with the initial state variables $\bar{\mathbf{x}}_0$ is referred to as an initial value problem (IVP).

As the complexity of OCPs increases, numerical methods are employed to obtain an optimal solution. Numerical solution methods for OCPs can be divided into three main categories:

- **Indirect methods** are based on the calculus of variations to obtain the necessary first-order optimality conditions of the OCP. The optimality conditions are in the form of a two-point boundary value problem (BVP), which has to be solved to determine the optimal control. This approach is also called *first differentiate, then optimize*.
- **Direct methods** are based on parameterizing the state and/or control variables to transcribe the original infinite-dimensional optimization problem into a finite-dimensional NLP problem. The NLP problem can be treated by well-known methods to determine the optimal control. This approach is also called *first discretize, then optimize*.
- **Dynamic programming** is based on the principle of optimality, which leads to partial differential equations referred to as the Hamilton-Jacobi-Bellman equations. For details on dynamic programming, the reader is referred to [8, 10].

The main approaches for solving an OCP using indirect and direct methods are discussed in a comprehensive survey of numerical methods provided by Rao [129]. In addition, von Stryk and Bulirsch [159] provide commonly used methods for solving an OCP and introduce a hybrid approach to combine indirect and direct methods. An overview of numerical methods for solving an OCP is given in Fig. 3.1 focusing on direct methods. Indirect and direct methods are discussed further as the theoretical basis for this dissertation.

The scope of this chapter is to provide sufficient details to understand the scientific contributions of the author's publications in [99, 101, 102]. It is not an aim to provide a complete treatment of theory and applications on optimal control. Instead, interested readers are referred to excellent textbooks, e.g., on general optimal control theory [14, 30, 95], and on convex and numerical optimization [25, 69, 115]. The theoretical aspects of optimal control presented in the next sections mainly follow the work of these textbooks.

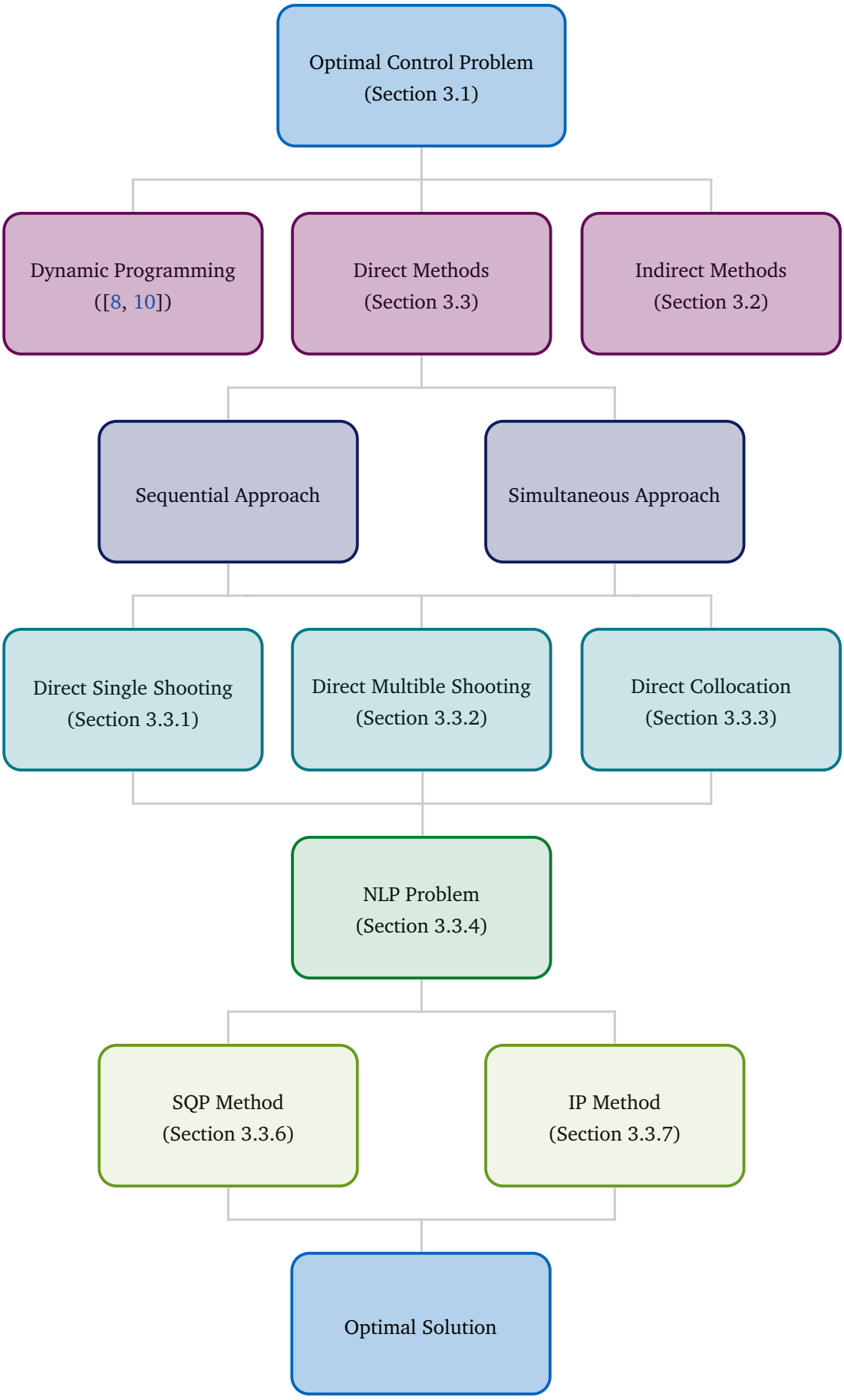



**Figure 3.1:** Overview of numerical methods for solving optimal control problems focusing on direct methods

## 3.1 Problem Formulation

An OCP aims to determine a continuous control $\mathbf{u}(t) = \mathbf{u}^*$, continuous state variables $\mathbf{x}(t) = \mathbf{x}^*$, a final time $t_\mathrm{f} = t_\mathrm{f}^*$ and possibly a set of parameters $\boldsymbol{\xi} = \boldsymbol{\xi}^*$ to minimize a scalar cost functional $J$ without violating (in)equality constraints. The control and the state variables are time-dependent functions in the time interval $t \in [t_0, t_\mathrm{f}]$ and, therefore, the OCP is an infinite-dimensional optimization problem. An OCP is formulated as

$$\min_{\mathbf{x},\mathbf{u},\boldsymbol{\xi},t_\mathrm{f}} J = \int_{t_0}^{t_\mathrm{f}} L(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\xi}, t)\,\mathrm{d}t + E(\mathbf{x}(t_\mathrm{f}), t_\mathrm{f}) \tag{3.2}$$

s.t.

$$\boldsymbol{\phi}(\mathbf{x}(t_\mathrm{f}), t_\mathrm{f}) = \mathbf{0} \tag{3.3}$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0} \tag{3.4}$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \tag{3.5}$$

$$\mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \tag{3.6}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\xi}) \tag{3.7}$$

where the integral cost $L : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \times \mathbb{R}^{N_\mathrm{l}} \times \mathbb{R} \to \mathbb{R}$ is denoted as Lagrange term and the final cost $E : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R} \to \mathbb{R}$ is denoted as Mayer term. An OCP formulated with a Lagrange term and a Mayer term is referred to as a Bolza type. Different forms of an OCP are presented in [23] or in [116]. The above-defined OCP is concerned to final constraints $\boldsymbol{\phi} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R} \to \mathbb{R}^{N_{\boldsymbol{\phi}}}$, equality constraints $\mathbf{g} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \to \mathbb{R}^{N_\mathrm{g}}$, and inequality constraints $\mathbf{h} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \to \mathbb{R}^{N_\mathrm{h}}$ while satisfying the state equations with the given initial state variables $\bar{\mathbf{x}}_0$ in the time interval $t \in [t_0, t_\mathrm{f}]$.

As aforementioned, the OCP can be treated by indirect methods, direct methods, and dynamic programming. The theoretical concept of indirect and direct methods is discussed in the subsequent sections.

## 3.2 Indirect Methods

Indirect methods are proven to provide results with high accuracy regarding an OCP. The basic idea is to derive the necessary first-order optimality conditions by applying the calculus of variations and to solve the resulting equations. These optimality conditions formulate a two-point BVP. An analytic solution of the derived BVP can only be found in some special cases. For the general case, iterative methods are employed to solve the BVP, e.g., indirect shooting, indirect collocation, or indirect gradient-based methods. The following section derives the necessary first-order optimality conditions using a variational approach for an unbounded OCP. These conditions are then extended to incorporate bounds, which leads to Pontryagin's minimum principle.

### 3.2.1 Unbounded Optimal Control Problems

This section focuses on formulating necessary first-order optimality conditions for the simple but fundamentally important case in which the OCP is unbounded. The simplification yields

the unbounded OCP formulation

$$\min_{\mathbf{x},\mathbf{u},t_\mathrm{f}} J = \int_{t_0}^{t_\mathrm{f}} L(\mathbf{x}(t),\mathbf{u}(t),t)\,\mathrm{d}t + E(\mathbf{x}(t_\mathrm{f}),t_\mathrm{f}) \tag{3.8}$$

s.t.

$$\mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \tag{3.9}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) \tag{3.10}$$

Note that (soft) inequality constraints on the state variables and the control can be considered by introducing a proper penalty function $P(\mathbf{x},\mathbf{u}) : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \to \mathbb{R}$ added to the integral cost $L$. The penalty function is zero if the inequality constraints are satisfied and increases if the inequality constraints are violated. Detailed information on the practical use of penalty functions is presented in [14]. In addition to inequality constraints, the final constraints of the state variables can be considered in the Mayer term $E$. However, the above OCP formulation is used in the subsequent sections to derive the necessary first-order optimality conditions.

Following the fundamental work by Bryson and Ho [30], an extended cost functional

$$\bar{J} = \int_{t_0}^{t_\mathrm{f}} \left[ L(\mathbf{x}(t),\mathbf{u}(t),t) + \mathbf{p}(t)^\mathsf{T} \left(\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) - \dot{\mathbf{x}}(t)\right)\right] \mathrm{d}t + E(\mathbf{x}(t_\mathrm{f}),t_\mathrm{f}) \tag{3.11}$$

is formulated by coupling the state equations (3.10) with the cost functional (3.8). When the state equations are satisfied, the additional term does not change the numerical value of the cost functional $J$ for any choice of the so-called adjoint variables $\mathbf{p} \in \mathbb{R}^{N_\mathrm{n}}$. For the sake of convenience, the Hamiltonian $\mathcal{H} : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{m}} \times \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R} \to \mathbb{R}$ associated with the extended cost functional is introduced by

$$\mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{p}(t),t) \coloneqq L(\mathbf{x}(t),\mathbf{u}(t),t) + \mathbf{p}^\mathsf{T}(t)\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)). \tag{3.12}$$

Using the Hamiltonian, the extended cost functional reads

$$\bar{J} = \int_{t_0}^{t_\mathrm{f}} \left[ \mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{p}(t),t) - \mathbf{p}(t)^\mathsf{T}\dot{\mathbf{x}}(t)\right] \mathrm{d}t + E(\mathbf{x}(t_\mathrm{f}),t_\mathrm{f}). \tag{3.13}$$

The goal is to find optimal functions $\mathbf{x}^*$, $\mathbf{u}^*$, $\mathbf{p}^*$, and a final time $t_\mathrm{f}^*$, such that the extended cost functional (3.13) becomes stationary. The optimality of indirect methods is investigated by a perturbation of the optimal solution. The modified functions are given by

$$\mathbf{x}(t) = \mathbf{x}^*(t) + \delta\mathbf{x}(t) = \mathbf{x}^*(t) + \varepsilon\boldsymbol{\eta}_1(t), \tag{3.14}$$

$$\mathbf{u}(t) = \mathbf{u}^*(t) + \delta\mathbf{u}(t) = \mathbf{u}^*(t) + \varepsilon\boldsymbol{\eta}_2(t), \tag{3.15}$$

$$\mathbf{p}(t) = \mathbf{p}^*(t) + \delta\mathbf{p}(t) = \mathbf{p}^*(t) + \varepsilon\boldsymbol{\eta}_3(t), \tag{3.16}$$

where $\delta\mathbf{x}$, $\delta\mathbf{u}$, $\delta\mathbf{p}$ are admissible variations, $\boldsymbol{\eta}_{1,2,3}$ are proper test functions and $\varepsilon$ is a small perturbation parameter. For the following derivation of the necessary optimality conditions, the argument $t$ of time-dependent functions is omitted to improve the readability. In the general case, the final time is free and, therefore, also modified by

$$t_\mathrm{f} = t_\mathrm{f}^* + \delta t_\mathrm{f} = t_\mathrm{f}^* + \varepsilon\eta_4, \tag{3.17}$$

where $\delta t_\mathrm{f}$ is the variation of the final time and $\eta_4$ is a test parameter. In addition, the variation of the extended cost functional $\delta\bar{J}$ caused by a perturbation of the optimal solution in (3.14)–(3.17) is given as

$$\delta\bar{J} = \bar{J} - \bar{J}^* = \bar{J}(\mathbf{x}^* + \varepsilon\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3, t_\mathrm{f}^* + \varepsilon\eta_4) - \bar{J}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t_\mathrm{f}^*). \tag{3.18}$$

The extended cost functional caused by a perturbation of the optimal solution can be expressed in terms of a Taylor series expansion. The linear approximation of the extended cost functional yields

$$\begin{aligned}\bar{J}(\mathbf{x}^* + \varepsilon\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3, t_\mathrm{f}^* + \varepsilon\eta_4) \\ \approx \bar{J}(\mathbf{x}^* + \varepsilon_0\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon_0\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon_0\boldsymbol{\eta}_3, t_\mathrm{f}^* + \varepsilon_0\eta_4) \\ + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=\varepsilon_0}\Big\{\bar{J}(\mathbf{x}^* + \varepsilon\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3, t_\mathrm{f}^* + \varepsilon\eta_4)\Big\}(\varepsilon - \varepsilon_0),\end{aligned} \tag{3.19}$$

where $\varepsilon_0$ is the expansion point. Considering the linear approximation at the optimal solution, i.e., $\varepsilon_0 = 0$, and substituting (3.19) into (3.18) yields the variation of the extended cost functional

$$\delta\bar{J} = \varepsilon\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0}\Big\{\bar{J}(\mathbf{x}^* + \varepsilon\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3, t_\mathrm{f}^* + \varepsilon\eta_4)\Big\}. \tag{3.20}$$

The variation of the extended cost functional formulates the necessary first-order optimality conditions of an optimal solution. For an optimal solution, the variation becomes stationary, i.e., the condition $\delta\bar{J} = 0$ has to be satisfied. The variation of the extended cost functional (3.20) is written in terms of the extended cost functional introduced in (3.13), which results in

$$\begin{aligned}\delta\bar{J} =& \varepsilon\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0}\Big\{E(\mathbf{x}^*(t_\mathrm{f}^* + \varepsilon\eta_4) + \varepsilon\boldsymbol{\eta}_1(t_\mathrm{f}^* + \varepsilon\eta_4), t_\mathrm{f}^* + \varepsilon\eta_4) \\ &+ \int_{t_0}^{t_\mathrm{f}^*}\Big[\mathcal{H}(\mathbf{x}^* + \varepsilon\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3, t) - (\mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3)^\mathsf{T}(\dot{\mathbf{x}}^* + \varepsilon\dot{\boldsymbol{\eta}}_1)\Big]\mathrm{d}t \\ &+ \int_{t_\mathrm{f}^*}^{t_\mathrm{f}^* + \varepsilon\eta_4}\Big[\mathcal{H}(\mathbf{x}^* + \varepsilon\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3, t) - (\mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3)^\mathsf{T}(\dot{\mathbf{x}}^* + \varepsilon\dot{\boldsymbol{\eta}}_1)\Big]\mathrm{d}t\Big\}.\end{aligned} \tag{3.21}$$

Therein, the interval of the integration is divided into two parts.

The next step in deriving the optimality conditions is to perform the total derivative of the extended cost functional with respect to the perturbation parameter. The total derivative with respect to the perturbation parameter can be carried out straightforwardly for the Mayer term and for the integral in the time interval $t \in [t_0, t_\mathrm{f}^*]$ by applying consequently the chain rule of differentiation. The total derivative of the Mayer term results in

$$\varepsilon\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0}\Big\{E(\mathbf{x}(t_\mathrm{f}(\varepsilon), \varepsilon), t_\mathrm{f}(\varepsilon))\Big\} = \varepsilon\bigg\{\frac{\partial E(\mathbf{x}, t_\mathrm{f})}{\partial\mathbf{x}}\Big[\frac{\partial\mathbf{x}}{\partial t_\mathrm{f}}\frac{\partial t_\mathrm{f}}{\partial\varepsilon} + \frac{\partial\mathbf{x}}{\partial\varepsilon}\Big] + \frac{\partial E(\mathbf{x}, t_\mathrm{f})}{\partial t_\mathrm{f}}\frac{\partial t_\mathrm{f}}{\partial\varepsilon}\bigg\}\bigg|_{\varepsilon=0}. \tag{3.22}$$

Evaluating the above total derivative at $\varepsilon = 0$ yields

$$\begin{aligned}\varepsilon\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0}\Big\{E(\mathbf{x}(t_\mathrm{f}(\varepsilon), \varepsilon), t_\mathrm{f}(\varepsilon))\Big\} &= \varepsilon\bigg\{\frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial\mathbf{x}}\Big[\dot{\mathbf{x}}^*(t_\mathrm{f}^*)\eta_4 + \boldsymbol{\eta}_1(t_\mathrm{f}^*)\Big] + \frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial t}\eta_4\bigg\} \\ &= \frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial\mathbf{x}}\Big[\underbrace{\dot{\mathbf{x}}^*(t_\mathrm{f}^*)\delta t_\mathrm{f} + \delta\mathbf{x}(t_\mathrm{f}^*)}_{\delta\mathbf{x}_\mathrm{f}}\Big] + \frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial t}\delta t_\mathrm{f},\end{aligned} \tag{3.23}$$

where the product of the perturbation parameter times a test parameter/function is expressed with the corresponding variation defined in (3.14)–(3.17). The majority of papers and textbooks on optimal control theory introduce the variation $\delta\mathbf{x}_\mathrm{f}$ first and then derive the explicit

terms with

$$\begin{aligned}\delta\mathbf{x}_\mathrm{f} &= \mathbf{x}(t_\mathrm{f}^* + \delta t_\mathrm{f}) - \mathbf{x}^*(t_\mathrm{f}^*) \\ &= \mathbf{x}^*(t_\mathrm{f}^* + \delta t_\mathrm{f}) + \delta\mathbf{x}(t_\mathrm{f}^* + \delta t_\mathrm{f}) - \mathbf{x}^*(t_\mathrm{f}^*) \\ &\approx \mathbf{x}^*(t_\mathrm{f}^*) + \dot{\mathbf{x}}^*(t_\mathrm{f}^*)\delta t_\mathrm{f} + \delta\mathbf{x}(t_\mathrm{f}^*) + \delta\dot{\mathbf{x}}(t_\mathrm{f}^*)\delta t_\mathrm{f} - \mathbf{x}^*(t_\mathrm{f}^*) \\ &\approx \dot{\mathbf{x}}^*(t_\mathrm{f}^*)\delta t_\mathrm{f} + \delta\mathbf{x}(t_\mathrm{f}^*),\end{aligned} \tag{3.24}$$

where the state variables $\mathbf{x}^*(t_\mathrm{f}^* + \delta t_\mathrm{f})$ and the variation $\delta\mathbf{x}(t_\mathrm{f}^* + \delta t_\mathrm{f})$ are approximated by a first-order Taylor series expansion at the expansion point $t_\mathrm{f}^*$, and where the higher-order term $\delta\dot{\mathbf{x}}(t_\mathrm{f}^*)\delta t_\mathrm{f}$ is neglected. Note that the variation of the final time $\delta t_\mathrm{f}$ influences the variation of the state variables $\delta\mathbf{x}$; see Fig. 3.2 for a visualization of the relationship of the variations. In the special case of a fixed final time, the variation (3.24) simplifies to $\delta\mathbf{x}_\mathrm{f} = \delta\mathbf{x}(t_\mathrm{f}^*)$.

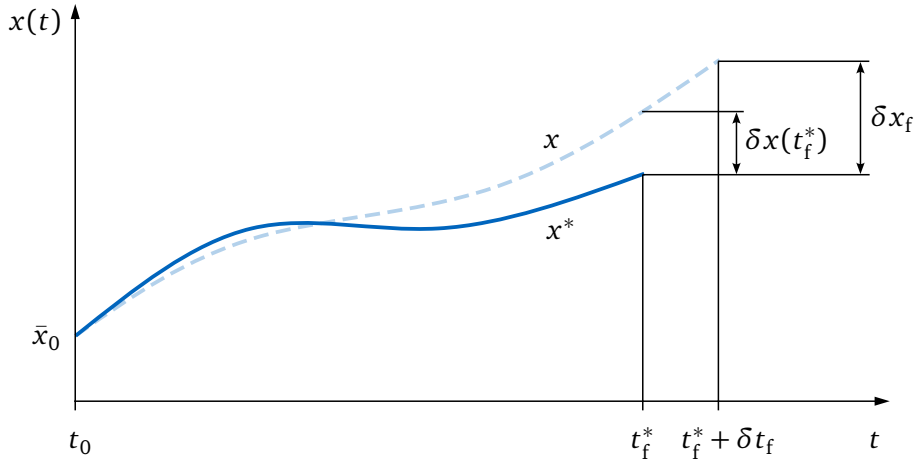



**Figure 3.2:** Optimal and modified state variable: Relationship between $\delta x(t_\mathrm{f})$, $\delta x_\mathrm{f}$ and $\delta t_\mathrm{f}$

In addition to the total derivative of the Mayer term, the total derivative of the second term in (3.21) reads

$$\begin{aligned}&\varepsilon\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0}\left\{\int_{t_0}^{t_\mathrm{f}^*}\Big[\mathcal{H}(\mathbf{x}^*+\varepsilon\boldsymbol{\eta}_1,\mathbf{u}^*+\varepsilon\boldsymbol{\eta}_2,\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3,t)-(\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3)^\mathsf{T}(\dot{\mathbf{x}}^*+\varepsilon\dot{\boldsymbol{\eta}}_1)\Big]\mathrm{d}t\right\} \\ &\qquad= \int_{t_0}^{t_\mathrm{f}^*}\bigg[\frac{\partial\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)}{\partial\mathbf{x}}\delta\mathbf{x}+\frac{\partial\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)}{\partial\mathbf{u}}\delta\mathbf{u} \\ &\qquad+\frac{\partial\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)}{\partial\mathbf{p}}\delta\mathbf{p}-\mathbf{p}^{*\mathsf{T}}\delta\dot{\mathbf{x}}-\dot{\mathbf{x}}^{*\mathsf{T}}\delta\mathbf{p}\bigg]\mathrm{d}t.\end{aligned} \tag{3.25}$$

The total derivative of the third term in (3.21) requires to apply the Leibnitz integral rule since the upper integration bound depends on the perturbation parameter $\varepsilon$. Using the integration rule leads to

$$\begin{aligned}&\varepsilon\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0}\left\{\int_{t_\mathrm{f}^*}^{t_\mathrm{f}^*+\varepsilon\eta_4}\Big[\mathcal{H}(\mathbf{x}^*+\varepsilon\boldsymbol{\eta}_1,\mathbf{u}^*+\varepsilon\boldsymbol{\eta}_2,\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3,t)-(\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3)^\mathsf{T}(\dot{\mathbf{x}}^*+\varepsilon\dot{\boldsymbol{\eta}}_1)\Big]\mathrm{d}t\right\} \\ &=\varepsilon\left\{\int_{t_\mathrm{f}^*}^{t_\mathrm{f}^*+\varepsilon\eta_4}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big[\mathcal{H}(\mathbf{x}^*+\varepsilon\boldsymbol{\eta}_1,\mathbf{u}^*+\varepsilon\boldsymbol{\eta}_2,\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3,t)-(\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3)^\mathsf{T}(\dot{\mathbf{x}}^*+\varepsilon\dot{\boldsymbol{\eta}}_1)\Big]\mathrm{d}t\right\}\bigg|_{\varepsilon=0} \\ &+\varepsilon\left\{\Big[\mathcal{H}(\mathbf{x}^*+\varepsilon\boldsymbol{\eta}_1,\mathbf{u}^*+\varepsilon\boldsymbol{\eta}_2,\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3,t)-(\mathbf{p}^*+\varepsilon\boldsymbol{\eta}_3)^\mathsf{T}(\dot{\mathbf{x}}^*+\varepsilon\dot{\boldsymbol{\eta}}_1)\Big]\Big|_{t_\mathrm{f}^*+\varepsilon\eta_4}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}(t_\mathrm{f}^*+\varepsilon\eta_4)\right\}\bigg|_{\varepsilon=0}.\end{aligned} \tag{3.26}$$

Evaluating the above total derivative at $\varepsilon = 0$ vanishes the integral term and simplifies the total derivative to

$$\begin{aligned}\varepsilon \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0} &\left\{ \int_{t_\mathrm{f}^*}^{t_\mathrm{f}^*+\varepsilon\eta_4} \Big[ \mathcal{H}(\mathbf{x}^* + \varepsilon\boldsymbol{\eta}_1, \mathbf{u}^* + \varepsilon\boldsymbol{\eta}_2, \mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3, t) - (\mathbf{p}^* + \varepsilon\boldsymbol{\eta}_3)^\mathsf{T}(\dot{\mathbf{x}}^* + \varepsilon\dot{\boldsymbol{\eta}}_1) \Big] \mathrm{d}t \right\} \\ &= \big[ \mathcal{H}(\mathbf{x}^*(t_\mathrm{f}^*), \mathbf{u}^*(t_\mathrm{f}^*), \mathbf{p}^*(t_\mathrm{f}^*), t_\mathrm{f}^*) - \mathbf{p}^*(t_\mathrm{f}^*)^\mathsf{T}\dot{\mathbf{x}}^*(t_\mathrm{f}^*) \big] \delta t_\mathrm{f}.\end{aligned} \tag{3.27}$$

Finally, the total derivatives formulated in (3.23), (3.25), and (3.27) can now be used in the variation of the extended cost functional (3.21). Hence, the variation of the extended cost functional reads

$$\begin{aligned}\delta\bar{J} = \int_{t_0}^{t_\mathrm{f}^*} \Bigg[ &\frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{x}} \delta\mathbf{x} + \frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{u}} \delta\mathbf{u} \\ &+ \left( \frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{p}} - \dot{\mathbf{x}}^{*\mathsf{T}} \right) \delta\mathbf{p} - \mathbf{p}^{*\mathsf{T}} \delta\dot{\mathbf{x}} \Bigg] \mathrm{d}t + \frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial \mathbf{x}} \delta\mathbf{x}_\mathrm{f} \\ &+ \left( \frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial t} + \mathcal{H}(\mathbf{x}^*(t_\mathrm{f}^*), \mathbf{u}^*(t_\mathrm{f}^*), \mathbf{p}^*(t_\mathrm{f}^*), t_\mathrm{f}^*) - \mathbf{p}^*(t_\mathrm{f}^*)^\mathsf{T}\dot{\mathbf{x}}^*(t_\mathrm{f}^*) \right) \delta t_\mathrm{f}.\end{aligned} \tag{3.28}$$

The integral of the term $\mathbf{p}^{*\mathsf{T}}\delta\dot{\mathbf{x}}$ is computed by applying integration by parts

$$\begin{aligned}-\int_{t_0}^{t_\mathrm{f}^*} \mathbf{p}^{*\mathsf{T}} \delta\dot{\mathbf{x}} \, \mathrm{d}t &= \int_{t_0}^{t_\mathrm{f}^*} \dot{\mathbf{p}}^{*\mathsf{T}} \delta\mathbf{x} \, \mathrm{d}t - \mathbf{p}^*(t_\mathrm{f}^*)^\mathsf{T} \delta\mathbf{x}(t_\mathrm{f}^*) + \underbrace{\mathbf{p}^*(t_0)^\mathsf{T} \delta\mathbf{x}(t_0)}_{0} \\ &= \int_{t_0}^{t_\mathrm{f}^*} \dot{\mathbf{p}}^{*\mathsf{T}} \delta\mathbf{x} \, \mathrm{d}t - \mathbf{p}^*(t_\mathrm{f}^*)^\mathsf{T} [\delta\mathbf{x}_\mathrm{f} - \dot{\mathbf{x}}^*(t_\mathrm{f}^*) \delta t_\mathrm{f}],\end{aligned} \tag{3.29}$$

where the variation $\delta\mathbf{x}(t_0)$ vanishes due to the fixed initial conditions $\mathbf{x}(t_0) = \bar{\mathbf{x}}_0$, and the variation $\delta\mathbf{x}(t_\mathrm{f}^*)$ is expressed in terms of (3.24). Substituting (3.29) in (3.28) and sorting the variations $\delta\mathbf{x}$, $\delta\mathbf{u}$, $\delta\mathbf{p}$, $\delta\mathbf{x}_\mathrm{f}$, and $\delta t_\mathrm{f}$ results in

$$\begin{aligned}\delta\bar{J} = \int_{t_0}^{t_\mathrm{f}^*} \Bigg[ &\left( \frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{x}} + \dot{\mathbf{p}}^{*\mathsf{T}} \right) \delta\mathbf{x} + \frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{u}} \delta\mathbf{u} \\ &+ \left( \frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{p}} - \dot{\mathbf{x}}^{*\mathsf{T}} \right) \delta\mathbf{p} \Bigg] \mathrm{d}t + \left( \frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial \mathbf{x}} - \mathbf{p}^*(t_\mathrm{f}^*)^\mathsf{T} \right) \delta\mathbf{x}_\mathrm{f} \\ &+ \left( \frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial t} + \mathcal{H}(\mathbf{x}^*(t_\mathrm{f}^*), \mathbf{u}^*(t_\mathrm{f}^*), \mathbf{p}^*(t_\mathrm{f}^*), t_\mathrm{f}^*) \right) \delta t_\mathrm{f}.\end{aligned} \tag{3.30}$$

The extended cost functional becomes stationary when $\delta\bar{J} = 0$ holds for any admissible variations $\delta\mathbf{x}$, $\delta\mathbf{u}$, $\delta\mathbf{p}$, $\delta\mathbf{x}_\mathrm{f}$, and $\delta t_\mathrm{f}$. Hence, the necessary first-order optimality conditions for an unbounded OCP are defined by:

- **Minimum conditions**

$$\mathbf{0} = \left( \frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{u}} \right)^\mathsf{T} \tag{3.31}$$

- **State equations**

$$\dot{\mathbf{x}}^* = \left( \frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{p}} \right)^\mathsf{T} \quad \text{with} \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \tag{3.32}$$

- **Adjoint equations**

$$\dot{\mathbf{p}}^* = -\left(\frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{x}}\right)^\mathsf{T} \tag{3.33}$$

- **Transversality conditions**

$$\begin{aligned} 0 =& \left(\frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial \mathbf{x}} - \mathbf{p}^*(t_\mathrm{f}^*)^\mathsf{T}\right)\delta\mathbf{x}_\mathrm{f} \\ &+ \left(\frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial t} + \mathcal{H}(\mathbf{x}^*(t_\mathrm{f}^*), \mathbf{u}^*(t_\mathrm{f}^*), \mathbf{p}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)\right)\delta t_\mathrm{f} \end{aligned} \tag{3.34}$$

The equations in (3.31) are the necessary minimum conditions for unbounded controls, which is a simplified statement of Pontryagin's minimum principle. The minimum principle for bounded controls is discussed in Section 3.2.2. The first-order differential equations (3.32) and (3.33) are referred to as canonical equations, where the boundary conditions for the adjoint variables are specified at the final time using the transversality conditions in (3.34). Note that the Hamiltonian for autonomous systems has a special behavior. The total time derivative of the Hamiltonian states that

$$\begin{aligned} \frac{\mathrm{d}\mathcal{H}}{\mathrm{d}t} &= \frac{\partial \mathcal{H}}{\partial \mathbf{x}}\dot{\mathbf{x}}^* + \frac{\partial \mathcal{H}}{\partial \mathbf{u}}\dot{\mathbf{u}}^* + \frac{\partial \mathcal{H}}{\partial \mathbf{p}}\dot{\mathbf{p}}^* \\ &= -\dot{\mathbf{p}}^{*\mathsf{T}}\dot{\mathbf{x}}^* + \frac{\partial \mathcal{H}}{\partial \mathbf{u}}\dot{\mathbf{u}}^* + \dot{\mathbf{x}}^{*\mathsf{T}}\dot{\mathbf{p}}^* \\ &= 0, \end{aligned} \tag{3.35}$$

holds for all $t \in [t_0, t_\mathrm{f}]$, where the optimality conditions (3.31)–(3.33) are utilized. Consequently, the solution of an OCP leads to a constant Hamiltonian over time for autonomous systems.

The optimality conditions (3.31)–(3.33) consist of $N_\mathrm{m}$ algebraic equations (minimum principle) and $2N_\mathrm{n}$ first-order differential equations (state and adjoint equations) to determine the optimal control $\mathbf{u}^* \in \mathbb{R}^{N_\mathrm{m}}$, the state variables $\mathbf{x}^* \in \mathbb{R}^{N_\mathrm{n}}$, and the adjoint variables $\mathbf{p}^* \in \mathbb{R}^{N_\mathrm{n}}$ for a fixed final time $t_\mathrm{f}$. If the final time is free, the transversality conditions are used to consider an additional equation. In general, the optimality conditions (3.31)–(3.33) formulate a two-point BVP for the time interval $t \in [t_0, t_\mathrm{f}]$ with $N_\mathrm{n}$ initial conditions $\mathbf{x}(t_0) = \bar{\mathbf{x}}_0$ and an additional set of $N_\mathrm{n}$ or $(N_\mathrm{n}+1)$ final conditions defined by the transversality conditions. The number of final constraints depends on whether the final time is specified or free. In both cases, the actual values of the final conditions depend on the formulation of the OCP, e.g., problems where some state variables are specified or where the cost functional is defined without a Mayer term. The following sections discuss two cases to determine the final conditions based on the transversality conditions (3.34). A comprehensive overview of final conditions for different OCPs is provided in [95].

#### Final conditions for a fixed final time

In this case, the final time is assumed to be fixed, and some state variables are specified at the final time. The fixed final time leads to $\delta t_\mathrm{f} = 0$, i.e., the second term in the transversality conditions (3.34) vanishes. Assuming that the state variables are ordered by $N_\mathrm{s}$ specified states $\mathbf{x}_\mathrm{s}$ followed by $(N_\mathrm{n} - N_\mathrm{s})$ unspecified states $\mathbf{x}_\mathrm{u}$, i.e., $\mathbf{x} = (\mathbf{x}_\mathrm{s}^\mathsf{T}, \mathbf{x}_\mathrm{u}^\mathsf{T})^\mathsf{T}$ with $N_\mathrm{s} \leq N_\mathrm{n}$, the transversality conditions regarding specified and unspecified state variables can be decoupled by

$$0 = \left(\frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial \mathbf{x}_\mathrm{s}} - \mathbf{p}_\mathrm{s}^*(t_\mathrm{f}^*)^\mathsf{T}\right)\delta\mathbf{x}_{\mathrm{f,s}} + \left(\frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial \mathbf{x}_\mathrm{u}} - \mathbf{p}_\mathrm{u}^*(t_\mathrm{f}^*)^\mathsf{T}\right)\delta\mathbf{x}_{\mathrm{f,u}}. \tag{3.36}$$

The specified state variables $\mathbf{x}_{\mathrm{f,s}}$ lead to $\delta\mathbf{x}_{\mathrm{f,s}} = \mathbf{0}$, while the variation of the unspecified state variables $\delta\mathbf{x}_{\mathrm{f,u}}$ is free. Hence, the first term in (3.36) vanishes, and the second term must be defined in such a way that the transversality conditions are satisfied. To this end, the necessary $N_{\mathrm{n}}$ final conditions are defined by

$$\mathbf{x}_{\mathrm{s}}^*(t_{\mathrm{f}}^*) = \mathbf{x}_{\mathrm{f,s}}, \tag{3.37}$$

$$\mathbf{p}_{\mathrm{u}}^*(t_{\mathrm{f}}^*) = \left(\frac{\partial E(\mathbf{x}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*)}{\partial \mathbf{x}_{\mathrm{u}}}\right)^{\mathsf{T}}. \tag{3.38}$$

#### Final conditions for a free final time

In this case, the final time is assumed to be free, and some state variables are specified at the final time. Hence, the variation of the final time does not vanish, and the transversality conditions (3.34) can be decoupled as in the previous case by

$$\begin{aligned} 0 =& \left(\frac{\partial E(\mathbf{x}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*)}{\partial \mathbf{x}_{\mathrm{s}}} - \mathbf{p}_{\mathrm{s}}^*(t_{\mathrm{f}}^*)^{\mathsf{T}}\right)\delta\mathbf{x}_{\mathrm{f,s}} + \left(\frac{\partial E(\mathbf{x}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*)}{\partial \mathbf{x}_{\mathrm{u}}} - \mathbf{p}_{\mathrm{u}}^*(t_{\mathrm{f}}^*)^{\mathsf{T}}\right)\delta\mathbf{x}_{\mathrm{f,u}} \\ &+ \left(\frac{\partial E(\mathbf{x}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*)}{\partial t} + \mathcal{H}(\mathbf{x}^*(t_{\mathrm{f}}^*), \mathbf{u}^*(t_{\mathrm{f}}^*), \mathbf{p}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*)\right)\delta t_{\mathrm{f}}. \end{aligned} \tag{3.39}$$

The specified state variables $\mathbf{x}_{\mathrm{f,s}}$ lead to $\delta\mathbf{x}_{\mathrm{f,s}} = \mathbf{0}$, while the variation of the unspecified state variables $\delta\mathbf{x}_{\mathrm{f,u}}$ and the variation of the final time $\delta t_{\mathrm{f}}$ are free. Hence, the first term in (3.39) vanishes, and the second and third terms must be defined in such a way that the transversality conditions are satisfied. To this end, the necessary $(N_{\mathrm{n}}+1)$ final conditions are defined by

$$\mathbf{x}_{\mathrm{s}}^*(t_{\mathrm{f}}^*) = \mathbf{x}_{\mathrm{f,s}}, \tag{3.40}$$

$$\mathbf{p}_{\mathrm{u}}^*(t_{\mathrm{f}}^*) = \left(\frac{\partial E(\mathbf{x}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*)}{\partial \mathbf{x}_{\mathrm{u}}}\right)^{\mathsf{T}}, \tag{3.41}$$

$$\mathcal{H}(\mathbf{x}^*(t_{\mathrm{f}}^*), \mathbf{u}^*(t_{\mathrm{f}}^*), \mathbf{p}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*) = -\frac{\partial E(\mathbf{x}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*)}{\partial t}. \tag{3.42}$$

When the OCP is formulated without a Mayer term, the $(N_{\mathrm{n}}+1)$ final conditions are simplified to

$$\mathbf{x}_{\mathrm{s}}^*(t_{\mathrm{f}}^*) = \mathbf{x}_{\mathrm{f,s}}, \tag{3.43}$$

$$\mathbf{p}_{\mathrm{u}}^*(t_{\mathrm{f}}^*) = \mathbf{0}, \tag{3.44}$$

$$\mathcal{H}(\mathbf{x}^*(t_{\mathrm{f}}^*), \mathbf{u}^*(t_{\mathrm{f}}^*), \mathbf{p}^*(t_{\mathrm{f}}^*), t_{\mathrm{f}}^*) = 0. \tag{3.45}$$

The final condition (3.45) states that the Hamiltonian is zero at the final time, and using (3.35), it holds that the Hamiltonian for autonomous systems is zero for all $t \in [t_0, t_{\mathrm{f}}^*]$.

### 3.2.2 Bounded Optimal Control Problems

The previous section assumes that the control $\mathbf{u}$ and states $\mathbf{x}$ are not bounded. However, such boundaries commonly occur in practical problems, e.g., if a robot should be manipulated from an initial state to a final state without violating the maximum permissible torque of the drives. Thus, the boundaries must be considered to derive the necessary optimality conditions. This section focuses only on the boundaries of the control variables. For boundaries regarding the state variables, the reader is referred to [95].

The minimum conditions formulated in (3.31) are derived for any arbitrary variation of the control $\delta\mathbf{u}$. This statement does not hold for bounded controls because the variation

of the control is not arbitrary for bounded problems; see Fig. 3.3 for an extremal control and (in)admissible control variations. However, the optimality conditions for unbounded controls (3.32)–(3.34) remain also valid for bounded controls. Assuming that the later men-

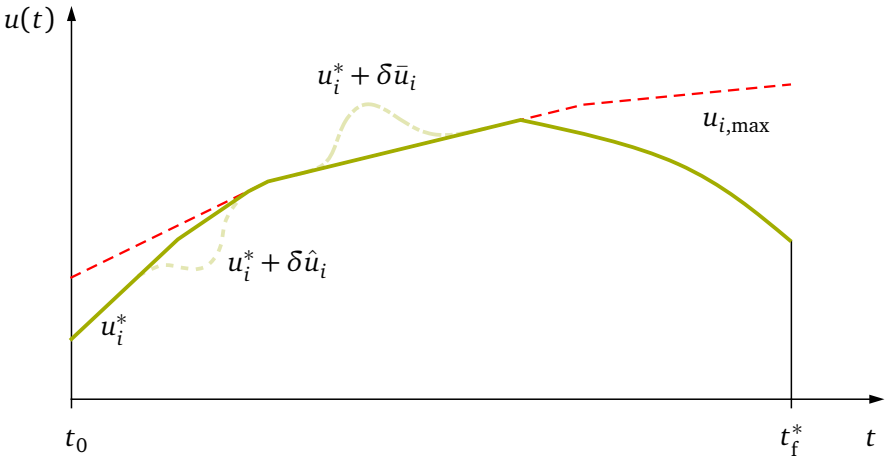



**Figure 3.3:** An extremal control $u_i^*$ and a perturbation by an admissible variation $\delta\hat{u}_i$ and an inadmissible variation $\delta\bar{u}_i$ [95]

tioned optimality conditions are satisfied, the variation of the extended cost functional (3.30) simplifies to

$$\delta\bar{J} = \int_{t_0}^{t_f^*} \frac{\partial\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)}{\partial\mathbf{u}}\delta\mathbf{u}\,\mathrm{d}t. \tag{3.46}$$

The partial derivative of the Hamiltonian with respect to the control can be approximated by a Taylor series expansion as

$$\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*+\delta\mathbf{u},\mathbf{p}^*,t) \approx \mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t) + \frac{\partial\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)}{\partial\mathbf{u}}\delta\mathbf{u}, \tag{3.47}$$

caused by a change of the control. Substituting (3.47) into (3.46) results in

$$\delta\bar{J} = \int_{t_0}^{t_f^*} \Big[\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*+\delta\mathbf{u},\mathbf{p}^*,t) - \mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)\Big]\mathrm{d}t. \tag{3.48}$$

The necessary first-order optimality condition for a bounded control can be derived from (3.48) with $\delta\bar{J} \geq 0$. Thus, the condition

$$\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*+\delta\mathbf{u},\mathbf{p}^*,t) \geq \mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t) \tag{3.49}$$

must hold for any admissible variation $\delta\mathbf{u}$. The condition (3.49) is the well-known Pontryagin's minimum principle [123] and states that the Hamiltonian to an optimal solution must take a minimum. To summarize, the necessary first-order optimality conditions for a bounded OCP are defined by:

- **Pontryagin's minimum principle**

$$\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*+\delta\mathbf{u},\mathbf{p}^*,t) \geq \mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t) \tag{3.50}$$

- **State equations**

$$\dot{\mathbf{x}}^* = \left(\frac{\partial\mathcal{H}(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)}{\partial\mathbf{p}}\right)^\mathsf{T} \quad \text{with} \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \tag{3.51}$$

- **Adjoint equations**

$$\dot{\mathbf{p}}^* = -\left(\frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{x}}\right)^{\mathsf{T}} \tag{3.52}$$

- **Transversality conditions**

$$\begin{aligned} 0 = &\left(\frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial \mathbf{x}} - \mathbf{p}^*(t_\mathrm{f}^*)^{\mathsf{T}}\right)\delta\mathbf{x}_\mathrm{f} \\ &+ \left(\frac{\partial E(\mathbf{x}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)}{\partial t} + \mathcal{H}(\mathbf{x}^*(t_\mathrm{f}^*), \mathbf{u}^*(t_\mathrm{f}^*), \mathbf{p}^*(t_\mathrm{f}^*), t_\mathrm{f}^*)\right)\delta t_\mathrm{f} \end{aligned} \tag{3.53}$$

#### Time-Optimal Control Problems

An important application of OCPs is to drive a mechanical system from an initial state to a final state in the shortest possible time. Such a problem is classified as a time-optimal control problem and has been studied by various authors [30, 53, 95, 100]. Time-optimal control problems aim to compute a control $\mathbf{u} = \mathbf{u}^*$, state variables $\mathbf{x} = \mathbf{x}^*$, and a final time $t_\mathrm{f} = t_\mathrm{f}^*$, such that the scalar cost functional $J$ becomes stationary

$$\min_{\mathbf{x},\mathbf{u},t_\mathrm{f}} J = \int_{t_0}^{t_\mathrm{f}} 1\,\mathrm{d}t = t_\mathrm{f} - t_0 \tag{3.54}$$

s.t.

$$\boldsymbol{\phi}(\mathbf{x}(t_\mathrm{f}), t_\mathrm{f}) = \mathbf{0} \tag{3.55}$$

$$u_{i,\min} \le u_i(t) \le u_{i,\max} \quad \text{with} \quad i = 1, \ldots, N_\mathrm{m} \tag{3.56}$$

$$\mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \tag{3.57}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{3.58}$$

concerning equality constraints and bounds. The time-optimal control problem is subject to final constraints (3.55), lower and upper bounds of the control variables (3.56), while satisfying the state equations with the given initial state variables $\bar{\mathbf{x}}_0$.

In mechanics, the control $\mathbf{u}$ usually appears linear in the state equations. Therefore, the first-order differential equation (3.58) can be formulated as

$$\dot{\mathbf{x}}(t) = \mathbf{a}(\mathbf{x}(t)) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t), \tag{3.59}$$

where $\mathbf{a} : \mathbb{R}^{N_\mathrm{n}} \to \mathbb{R}^{N_\mathrm{n}}$ is the system vector and $\mathbf{B} : \mathbb{R}^{N_\mathrm{n}} \to \mathbb{R}^{N_\mathrm{n} \times N_\mathrm{m}}$ is the input matrix which may depend on the state variables. The Hamiltonian to the time-optimal control problem is formulated as introduced in (3.12) by

$$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t)) = 1 + \mathbf{p}^{\mathsf{T}}(t)\big[\mathbf{a}(\mathbf{x}(t)) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t)\big]. \tag{3.60}$$

Following Pontyagin's minimum principle (3.50), the optimality condition

$$1 + \mathbf{p}^{*\mathsf{T}}(t)\big[\mathbf{a}(\mathbf{x}^*(t)) + \mathbf{B}(\mathbf{x}^*(t))\mathbf{u}(t)\big] \ge 1 + \mathbf{p}^{*\mathsf{T}}(t)\big[\mathbf{a}(\mathbf{x}^*(t)) + \mathbf{B}(\mathbf{x}^*(t))\mathbf{u}^*(t)\big] \tag{3.61}$$

must hold for all admissible control functions $\mathbf{u}$ and for all $t \in [t_0, t_\mathrm{f}^*]$. The optimality condition can be reformulated in index notation to

$$\sum_{i=1}^{N_\mathrm{m}} \underbrace{\mathbf{p}^{*\mathsf{T}}(t)\mathbf{b}_i(\mathbf{x}^*(t))}_{h_i(\mathbf{x}^*(t), \mathbf{p}^*(t))}\big[u_i(t) - u_i^*(t)\big] \ge 0, \tag{3.62}$$

where $\mathbf{b}_i$ denotes the $i$-th column of the matrix $\mathbf{B}$ and $h_i : \mathbb{R}^{N_\mathrm{n}} \times \mathbb{R}^{N_\mathrm{n}} \to \mathbb{R}$ is the so-called switching function. Assuming that all control variables $u_i$ are linearly independent, the optimality condition in (3.62) yields the optimal control variables. Satisfying (3.62) for all admissible $u_i$, the control variable $u_i^*$ must lie on the minimum bound if the switching function $h_i$ is positive, the control variable $u_i^*$ must lie on the maximum bound if the switching function $h_i$ is negative. When the switching function becomes zero, the control variable $u_i^*$ can not be determined by Prontryagin's minimum principle. This case is called singular interval; the interested reader is referred to [95] for further details. To summarize, the optimality conditions for a time-optimal control problem are defined by

$$u_i^*(t) := \begin{cases} u_{i,\max} & \text{for} \quad h_i(\mathbf{x}^*(t), \mathbf{p}^*(t)) < 0 \\ u_{i,\min} & \text{for} \quad h_i(\mathbf{x}^*(t), \mathbf{p}^*(t)) > 0 \\ \text{singular} & \text{for} \quad h_i(\mathbf{x}^*(t), \mathbf{p}^*(t)) = 0 \end{cases} . \tag{3.63}$$

Following [95], the optimality condition for time-optimal control problems is referred to

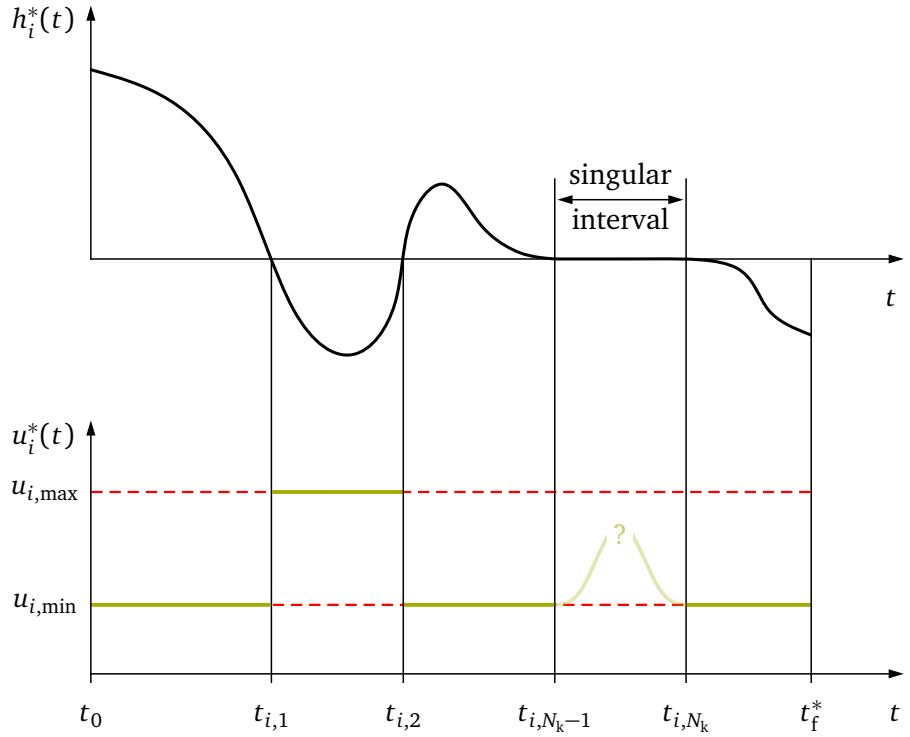



**Figure 3.4:** Switching function $h_i$ and corresponding control $u_i$ for a time-optimal control problem [95]

as the bang-bang principle and can be interpreted as maximizing the control effort for all $t \in [t_0, t_\mathrm{f}^*]$. A graphical representation of the bang-bang principle is shown in Fig. 3.4 with $N_\mathrm{k}$ switching points. For linear systems, the number of switching points can be computed using the eigenvalues of the matrix $\mathbf{A} \in \mathbb{R}^{N_\mathrm{n} \times N_\mathrm{n}}$, which results from reformulating the system vector as $\mathbf{a}(\mathbf{x}(t)) = \mathbf{A}\mathbf{x}(t)$. If all of the eigenvalues of $\mathbf{A}$ are real and nonpositive, then each control $u_i$ consists of $N_\mathrm{k} \leq N_\mathrm{n} - 1$ switching points [95]. For nonlinear systems, the number of switching points can only be determined using an (approximated) solution of the time-optimal control problem. However, assuming no singular interval exists and the number of switching functions is known, the time optimal-control problem can be transformed into a parameter optimization problem regarding the final time and the time points where a switching point occurs. The resulting NLP problem, including the equality constraints (3.55) regarding the final states, can be solved with different methods, e.g., using an approach as shown in Section 3.3 or as presented in [53].

### 3.2.3 Solution Approaches

The necessary first-order optimality conditions provide the basis for computing an optimal solution of the OCP using indirect methods. These optimality conditions formulate a two-point BVP, which usually can not be solved analytically to obtain the optimal control law. Instead, iterative numerical approaches have been developed to compute an optimal solution for the OCP. The most common solution approaches in the class of indirect methods are: indirect multiple shooting, indirect collocation, and gradient-based methods. The first two methods focus on solving the two-point BVP derived from the first-order optimality conditions. Various solution strategies based on the principle concept of multiple shooting have been developed, e.g., by Keller [90], Bulirsch [31], and Bock [21]. An introduction to indirect multiple shooting is given by Stoer and Bulirsch [149]. For collocation methods, relevant developments are provided, for example, by Ascher et al. [5] and Dickmanns and Well [45]. The general solution strategy for an unbounded OCP using an indirect multiple shooting or a collocation method follows the procedure outlined below [95]:

1. The control $\mathbf{u}$ is expressed as a function of the state variables $\mathbf{x}$, the adjoint variables $\mathbf{p}$, and the time $t$ by

$$\mathbf{u} = \boldsymbol{\psi}(\mathbf{x}, \mathbf{p}, t). \tag{3.64}$$

   This expression is obtained by exploiting the minimum conditions (3.31). If the control appears linear in the Hamiltonian, the minimum conditions provide no information on the control. However, since the minimum conditions must hold for an optimal solution, the $r$-th total time derivative of the minimum conditions can be used to obtain an expression of the control. The time derivative is applied $r$-th times until the expression

$$\frac{\mathrm{d}^r}{\mathrm{d}t^r}\left(\frac{\partial \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}^*, t)}{\partial \mathbf{u}}\right)^\mathsf{T} = \mathbf{0} \tag{3.65}$$

   is explicitly dependent on the control $\mathbf{u}$. Thus, an expression in the form of (3.64) can be determined for the control.

2. Eliminating the control $\mathbf{u}$ from the canonical equations (3.32) and (3.33) using the expression in (3.64) leads to the differential equations

$$\dot{\mathbf{x}} = \left(\frac{\partial \mathcal{H}(\mathbf{x}, \boldsymbol{\psi}(\mathbf{x}, \mathbf{p}, t), \mathbf{p}, t)}{\partial \mathbf{p}}\right)^\mathsf{T} \quad \text{with} \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \tag{3.66}$$

$$\dot{\mathbf{p}} = -\left(\frac{\partial \mathcal{H}(\mathbf{x}, \boldsymbol{\psi}(\mathbf{x}, \mathbf{p}, t), \mathbf{p}, t)}{\partial \mathbf{x}}\right)^\mathsf{T}, \tag{3.67}$$

   which formulate a two-point BVP with the appropriate boundary conditions, e.g., using the conditions provided in Section 3.2.1 or 3.2.1. The modified two-point BVP does not depend on the control $\mathbf{u}$ and can be solved numerically by shooting or collocation methods.

3. Solve the modified two-point BVP to obtain the state variables $\mathbf{x}^*$, the adjoint variables $\mathbf{p}^*$, and the final time $t_\mathrm{f}^*$ by using a numerical solver, e.g., the MATLAB routine bvp4c, which is an implementation based on the collocation approach described by Kierzenka and Shampine [93].

4. The evolution of the optimal control $\mathbf{u}^*$ is computed using the expression in (3.64).

Using the above procedure to solve an OCP requires a numerical approach for solving the two-point BVP, which solution is usually difficult to compute. An alternative to solving the two-point BVP directly is provided by gradient-based methods. The key idea is to minimize the cost functional in an iterative process by using information on gradients. The gradients of the cost functional are computed in each iteration in a sequential process by solving the state equations forward in time, followed by solving the adjoint equations backward in time. Various gradient-based solution strategies are available to obtain the optimal control law, e.g., the steepest descent method, which has been developed independently by Kelley [91] and Bryson et al. [29], or a quasi-Newton method where the Hessian matrix is approximated with gradients from previous iterations. The basic solution strategy for an unbounded OCP with a fixed final time $t_\mathrm{f}$ and unspecified state variables at the final time follows the procedure outlined below [95]:

1. Select an initial control $\mathbf{u}_0$ and set the number of iterations $k$ to zero.

2. Solve the state equations (3.32) for the $k$-th iteration

$$\dot{\mathbf{x}}_k = \left( \frac{\partial \mathcal{H}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k, t)}{\partial \mathbf{p}} \right)^\mathsf{T} \quad \text{with} \quad \mathbf{x}_k(t_0) = \bar{\mathbf{x}}_0 \tag{3.68}$$

   forward in time using an ODE solver. The evolution of the state variables must be stored in memory for the subsequent backward time integration to compute the evolution of the adjoint variables.

3. Solve the adjoint equations (3.33) for the $k$-th iteration

$$\dot{\mathbf{p}}_k = -\left( \frac{\partial \mathcal{H}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k, t)}{\partial \mathbf{x}} \right)^\mathsf{T} \quad \text{with} \quad \mathbf{p}_k(t_\mathrm{f}) = \left( \frac{\partial E(\mathbf{x}_k(t_\mathrm{f}), t_\mathrm{f})}{\partial \mathbf{x}} \right)^\mathsf{T} \tag{3.69}$$

   backward in time using an ODE solver. Final conditions of the adjoint variables for problems with unspecified state variables at the final time are given by (3.38). The terminal-value problem can be transformed into an IVP using a time transformation as described in Section 2.4.2.

4. An optimal solution for the $k$-th iteration is found if the norm of the minimum conditions (3.31) satisfies the condition

$$\left\| \left( \frac{\partial \mathcal{H}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k, t)}{\partial \mathbf{u}} \right)^\mathsf{T} \right\| \leq \varepsilon, \tag{3.70}$$

   where the scalar $\varepsilon > 0$ is an a priori defined optimality tolerance.

5. If the termination criterion (3.70) is not satisfied, an iterative process updates the control until the termination criterion is satisfied. The control from iteration $k$ to iteration $k+1$ is updated by

$$\mathbf{u}_{k+1} \coloneqq \mathbf{u}_k + \alpha_k \mathbf{d}_k^\mathrm{u}, \tag{3.71}$$

   where $\mathbf{d}_k^\mathrm{u}$ is the search direction and $\alpha_k > 0$ is a sufficiently small step size for the $k$-th iteration. The search direction of the control is computed by using the variation of the cost functional in (3.30). Since the state equations (3.68) and adjoint equations (3.69) are satisfied, the variation of the extended cost functional simplifies to

$$\delta \bar{J} = \int_{t_0}^{t_\mathrm{f}} \left[ \frac{\partial \mathcal{H}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k, t)}{\partial \mathbf{u}} \delta \mathbf{u} \right] \mathrm{d}t. \tag{3.72}$$

The largest possible change of $\delta\bar{J}$ is obtained if the variation of the control $\delta\mathbf{u}$ points into the direction of the derivative of the Hamiltonian with respect to the control. Thus, the term $\partial\mathcal{H}/\partial\mathbf{u}$ can be interpreted as the gradient of the extended cost functional, and the steepest descent method uses the negative gradient to define the search direction

$$\mathbf{d}_k^{\mathrm{u}} := -\left(\frac{\partial\mathcal{H}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k, t)}{\partial\mathbf{u}}\right)^{\mathsf{T}}, \tag{3.73}$$

which results in a reduction of the extended cost functional. The control $\mathbf{u}_{k+1}$ is then used to continue the iterative process with Step 2 until the termination criterion (3.70) is satisfied.

The basic version of the steepest descent method presented above can be extended to handle an OCP with a free final time regarding final constraints [53]. The main advantages of the steepest descent method are that (1) the control must not be expressed from the minimum conditions, (2) an initial guess of the adjoint variables is not required, and (3) the method is robust concerning the initial guess of the control [14]. Contrarily, the main disadvantage of the method is the rather slow convergence, which can be improved by applying a quasi-Newton method.

Indirect methods provide highly accurate solutions for the OCP under investigation. However, using such a method requires significant knowledge and expertise in control theory. As pointed out by Betts [14], three main challenges occur in practice using indirect methods:

- Partial derivatives of the Hamiltonian may be complex to compute and need to be updated for changes in the problem definition.
- Inequality constraints on the state variables are hard to handle. An a priori estimation of the sequence of active inequality constraints is required, which is difficult to specify.
- Basic methods to solve the two-point BVP are sensitive concerning initial guesses for the state variables, control variables, adjoint variables, and the final time. An appropriate initial guess for the adjoint variables is particularly difficult to define. The sensitivity to initial guesses can be reduced by using gradient-based methods or homotopy methods [6].

The practical challenges associated with indirect methods motivate the introduction of an alternative approach to solve the OCP using so-called direct methods. The following section introduces the main concepts of direct methods. Therein, approaches to reformulate the OCP regarding an NLP problem, necessary optimality conditions for a feasible solution, and numerical techniques to solve the NLP problem are discussed.

## 3.3 Direct Methods

The basic idea of direct methods is to transcribe the original infinite-dimensional OCP defined in (3.2)–(3.7) into a finite-dimensional NLP problem. Therein, the OCP is first discretized, and the resulting NLP problem is then treated by well-known classical optimization methods, e.g., the SQP method or the IP method. Due to the sequence of discretizing followed by optimizing, direct methods are referred to as first discretize, then optimize approaches. All direct methods parameterize the control $\mathbf{u}$, but they differ in the way how to handle the state variables $\mathbf{x}$ [46]. The transcription of the infinite-dimensional cost functional (3.2), the equality constraints (3.4), and the inequality constraints (3.5) into their finite-dimensional counterparts is straightforward. The cost functional is approximated by numerical integration, e.g.,

using a quadrature rule, and the (in)equality constraints are enforced to hold on a defined time grid. Considering the state equations (3.7) is more advanced and can be accomplished differently. Generally, there are two main approaches for direct methods [126]:

- **Sequential approach**
  The evolution of the state variables is computed in each iteration of the optimization procedure by an embedded numerical time integration scheme. Thus, the state variables are considered as an implicit function of the initial state, the parameters, and the control parameterization employed. In this approach, the optimization iterations and time integration proceed sequentially until an optimal solution is found.

- **Simultaneous approach**
  The evolution of the state variables and the control are parameterized. Along with the control parameterization, the parameterized state variables are considered to serve as additional optimization variables. Thus, the state equations can be violated during the optimization, but they must be satisfied in the optimal solution due to equality constraints on the state variables. In this approach, the optimization iterations and the time integration proceed simultaneously until an optimal solution is found.

Both approaches differ in transcribing the infinite-dimensional OCP into a finite-dimensional NLP problem. The most common direct methods associated with the above approaches are: direct single shooting (a pure sequential approach), direct collocation (a pure simultaneous approach), and direct multiple shooting (a mixture of a pure sequential approach and a pure simultaneous approach) [116]. These methods are discussed in the following sections.

In direct methods, the time horizon $t \in [t_0, t_\mathrm{f}]$ is split up into $M$ intervals defined by the time grid

$$\mathbb{G}_\mathrm{u} = \{t_0 < \ldots < t_k < \ldots < t_{M-1} < t_M = t_\mathrm{f}\}, \tag{3.74}$$

and the control $\mathbf{u}$ is usually parameterized on the grid $\mathbb{G}_\mathrm{u}$ by a discrete set of variables

$$\mathbf{w}_\mathrm{u} = \left(\mathbf{u}_0^\mathsf{T}, \mathbf{u}_1^\mathsf{T}, \ldots, \mathbf{u}_M^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^{N_\mathrm{u}}, \tag{3.75}$$

with $N_\mathrm{u} = N_\mathrm{m}(M+1)$, and $N_\mathrm{m}$ denotes the number of control variables. Therein, the abbreviation $(\cdot)_j := (\cdot)(t_j)$ is used to denote an evaluation of a time-dependent function $(\cdot)$ at a particular time $t = t_j$. For example, the parameterization $\mathbf{w}_\mathrm{u}$ can represent control values for a piecewise constant parameterization or coefficients of polynomials. However, the general form of the control parameterization $\mathbf{w}_\mathrm{u}$ is used in the following sections for transcribing the infinite-dimensional OCP into a finite-dimensional NLP problem.

### 3.3.1 Direct Single Shooting

The concept of direct single shooting has been introduced by Hicks and Ray [85]. It is probably the most intuitive approach to transcribe the infinite-dimensional OCP into a finite-dimensional NLP problem. The method is based on approximating the time-dependent control $\mathbf{u}$ by a suitable parameterization $\mathbf{w}_\mathrm{u}$. The evolution of the state variables $\mathbf{x}$ is computed by numerical time integration, starting at the given initial state variables $\bar{\mathbf{x}}_0$. Therefore, the state constraints of the infinite-dimensional OCP in (3.6) and (3.7) are satisfied by using an embedded ODE solver in each iteration. Direct single shooting is a sequential approach since a numerical time integration is performed in each iteration of the optimization.

The direct single shooting approach transcribes the OCP defined in (3.2)–(3.7) into the following finite-dimensional NLP problem formulated as

$$\min_{\mathbf{w}_\mathrm{u},\xi,t_\mathrm{f}} J = \int_{t_0}^{t_\mathrm{f}} L(\mathbf{x},\mathbf{u},\xi,t)\,\mathrm{d}t + E(\mathbf{x}_M,t_M) \tag{3.76}$$

s.t.

$$\boldsymbol{\phi}(\mathbf{x}_M,t_M) = \mathbf{0} \tag{3.77}$$

$$\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k) = \mathbf{0} \quad k \in \{0,\ldots,M\} \tag{3.78}$$

$$\mathbf{h}(\mathbf{x}_k,\mathbf{u}_k) \leq \mathbf{0} \quad k \in \{0,\ldots,M\} \tag{3.79}$$

where the set of control variables $\mathbf{w}_\mathrm{u}$, the parameters $\xi$, and the final time $t_\mathrm{f}$ serve as optimization variables. The optimal values of these variables are computed by using a numerical optimization method, e.g., as described in Sections 3.3.6 and 3.3.7. The (in)equality constraints (3.78) and (3.79) are typically enforced to hold on the time grid of the control parameterization $\mathbb{G}_\mathrm{u}$.

Due to its simplicity, the direct single shooting method is often used to solve OCPs arising in engineering. The main advantages of the approach can be summarized as: (1) state-of-the-art ODE solvers can be used to compute the evolution of the state variables, and (2) only the control is discretized (and not the state variables) leading to an NLP problem that consists of few optimization variables even for large ODE systems. Contrarily, the main disadvantages are: (1) the evolution of the state variables can not be used directly in the initialization of the NLP problem, and (2) unstable systems are difficult to handle, especially for problems with a long time horizon [46].

Figure 3.5 visualizes the evolution of the control variable $u$ (–) and the state variable $x$ (–) in the direct single shooting approach. Therein, the control is parameterized with $(M+1)$ grid nodes $u_k$ (●) whereas values between the grid nodes are evaluated by interpolation. The evolution of the state variable is computed by numerical time integration, where the value of the last integration point (○) has to fulfill the a priori defined final value $\bar{x}_\mathrm{f}$ (×) for an optimal solution of the NLP problem.

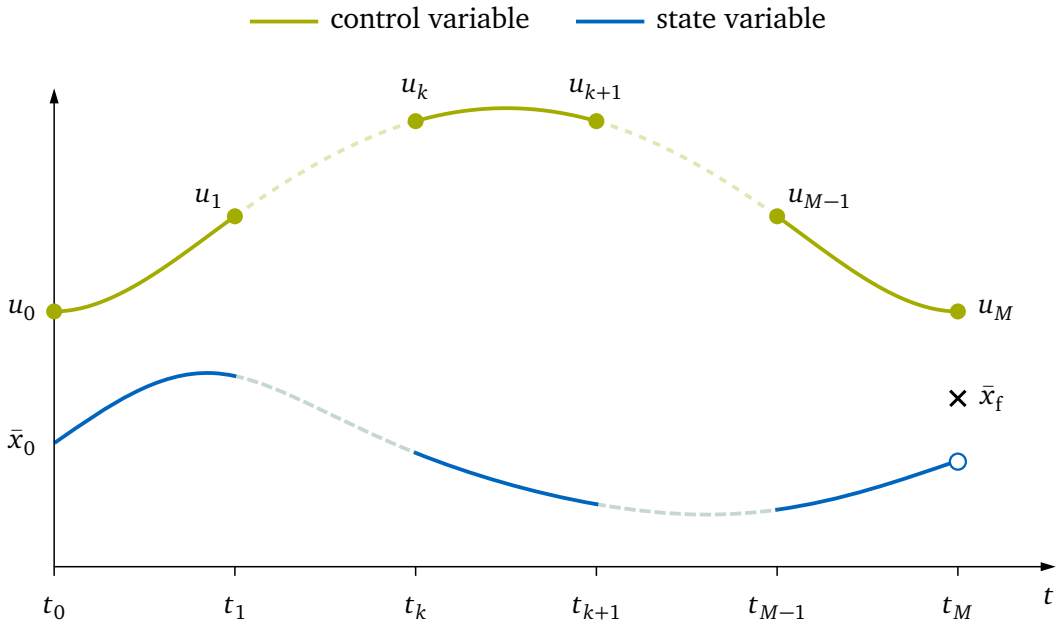



**Figure 3.5:** Evolution of the control and state variable in the direct single shooting method

### 3.3.2 Direct Multiple Shooting

The direct multiple shooting method has been introduced by Bock and Plitt [22] and is based on the general concept of the direct single shooting method. The key idea of direct multiple shooting is to divide the time horizon into $K$ so-called shooting intervals defined by the time grid

$$\mathbb{G}_{\mathrm{x}_0} = \{t_0 < \ldots < t_k < \ldots < t_{K-1} < t_K = t_\mathrm{f}\}, \tag{3.80}$$

where the evolution of the state variables is computed in each shooting interval independently by numerical time integration starting at artificial initial conditions $\mathbf{x}_k$, $k \in \{0, \ldots, K\}$. These artificial initial conditions serve as additional optimization variables compared to the direct single shooting method. Positive influence on the sparsity of the optimization problem can be achieved if the artificial initial conditions are defined on the same time grid as the control parameterization, i.e., $\mathbb{G}_{\mathrm{x}_0} = \mathbb{G}_\mathrm{u}$ with $K = M$. The artificial initial conditions are concatenated in the vector

$$\mathbf{w}_{\mathrm{x}_0} = \left(\mathbf{x}_0^\mathsf{T}, \mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_M^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^{N_{\mathrm{x}_0}}, \tag{3.81}$$

where the size of the vector is $N_{\mathrm{x}_0} = N_\mathrm{n}(M+1)$, and $N_\mathrm{n}$ denotes the number of state variables. Both the artificial initial conditions $\mathbf{w}_{\mathrm{x}_0}$ and the control parameterization $\mathbf{w}_\mathrm{u}$ influence the evolution of the state variables within the shooting intervals. State variables $\mathbf{x}$ within the shooting intervals are computed by numerical time integration, but they may not guarantee continuity at the boundaries of the shooting intervals due to the artificial initial conditions $\mathbf{w}_{\mathrm{x}_0}$. Discontinuities over the shooting intervals are referred to as multiple shooting defects, where equality constraints are considered in the NLP problem formulation to enforce continuity over the shooting intervals for a converged solution.

The direct multiple shooting approach transcribes the OCP defined in (3.2)–(3.7) into the following finite-dimensional NLP problem formulated as

$$\min_{\mathbf{w}_\mathrm{u}, \mathbf{w}_{\mathrm{x}_0}, \boldsymbol{\xi}, t_\mathrm{f}} J = \sum_{k=0}^{M-1} \int_{t_k}^{t_{k+1}} L(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}, t)\,\mathrm{d}t + E(\mathbf{x}_M, t_M) \tag{3.82}$$

s.t.

$$\boldsymbol{\phi}(\mathbf{x}_M, t_M) = \mathbf{0} \tag{3.83}$$

$$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0} \quad k \in \{0, \ldots, M\} \tag{3.84}$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{0} \quad k \in \{0, \ldots, M\} \tag{3.85}$$

$$\bar{\mathbf{x}}_0 = \mathbf{x}_0 \tag{3.86}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi})\,\mathrm{d}t \quad k \in \{0, \ldots, M-1\} \tag{3.87}$$

where the set of control variables $\mathbf{w}_\mathrm{u}$, the artificial initial conditions $\mathbf{w}_{\mathrm{x}_0}$, the parameters $\boldsymbol{\xi}$, and the final time $t_\mathrm{f}$ serve as optimization variables. The above NLP problem is equivalent to the direct single shooting NLP problem but contains artificial initial conditions $\mathbf{w}_{\mathrm{x}_0}$ which serve as optimization variables [46]. Therein, the equality constraints (3.87) enforce continuity of the state variables over the shooting interval boundaries, while the equality constraints (3.86) ensure that the given initial conditions $\bar{\mathbf{x}}_0$ are fulfilled. Note that the variables $\mathbf{x}_0$ and $\mathbf{x}_M$ could be eliminated from the set of optimization variables by starting the time integration of the first shooting interval in (3.87) at $\bar{\mathbf{x}}_0$ and by enforcing that the last integration point of the last shooting interval coincide with $\bar{\mathbf{x}}_\mathrm{f}$, respectively. However, considering all $(M+1)$ artificial initial conditions positively influences the sparsity of the optimization problem.

Similar to the direct single shooting method, state-of-the-art ODE solvers can be used to compute the evolution of the state variables. But the direct multiple shooting method allows a shorter interval of integration by introducing shooting intervals, and therefore, numerical time integration errors are reduced, especially when the time horizon $t \in [t_0, t_f]$ is long. Note that the time integration within a shooting interval is decoupled from its predecessors, enabling simultaneous computation of the state variables. Thus, the method is well suited for parallel computing [15]. In addition, the state variables at the boundaries of the shooting intervals can be directly initialized using the artificial initial conditions, which increases the robustness of handling unstable systems [47]. Compared to the direct single shooting method, the dimension of the direct multiple shooting NLP problem is significantly increased. Nonetheless, the structure of the optimization problem is sparse, which can be exploited using sparse solvers [39].

Figure 3.6 visualizes the evolution of the control variable $u$ (–) and the state variable $x$ (–) in the direct multiple shooting approach. Therein, the control is parameterized with $(M+1)$ grid nodes $u_k$ (●) whereas values between the grid nodes are evaluated by interpolation. In addition, each shooting interval consists of an artificial initial condition $x_k$ (●). The evolution of the state variable within the shooting intervals is computed by numerical time integration. The value of the last integration point (○) of a shooting interval has to coincide with the artificial condition of the subsequent shooting interval to ensure continuity over the time horizon. Moreover, the artificial initial conditions $x_0$ and $x_M$ have to coincide with the corresponding a priori given state variables (×) for an optimal solution of the NLP problem.

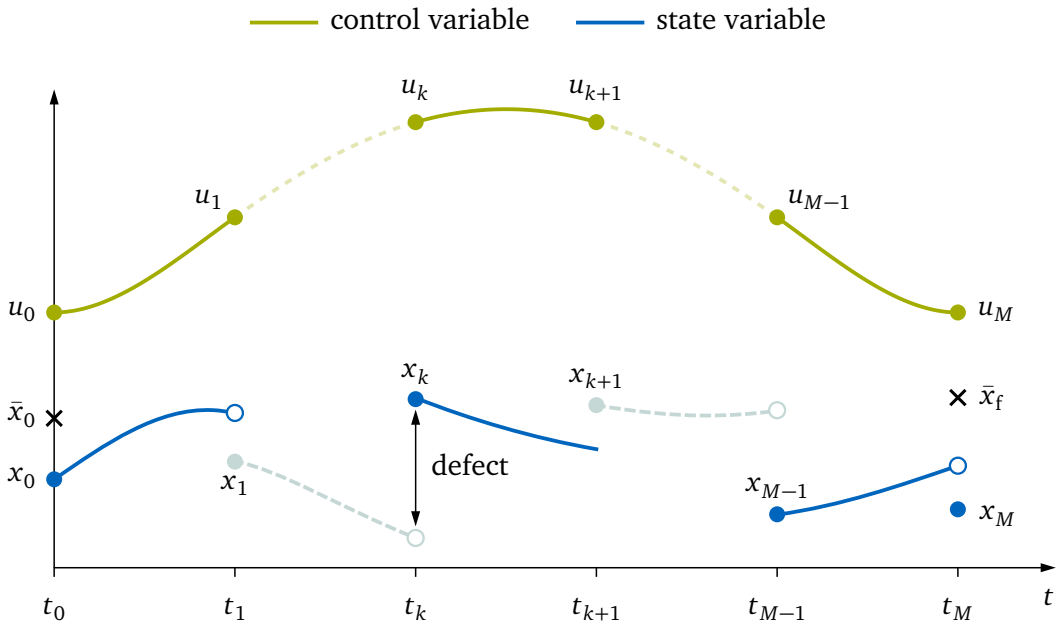



**Figure 3.6:** Evolution of the control and state variable in the direct multiple shooting method

### 3.3.3 Direct Collocation

The key idea of direct collocation is to parameterize the control $\mathbf{u}$ and the state variables $\mathbf{x}$ over the time horizon. Both the discrete set of control variables $\mathbf{w}_u$ and the discrete set of state variables $\mathbf{w}_x$ serve as optimization variables, and therefore such a method is referred to as a full discretization approach. Contrary to shooting methods, where the system dynamics are considered by numerical time integration, the direct collocation method considers the

system dynamics as nonlinear constraints of discretized state equations. In other words, the direct collocation approach formulates an NLP problem where the evolution of the state and control variables are solved simultaneously. Hence, the discretized state equations are satisfied only for an optimal solution of the NLP problem.

In direct collocation, the time horizon $t \in [t_0, t_\mathrm{f}]$ is divided into $K$ so-called collocation intervals defined by the time grid

$$\mathbb{G}_\mathrm{x} = \{t_0 < \ldots < t_k < \ldots < t_{K-1} < t_K = t_\mathrm{f}\}, \tag{3.88}$$

where the evolution of the state variables is approximated by polynomials in each collocation interval $t \in [t_k, t_{k+1})$, $k \in \{0, \ldots, K-1\}$. Usually, the collocation intervals are defined on the same time grid as the control parameterization, i.e., $\mathbb{G}_\mathrm{x} = \mathbb{G}_\mathrm{u}$ with $K = M$. Each polynomial is parameterized with $(d+1)$ coefficients leading to a function of degree $d$. The polynomials can be formulated in a number of equivalent ways, but usually the Lagrange interpolation formula is used [17]. Therein, the polynomial coefficients $\mathbf{c}_k$ of the $k$-th collocation interval represent state variables

$$\mathbf{c}_k = (\mathbf{x}_{k,0}, \ldots, \mathbf{x}_{k,d}) \in \mathbb{R}^{N_\mathrm{c}}, \tag{3.89}$$

with $N_\mathrm{c} = N_\mathrm{n}(d+1)$, defined on the $k$-th collocation grid

$$\mathbb{G}_{\mathbf{c}_k} = \{t_k = t_{k,0}, \ldots, t_{k,d}\}. \tag{3.90}$$

The first subscript denotes the $k$-th collocation interval, and the second subscript denotes the respective discretization point of the collocation interval. A common approach to select the collocation grid $\mathbb{G}_{\mathbf{c}_k}$ is to use the Gauss-Legendre collocation scheme [9]. This approach is known for the lowest integration error regarding a fixed number of function evaluations [76]. However, the Lagrange polynomials [98] for the $k$-th collocation interval $\mathbf{l} : \mathbb{R} \times \mathbb{R}^{N_\mathrm{c}} \to \mathbb{R}^{N_\mathrm{n}}$ are defined by the linear combination

$$\mathbf{l}_k(t, \mathbf{c}_k) = \sum_{i=0}^{d} \mathbf{x}_{k,i} \ell_{k,i}(t), \tag{3.91}$$

where the Lagrange basis functions $\ell : \mathbb{R} \to \mathbb{R}$ are given by

$$\ell_{k,i}(t) = \prod_{j=0, j\neq i}^{d} \frac{t - t_{k,i}}{t_{k,i} - t_{k,j}}. \tag{3.92}$$

The Lagrange basis functions have the property

$$\ell_{k,i}(t_{k,j}) = \begin{cases} 0 & \text{for} \quad i = j \\ 1 & \text{for} \quad i \neq j \end{cases}, \tag{3.93}$$

indicating that the interpolation coincides with the state variables on the collocation grid $\mathbb{G}_{\mathbf{c}_k}$, i.e.,

$$\mathbf{l}_k(t_{k,i}, \mathbf{c}_k) = \mathbf{x}_{k,i}. \tag{3.94}$$

Assuming that the state variables $\mathbf{x}_{k,0}$ at $t_{k,0}$ are known, one can define the remaining collocation state variables $\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,d}$ on the time grid $\mathbb{G}_{\mathbf{c}_k} \setminus \{t_{k,0}\}$ such that the discretized state equations are satisfied by the time derivative of (3.94):

$$\dot{\mathbf{l}}_k(t_{k,i}, \mathbf{c}_k) = \mathbf{f}(\underbrace{\mathbf{l}_k(t_{k,i}, \mathbf{c}_k)}_{\mathbf{x}_{k,i}}, \mathbf{u}_{k,i}, \boldsymbol{\xi}), \quad k \in \{0, \ldots, M-1\},\ i \in \{1, \ldots, d\}. \tag{3.95}$$

In addition, continuity of the polynomials over the collocation intervals can be enforced by

$$\mathbf{l}_k(t_{k+1,0}, \mathbf{c}_k) = \mathbf{x}_{k+1,0}. \tag{3.96}$$

The equations (3.95) and (3.96) are referred to as collocation constraints and ensure that the system dynamics are approximated by the polynomials $\mathbf{l}$. These collocation constraints are considered in terms of embedded equality constraints in the direct collocation NLP problem formulation. The optimization iterations and fulfilling the discretized state equations proceed simultaneously. Therefore, all state variables on the collocation grids $\mathbb{G}_{\mathbf{c}_k}$, $k \in \{0, \dots, M-1\}$, are collected as

$$\mathbf{w}_\mathrm{x} = \left(\mathbf{x}_{0,0}^\mathsf{T}, \mathbf{x}_{0,1}^\mathsf{T}, \dots, \mathbf{x}_{0,d}^\mathsf{T}, \mathbf{x}_{1,0}^\mathsf{T}, \dots, \mathbf{x}_{M-1,d}^\mathsf{T}, \mathbf{x}_{M,0}^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^{N_\mathrm{x}}, \tag{3.97}$$

and serve as optimization variables, with $N_\mathrm{x} = M \cdot N_\mathrm{c} + N_\mathrm{n}$.

The direct collocation approach transcribes the OCP defined in (3.2)–(3.7) into the following finite-dimensional NLP problem formulated as

$$\min_{\mathbf{w}_\mathrm{u}, \mathbf{w}_\mathrm{x}, \boldsymbol{\xi}, t_\mathrm{f}} J = \sum_{k=0}^{M-1} \int_{t_{k,0}}^{t_{k+1,0}} L(\mathbf{l}_k, \mathbf{u}, \boldsymbol{\xi}, t)\,\mathrm{d}t + E(\mathbf{x}_{M,0}, t_{M,0}) \tag{3.98}$$

s.t.

$$\boldsymbol{\phi}(\mathbf{x}_{M,0}, t_{M,0}) = \mathbf{0} \tag{3.99}$$

$$\mathbf{g}(\mathbf{x}_{k,0}, \mathbf{u}_{k,0}) = \mathbf{0} \quad k \in \{0, \dots, M-1\} \tag{3.100}$$

$$\mathbf{h}(\mathbf{x}_{k,0}, \mathbf{u}_{k,0}) \leq \mathbf{0} \quad k \in \{0, \dots, M-1\} \tag{3.101}$$

$$\bar{\mathbf{x}}_0 = \mathbf{x}_{0,0} \tag{3.102}$$

$$\mathbf{x}_{k+1,0} = \mathbf{l}_k(t_{k+1,0}, \mathbf{c}_k) \qquad k \in \{0, \dots, M-1\} \tag{3.103}$$

$$\dot{\mathbf{l}}_k(t_{k,i}, \mathbf{c}_k) = \mathbf{f}(\mathbf{x}_{k,i}, \mathbf{u}_{k,i}, \boldsymbol{\xi}) \quad k \in \{0, \dots, M-1\},\ i \in \{1, \dots, d\} \tag{3.104}$$

where the set of control variables $\mathbf{w}_\mathrm{u}$, the state variables $\mathbf{w}_\mathrm{x}$, the parameters $\boldsymbol{\xi}$, and the final time $t_\mathrm{f}$ serve as optimization variables. The (in)equality constraints (3.100) and (3.101) are typically enforced to hold on the boundaries of the collocation intervals. As noted by Bordalba et al. [24], this basic method significantly limits the kinematic accuracy of mechanical systems formulated with redundant coordinates. The state variables $\mathbf{x}_{k,0}$ tend to drift off from the state space manifold due to discretization errors in (3.103) and (3.104). The drift problem can not be prevented by considering an additional set of constraints

$$\dot{\mathbf{l}}_k(t_{k,0}, \mathbf{c}_k) = \mathbf{f}(\mathbf{x}_{k,0}, \mathbf{u}_{k,0}, \boldsymbol{\xi}) \tag{3.105}$$

for the direct collocation NLP problem because the continuity constraints (3.103) already specify the state variables $\mathbf{x}_{k,0}$. Considering the constraints (3.105) in the direct collocation NLP problem would introduce redundant constraints. Therefore, the first-order gradients of the constraints are not linearly independent, which is a necessary condition to solve the NLP problem; see Section 3.3.5 for regularity conditions. However, the drift problem can be solved using a projection method or a local coordinate method as introduced in [24].

Direct collocation yields a large-size NLP problem due to a full discretization scheme regarding control and state variables. Nonetheless, the structure of the problem is sparse since the continuity constraints depend only on some control variables and two to three state variables [19]. Thus, using an optimization solver tailored for sparse problems is essential to take advantage of the sparsity. However, solving the NLP problem is an iterative process. Therefore, an implicit ODE discretization can be used without solving a nonlinear equation in each time integration step since all state variables serve as optimization variables [17]. For more details on direct collocation methods, the reader is referred to [14, 54, 79, 158].

Figure 3.7 visualizes the evolution of the approximated state variable $l$ (–) in the direct collocation approach. Therein, the state variable of the $k$-th collocation interval is approximated by a polynomial of degree $d = 3$ with the state variables $x_{k,i}$ (●). The value of the polynomial $l_k$ evaluated at $t_{k+1,0}$ (○) has to coincide with the subsequent polynomial $l_{k+1}$ to ensure continuity over the time horizon. Moreover, the state equations have to be satisfied on the time grid $\mathbb{G}_{\mathbf{c}_k} \setminus \{t_{k,0}\}$ and the a priori given state variables (×) have to be satisfied by the polynomials for an optimal solution of the NLP problem.

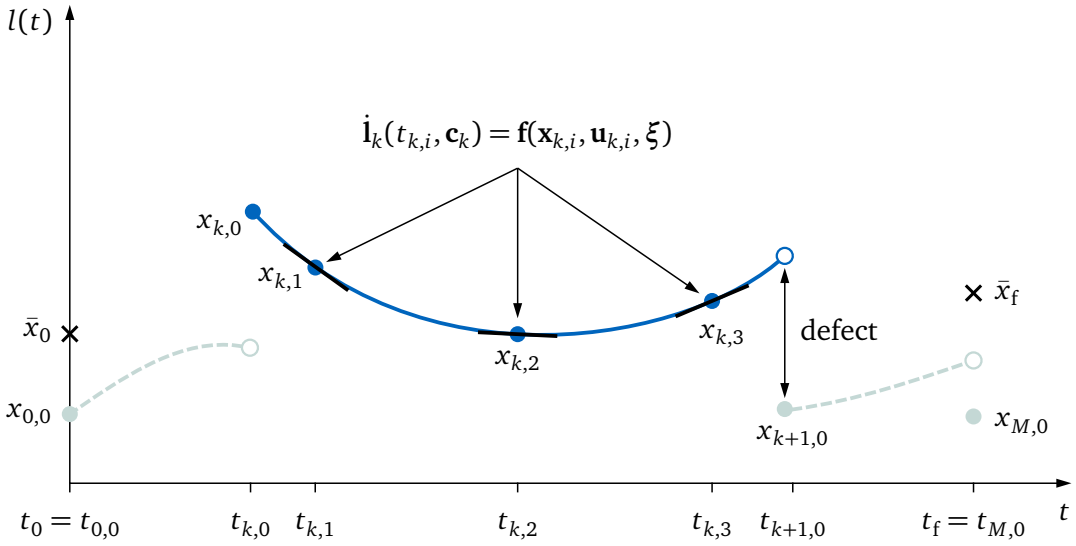



**Figure 3.7:** Evolution of the state variable in the direct collocation method for $d = 3$

### 3.3.4 Nonlinear Programming Problem Formulation

Direct methods transcribe the original infinite-dimensional OCP into a finite-dimensional NLP problem; see the problem formulations for direct single shooting, direct multiple shooting, and direct collocation in the previous sections. In general, an NLP problem is formulated as

$$\min_{\mathbf{z}} J(\mathbf{z}) \tag{3.106}$$

s.t.

$$\hat{\mathbf{g}}(\mathbf{z}) = \mathbf{0} \tag{3.107}$$

$$\hat{\mathbf{h}}(\mathbf{z}) \leq \mathbf{0} \tag{3.108}$$

The above NLP problem aims to find a set of optimization variables $\mathbf{z} = \mathbf{z}^* \in \mathbb{R}^{N_z}$ to minimize a defined cost function $J : \mathbb{R}^{N_z} \to \mathbb{R}$ concerning a set of constraints. The minimization problem can be inverted into an equivalent maximization problem by $\max J(\mathbf{z}) = \min -J(\mathbf{z})$. Equality and inequality constraints are denoted by $\hat{\mathbf{g}} : \mathbb{R}^{N_z} \to \mathbb{R}^{N_p}$ and $\hat{\mathbf{h}} : \mathbb{R}^{N_z} \to \mathbb{R}^{N_q}$, respectively. For the sake of convenience, (in)equality constraints are concatenated into a general set of nonlinear constraints by $\mathbf{c}^\mathsf{T} = \left(\hat{\mathbf{g}}^\mathsf{T}, \hat{\mathbf{h}}^\mathsf{T}\right) \in \mathbb{R}^{N_p+N_q}$. A constraint $c_i$ is called active at an optimal point if $c_i(\mathbf{z}^*) = 0$, i.e., all equality constraints must be active, while some inequality constraints may be inactive at an optimal point. The active (in)equality constraints are defined by the finite set of indices

$$\mathcal{A}(\mathbf{z}^*) = \{i : c_i(\mathbf{z}^*) = 0, i = 1 \ldots, N_p + N_q\} \subseteq \{1, \ldots, N_p + N_q\}, \tag{3.109}$$

which is called the active set of constraints concatenated into the vector $\mathbf{c}_\mathrm{a} \in \mathbb{R}^{N_\mathrm{a}}$.

The general NLP problem in (3.106)–(3.108) can be treated by well-known classical optimization methods, e.g., the SQP method described in Section 3.3.6 or the IP method described in Section 3.3.7. Various third-party solvers are available as software packages for solving an NLP problem. To solve the problem, the user must provide information through an interface of third-party solvers, as illustrated in Fig. 3.8. When interfacing with third-party solvers, the user must provide function evaluations of the cost function $J$, equality constraints $\hat{\mathbf{g}}$, and inequality constraints $\hat{\mathbf{h}}$ at each iteration. Providing user-computed first-order gradients is usually optional but significantly speeds up the optimization procedure; see the scientific contributions of the author's publications in [99, 101, 102] for an efficient and accurate computation of first-order gradients. Apart from first-order gradients, second-order gradients (Hessian) can also be provided to the solver. Nonetheless, the Hessian is usually computed internally by the solver using a quasi-Newton updating method, such as the well-known Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [26, 57, 70, 147]. In addition to the required information at each iteration, the initial starting point $\mathbf{z}_0$ for the initialization, optimization parameters such as the termination criteria, or the type of internal linear solver used for step computation are provided before the optimization procedure starts.

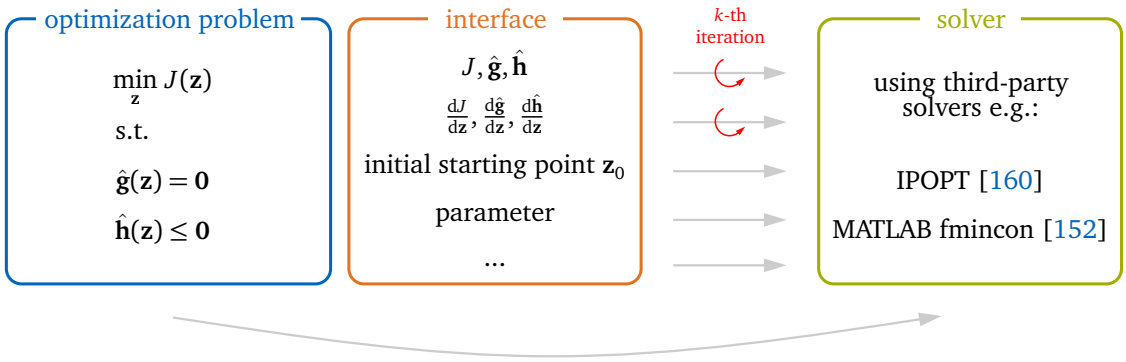



**Figure 3.8:** Schematic interface for solving an NLP problem by a third-party solver

### 3.3.5 Optimality Conditions

The general NLP problem in (3.106)–(3.108) is solved by introducing the Lagrangian function

$$\mathcal{L}(\mathbf{z}, \boldsymbol{\xi}, \boldsymbol{\eta}) = J(\mathbf{z}) + \boldsymbol{\xi}^\mathsf{T} \hat{\mathbf{g}}(\mathbf{z}) + \boldsymbol{\eta}^\mathsf{T} \hat{\mathbf{h}}(\mathbf{z}), \tag{3.110}$$

where $\boldsymbol{\xi} \in \mathbb{R}^{N_\mathrm{p}}$ and $\boldsymbol{\eta} \in \mathbb{R}^{N_\mathrm{q}}$ are the Lagrange multipliers for the equality and inequality constraints, respectively. Note that $\boldsymbol{\xi}$ is also used to represent a set of design parameters, but in this context, it specifically denotes the Lagrange multipliers associated with equality constraints. In this section, the optimality of an NLP problem is investigated by first-order conditions. In order to characterize an optimal point $\mathbf{z}^*$ of an NLP problem with linearized equations, the problem must fulfill some regularity conditions, also called constraint qualifications [17]. The linear inequality constraint qualification (LICQ) is the most frequently used constraint qualification. The LICQ is satisfied for $\mathbf{z}^*$ and the active set $\mathcal{A}(\mathbf{z}^*)$, if the first-order gradients of the active constraints are linear independent, i.e.,

$$\operatorname{rank}\left([\nabla_\mathbf{z} c_i(\mathbf{z}^*)]_{i \in \mathcal{A}(\mathbf{z}^*)}\right) = N_\mathrm{a}. \tag{3.111}$$

Assuming that $\mathbf{z}^*$ is an optimal point for the NLP problem in (3.106)–(3.108), the cost function and the (in)equality constraints are continuously differentiable, and that the LICQ holds at $\mathbf{z}^*$, then there exist unique Lagrange multipliers $\boldsymbol{\xi}^*$ and $\boldsymbol{\eta}^*$ such that the following conditions are satisfied:

- **Stationarity**

$$\nabla_{\mathbf{z}} \mathcal{L}(\mathbf{z}^*, \boldsymbol{\xi}^*, \boldsymbol{\eta}^*) = \mathbf{0} \tag{3.112}$$

- **Primal feasibility**

$$\nabla_{\boldsymbol{\xi}} \mathcal{L}(\mathbf{z}^*, \boldsymbol{\xi}^*, \boldsymbol{\eta}^*) = \hat{\mathbf{g}}(\mathbf{z}^*) = \mathbf{0} \tag{3.113}$$

$$\nabla_{\boldsymbol{\eta}} \mathcal{L}(\mathbf{z}^*, \boldsymbol{\xi}^*, \boldsymbol{\eta}^*) = \hat{\mathbf{h}}(\mathbf{z}^*) \leq \mathbf{0} \tag{3.114}$$

- **Dual feasibility**

$$\boldsymbol{\eta}^* \geq \mathbf{0} \tag{3.115}$$

- **Complementary slackness**

$$\boldsymbol{\eta}^{*\mathsf{T}} \hat{\mathbf{h}}(\mathbf{z}^*) = 0 \tag{3.116}$$

The conditions (3.112)–(3.116) are the so-called KKT conditions. These conditions were developed independently by Karush in his master's thesis in 1939 [89] and by Kuhn and Tucker in 1951 [97]. The complementary slackness condition implies whether the corresponding inequality constraint is active or inactive at an optimal point:

- **Inactive inequality constraint**
  The optimal point $\mathbf{z}^*$ lies in a region where $\hat{h}_i(\mathbf{z}^*) < 0$ holds for the $i$-th element of the vector $\hat{\mathbf{h}}$ and therefore, the corresponding Lagrange multiplier is $\eta_i^* = 0$.

- **Active inequality constraint**
  The optimal point $\mathbf{z}^*$ lies in a region where $\hat{h}_i(\mathbf{z}^*) = 0$ holds for the $i$-th element of the vector $\hat{\mathbf{h}}$ and therefore, the corresponding Lagrange multiplier is $\eta_i^* \geq 0$.

In general, the KKT conditions are nonlinear equations that can not be directly employed to compute an optimal point of the NLP problem. Iterative methods, e.g., the SQP method or the IP method, have to be utilized to compute an optimal point.

### 3.3.6 A Basic Sequential Quadratic Programming Method

The SQP method, which can be traced back historically to Wilson [163] and mainly popularized by Biggs [18], Han [77] and Powell [124, 125], is one of the most popular methods to solve small or medium-size NLP problems. The basic idea of SQP methods is to approximate the NLP problem by a quadratic programming (QP) subproblem and to use the solution of the subproblem iteratively to converge towards an optimal point of the original NLP problem. In QP problems, the cost function is quadratic, while the constraints are linear. The QP subproblem is derived by linearizing the constraints and approximating the Lagrangian function quadratically at the current major SQP iteration $(\mathbf{z}_k, \boldsymbol{\xi}_k, \boldsymbol{\eta}_k)$ by

$$\min_{\mathbf{d}_k^{\mathrm{z}}} J(\mathbf{z}_k) + \nabla_{\mathbf{z}} J(\mathbf{z}_k)^{\mathsf{T}} \mathbf{d}_k^{\mathrm{z}} + \frac{1}{2} \mathbf{d}_k^{\mathrm{z}\mathsf{T}} \nabla_{\mathbf{zz}}^2 \mathcal{L}(\mathbf{z}_k) \mathbf{d}_k^{\mathrm{z}} \tag{3.117}$$

s.t.

$$\hat{\mathbf{g}}(\mathbf{z}_k) + \nabla_{\mathbf{z}} \hat{\mathbf{g}}(\mathbf{z}_k)^{\mathsf{T}} \mathbf{d}_k^{\mathrm{z}} = \mathbf{0} \tag{3.118}$$

$$\hat{\mathbf{h}}(\mathbf{z}_k) + \nabla_{\mathbf{z}} \hat{\mathbf{h}}(\mathbf{z}_k)^{\mathsf{T}} \mathbf{d}_k^{\mathrm{z}} \leq \mathbf{0} \tag{3.119}$$

where $\mathbf{d}_k^z = \mathbf{z} - \mathbf{z}_k$ is the minimizer of the QP subproblem. The Hessian $\nabla^2_{\mathbf{zz}}\mathcal{L}(\mathbf{z}_k)$ of the Lagrangian function is a symmetric, positive definite approximation using a quasi-Newton updating method, usually employing the BFGS method. Solving the QP subproblem is an iterative process, where minor iterations at the current major SQP iteration are the QP iterations. The QP subproblem is solved by an active-set method, e.g., as shown in [115]. The solution of the subproblem is then used to update the optimization variables from iteration $k$ to iteration $k+1$ by

$$\mathbf{z}_{k+1} \coloneqq \mathbf{z}_k + \alpha_k \mathbf{d}_k^z, \tag{3.120}$$

$$\xi_{k+1} \coloneqq \xi_k + \alpha_k \left(\mathbf{d}_k^\xi - \xi_k\right), \tag{3.121}$$

$$\boldsymbol{\eta}_{k+1} \coloneqq \boldsymbol{\eta}_k + \alpha_k \left(\mathbf{d}_k^\eta - \boldsymbol{\eta}_k\right), \tag{3.122}$$

where $\mathbf{d}_k^\xi$ and $\mathbf{d}_k^\eta$ are the Lagrange multipliers of the QP subproblem. The method can made more robust by introducing the scalar step size $\alpha_k$. The step size is obtained by solving a one-dimensional minimization problem of a proper merit function.

The SQP method for practical implementation is much more sophisticated than the basic SQP method presented above. Various implementations based on the SQP method are available as software packages. The software package SNOPT by Gill et al. [66, 67] is an efficient implementation of the SQP method for large-scale problems, where the QP subproblem is solved using the software package SQOPT [68]. Additional SQP software packages are given by NLPQLP [133], the MATLAB optimization toolbox fmincon (`algorithm={'sqp', 'sqp-legacy','active-set'}`) [152], which is an implementation based on various publications, e.g., [78, 124], and the SciPy function minimize (`method='SLSQP'`) [157], which wraps the SLSQP optimization subroutine originally implemented by Kraft [96]. Besides the aforementioned software packages, many other commercial and open-source implementations are available.

### 3.3.7 A Basic Interior Point Method

The IP or barrier method, which can be traced back historically to Fiacco and McCormick [56] and Frisch [59], has proven to be very efficient for solving large-scale NLP problems. The key idea of IP methods is to transform inequality constraints $\hat{\mathbf{h}}(\mathbf{z}) \leq \mathbf{0}$ into equality constraints by introducing slack variables $\mathbf{s} \geq \mathbf{0} \in \mathbb{R}^{N_\mathrm{q}}$. To incorporate the lower bound on the slack variables, the cost function $J$ is augmented with a logarithmic barrier term, which results in the barrier problem

$$\min_{\mathbf{z},\mathbf{s}} J(\mathbf{z}) - \mu \sum_{i=1}^{N_\mathrm{q}} \ln(s_i) \tag{3.123}$$

s.t.

$$\hat{\mathbf{g}}(\mathbf{z}) = \mathbf{0} \tag{3.124}$$

$$\hat{\mathbf{h}}(\mathbf{z}) + \mathbf{s} = \mathbf{0} \tag{3.125}$$

where the scalar $\mu > 0$ is the barrier parameter. IP methods solve the barrier problem for a decreasing sequence of barrier parameters $\{\mu_k\}$ converging to zero. Thus, the solution of the barrier problem approaches the solution of the original NLP problem [35, 71]. The Lagrangian function of the barrier problem reads

$$\mathcal{L}(\mathbf{z}, \mathbf{s}, \xi, \boldsymbol{\eta}) = J(\mathbf{z}) - \mu \sum_{i=1}^{N_\mathrm{q}} \ln(s_i) + \xi^\mathsf{T} \hat{\mathbf{g}}(\mathbf{z}) + \boldsymbol{\eta}^\mathsf{T} \left(\hat{\mathbf{h}}(\mathbf{z}) + \mathbf{s}\right), \tag{3.126}$$

which can be used to formulate the KKT conditions as follows:

$$\nabla_{\mathbf{z}}\mathcal{L}(\mathbf{z}^*,\mathbf{s}^*,\boldsymbol{\xi}^*,\boldsymbol{\eta}^*)=\mathbf{0}: \qquad \nabla_{\mathbf{z}}J(\mathbf{z}^*)+\nabla_{\mathbf{z}}\hat{\mathbf{g}}(\mathbf{z}^*)\boldsymbol{\xi}^*+\nabla_{\mathbf{z}}\hat{\mathbf{h}}(\mathbf{z}^*)\boldsymbol{\eta}^*=\mathbf{0} \tag{3.127}$$

$$\nabla_{\mathbf{s}}\mathcal{L}(\mathbf{z}^*,\mathbf{s}^*,\boldsymbol{\xi}^*,\boldsymbol{\eta}^*)=\mathbf{0}: \qquad -\mu\mathbf{S}^{*-1}\mathbf{e}+\boldsymbol{\eta}^*=\mathbf{0} \tag{3.128}$$

$$\nabla_{\boldsymbol{\xi}}\mathcal{L}(\mathbf{z}^*,\mathbf{s}^*,\boldsymbol{\xi}^*,\boldsymbol{\eta}^*)=\mathbf{0}: \qquad \hat{\mathbf{g}}(\mathbf{z}^*)=\mathbf{0} \tag{3.129}$$

$$\nabla_{\boldsymbol{\eta}}\mathcal{L}(\mathbf{z}^*,\mathbf{s}^*,\boldsymbol{\xi}^*,\boldsymbol{\eta}^*)=\mathbf{0}: \qquad \hat{\mathbf{h}}(\mathbf{z}^*)+\mathbf{s}^*=\mathbf{0} \tag{3.130}$$

Here $\mathbf{S}:=\operatorname{diag}(\mathbf{s})$ is a diagonal matrix of the slack variables and $\mathbf{e}=(1,\ldots,1)^\mathsf{T}$ is a vector of ones. The key advantage of the IP method is that it avoids using an active-set method to solve the KKT conditions. Therefore, the nonlinear system (3.127)–(3.130) can be solved directly by applying Newton's method

$$\begin{pmatrix} \nabla^2_{\mathbf{zz}}\mathcal{L} & \mathbf{0} & \nabla_{\mathbf{z}}\hat{\mathbf{g}} & \nabla_{\mathbf{z}}\hat{\mathbf{h}} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} & \mathbf{S} \\ \nabla_{\mathbf{z}}\hat{\mathbf{g}}^\mathsf{T} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \nabla_{\mathbf{z}}\hat{\mathbf{h}}^\mathsf{T} & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{pmatrix}_k \begin{pmatrix} \mathbf{d}^{\mathrm{z}} \\ \mathbf{d}^{\mathrm{s}} \\ \mathbf{d}^{\xi} \\ \mathbf{d}^{\eta} \end{pmatrix}_k = -\begin{pmatrix} \nabla_{\mathbf{z}}J+\nabla_{\mathbf{z}}\hat{\mathbf{g}}\boldsymbol{\xi}+\nabla_{\mathbf{z}}\hat{\mathbf{h}}\boldsymbol{\eta} \\ \mathbf{S}\boldsymbol{\eta}-\mu\mathbf{e} \\ \hat{\mathbf{g}} \\ \hat{\mathbf{h}}+\mathbf{s} \end{pmatrix}_k, \tag{3.131}$$

where $\mathbf{H}:=\operatorname{diag}(\boldsymbol{\eta})$ is a diagonal matrix of the Lagrange multipliers. The Hessian $\nabla^2_{\mathbf{zz}}\mathcal{L}$ of the Lagrangian function is a symmetric, positive definite approximation using a quasi-Newton updating method, usually employing the BFGS method. The solution of the linear subsystem (3.131) is used to update the search directions from iteration $k$ to iteration $k+1$ by

$$\mathbf{z}_{k+1}:=\mathbf{z}_k+\alpha^{\mathrm{z}}_k\mathbf{d}^{\mathrm{z}}_k, \tag{3.132}$$

$$\mathbf{s}_{k+1}:=\mathbf{s}_k+\alpha^{\mathrm{s}}_k\mathbf{d}^{\mathrm{s}}_k, \tag{3.133}$$

$$\boldsymbol{\xi}_{k+1}:=\boldsymbol{\xi}_k+\alpha^{\xi}_k\mathbf{d}^{\xi}_k, \tag{3.134}$$

$$\boldsymbol{\eta}_{k+1}:=\boldsymbol{\eta}_k+\alpha^{\eta}_k\mathbf{d}^{\eta}_k, \tag{3.135}$$

where the step size parameters $\alpha_k^{\mathrm{z,s},\xi,\eta}\in(0,1]$ are computed by a proper merit function or a line search filter method, e.g., as described by Fletcher and Leyffer [58].

The basic IP method presented above provides the foundation of modern IP methods. However, several modifications and extensions are required for practical implementations. Various implementations based on the IP method are available as software packages. The software package IPOPT [160] by Wächter and Biegler is an efficient open-source implementation of the IP method for large-scale problems, where the performance of IPOPT depends critically on the selected solver for linear subproblems [151], e.g., HSL MA97 or PARDISO. Other additional IP software packages are given by KNITRO [36], LOQO [155], the MATLAB optimization toolbox fmincon (`algorithm='interior-point'`) [152], which is an implementation based on various publications, e.g., [33, 34, 161], and the SciPy function minimize (`method='trust-constr'`) [157], which is an implementation based on the method described in [34]. Besides the aforementioned software packages, many other commercial and open-source implementations are available.

## 3.4 Sensitivity Analysis

Gradient-based optimization methods utilize first and second-order gradients to update search directions from iteration $k$ to iteration $k+1$. The computation of gradients for an optimization problem is referred to as local sensitivity analysis [131]. An overview of commonly

used approaches in sensitivity analysis is given in [109, 153]. Different approaches for sensitivity analysis in the design optimization of flexible multibody systems are discussed in a comprehensive literature review provided by Gufler et al. [74].

The scope of this section is to discuss commonly used approaches to compute first-order sensitivities of the cost functional $J$ given as

$$J = \int_{t_0}^{t_\mathrm{f}} L(\mathbf{x}(\mathbf{z},t),\mathbf{u}(\mathbf{z},t),\boldsymbol{\xi}(\mathbf{z}))\,\mathrm{d}t + E(\mathbf{x}(\mathbf{z},t_\mathrm{f}),t_\mathrm{f}), \tag{3.136}$$

where the evolution of the state variables is obtained by solving the first-order differential equations

$$\dot{\mathbf{x}}(\mathbf{z},t) = \mathbf{f}(\mathbf{x}(\mathbf{z},t),\mathbf{u}(\mathbf{z},t),\boldsymbol{\xi}(\mathbf{z})) \quad \text{with} \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0. \tag{3.137}$$

Sensitivity analysis aims to compute first-order gradients

$$\nabla_\mathbf{z} J^\mathsf{T} = \left.\frac{\mathrm{d}J}{\mathrm{d}\mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_k}, \tag{3.138}$$

evaluated at the $k$-th iteration. Since the computation of gradients is required at each iteration, an accurate and efficient computation of first-order gradients is crucial in solving optimization problems. The accuracy of the gradient computation influences the convergence behavior of the used solver and the runtime required for a converged solution.

### 3.4.1 Finite-Difference Method

The finite-difference method relies on the Taylor series expansion to compute first-order gradients. It is the most straightforward approach to implement but suffers in terms of accuracy and efficiency, especially for a large number of optimization variables. Most third-party solvers use the finite-difference method by default if the user does not provide gradients. Components of the first-order gradient of the cost functional in (3.136) are approximated as

$$\frac{\mathrm{d}J}{\mathrm{d}z_i} \approx \frac{J(\mathbf{z}_k + \mathbf{e}_i\Delta h) - J(\mathbf{z}_k)}{\Delta h}, \quad i \in \{1,\dots,N_\mathrm{z}\} \tag{3.139}$$

$$\frac{\mathrm{d}J}{\mathrm{d}z_i} \approx \frac{J(\mathbf{z}_k) - J(\mathbf{z}_k - \mathbf{e}_i\Delta h)}{\Delta h}, \quad i \in \{1,\dots,N_\mathrm{z}\} \tag{3.140}$$

concerning forward and backward finite-differences, respectively. Computing a component of the gradient vector via forward and backward finite-differences requires the evaluation of the cost functional at $\mathbf{z}_k$ and at a small perturbation $\mathbf{z}_k \pm \mathbf{e}_i\Delta h$, where $\mathbf{e}_i$ is a unit vector and $\Delta h$ is a small perturbation parameter. Thus, the cost of computing the entire gradient vector is proportional to the number of optimization variables $N_\mathrm{z}$.

The truncation error of forward and backward finite-differences is proportional to the perturbation parameter $\Delta h$. Therefore, a smaller value of the perturbation parameter reduces the truncation error. In practical implementations, a lower bound of the perturbation parameter exists. This bound mainly depends on the accuracy of numerical methods to compute the cost functional, e.g., on the time integration method to compute the evolution of the state variables or the integration method to compute the cost functional [12]. However, finite-differences can be computed even when the cost functional is treated as a black box model. Therefore, the finite-differences are commonly used to verify more sophisticated methods, as discussed in the subsequent sections.

### 3.4.2 Direct Differentiation Method

Besides the finite-difference method, analytical approaches such as the direct differentiation or adjoint variable method can be used to compute sensitivities. Analytical approaches are generally more sophisticated and significantly more expensive to derive and implement than the finite-difference method. However, using analytical approaches in the sensitivity analysis provides high efficiency and high accuracy in computing first-order gradients. The direct differentiation method is based on directly applying the chain rule of differentiation to the function of interest. Sensitivities for the cost functional (3.136) are obtained by

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{z}} = \int_{t_0}^{t_\mathrm{f}} \left( \frac{\partial L}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial L}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial L}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}} \right) \mathrm{d}t + \frac{\partial E}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}, \tag{3.141}$$

where the sensitivities of the state equations $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ are computed by solving the linear matrix differential equations

$$\frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}\mathbf{z}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}} \quad \text{with} \quad \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} = \mathbf{0}. \tag{3.142}$$

The matrix differential equations are obtained by differentiating the state equations (3.137) with respect to the optimization variables $\mathbf{z}$. Thus, the dimension of (3.142) is $N_\mathrm{n} \times N_\mathrm{z}$, which can be interpreted as $N_\mathrm{z}$ independent first-order differential equations of dimension $N_\mathrm{n} \times 1$. Solving the matrix differential equations can become computationally expensive for a large number of optimization variables [162]. However, the state equations (3.137) and the matrix differential equations can be solved simultaneously, which reduces the memory requirements compared to the adjoint variable method.

Sensitivity analysis based on direct differentiation consists of an extensive literature. Detailed derivations of the direct differentiation method in rigid and flexible multibody systems can be found in [16, 27, 43, 49, 75]. Therein, the adjoint variable method is usually mentioned as an alternative and efficient approach in the case of a large number of optimization variables. The adjoint variable method can be interpreted as the dual problem concerning the primal problem of the direct differentiation method; see the discussion of linear duality in Section 2.3.2.

### 3.4.3 Adjoint Variable Method

The adjoint method is probably the most efficient approach to computing sensitivities, especially for optimization problems with numerous optimization variables. Adjoint-based approaches are based on avoiding the direct computation of the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ by introducing adjoint variables. Following the discussion of linear duality in Section 2.3.2, the sensitivities of the cost functional (3.136) using the adjoint variable method are given as

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{z}} = \int_{t_0}^{t_\mathrm{f}} \left[ \frac{\partial L}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial L}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}} + \mathbf{p}^\mathsf{T} \left( \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \xi} \frac{\mathrm{d}\xi}{\mathrm{d}\mathbf{z}} \right) \right] \mathrm{d}t, \tag{3.143}$$

where the adjoint equations are computed by solving the linear first-order differential equations backward in time

$$\dot{\mathbf{p}} = -\left( \frac{\partial L}{\partial \mathbf{x}} \right)^\mathsf{T} - \left( \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\mathsf{T} \mathbf{p} \quad \text{with} \quad \mathbf{p}(t_\mathrm{f}) = \left( \frac{\partial E}{\partial \mathbf{x}} \right)^\mathsf{T}. \tag{3.144}$$

The adjoint sensitivities in (3.143) are computed in a sequential process by solving the state equations forward in time followed by solving the adjoint equations backward in time; see the procedure for using adjoint sensitivities in [99]. Note that the adjoint variables $\mathbf{p}$ are of dimension $N_\mathrm{n} \times 1$, while the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ are of dimension $N_\mathrm{n} \times N_\mathrm{z}$. Thus, the computational effort of solving the adjoint system does not depend on the number of optimization variables, which is not the case when computing the state sensitivities in the direct differentiation approach. In general, the adjoint method is more efficient than the direct differentiation if the number of optimization variables is higher than the number of functions to differentiate, i.e., the number of objectives to minimize and the number of constraints.

Note that the adjoint equations in (3.144) correspond to the adjoint equations as introduced in (3.33) concerning an indirect method. This can be seen by using the Hamiltonian of the optimization problem as defined in (3.12) given by

$$\mathcal{H} = L(\mathbf{x}(\mathbf{z},t),\mathbf{u}(\mathbf{z},t),\boldsymbol{\xi}(\mathbf{z})) + \mathbf{p}^\mathsf{T}(t)\mathbf{f}(\mathbf{x}(\mathbf{z},t),\mathbf{u}(\mathbf{z},t),\boldsymbol{\xi}(\mathbf{z})). \tag{3.145}$$

Using the above Hamiltonian, the adjoint equations in (3.144) can be reformulated as

$$\dot{\mathbf{p}} = -\left(\frac{\partial \mathcal{H}}{\partial \mathbf{x}}\right)^\mathsf{T}, \tag{3.146}$$

which correspond to the adjoint equations as introduced in (3.33). In addition, the final conditions of the adjoint variables in (3.144) correspond to the final conditions as introduced in (3.38) for the case where the state variables are not specified at the final time.

Various formulations of the adjoint variable method for sensitivity analysis of rigid and flexible multibody systems have been developed for different applications, e.g., as presented in [37, 82, 113, 121]. The adjoint variable method in multibody dynamics consists of an active scientific research community that provides novel approaches for efficient sensitivity analysis. Recent developments of adjoint-based sensitivity analysis in multibody dynamics are presented in [40, 104, 105, 156]. Due to its computational efficiency, the general concept of the adjoint variable method is utilized in the scientific contributions of the author's publications in [99, 101, 102] for an efficient and accurate computation of sensitivities.

Chapter 4

# Large Deformation Problems in Flexible Multibody Systems

This chapter introduces the basics of flexible multibody systems, focusing on the absolute nodal coordinate formulation (ANCF). The theory of the ANCF is used in this thesis to describe the dynamical behavior of deformable structures. A brief overview in Section 4.1 highlights two commonly used formulations in flexible multibody dynamics. In Section 4.2, the equations of motion for a single ANCF element are derived. These equations are the basis for flexible multibody systems modeled with multiple elements. The assembly of elements for a flexible multibody system is described in Section 4.3.

The scope of this chapter is to provide sufficient details to understand the flexible multibody formulation used in the scientific contributions in [101, 102]. It is not an aim to provide a complete treatment of theory and applications on flexible multibody systems. Instead, interested readers are referred to excellent textbooks, e.g., on general multibody dynamics [120, 132, 135, 140, 142], and on flexible multibody dynamics [7, 61]. The theoretical aspects of flexible multibody systems presented in the next sections mainly follow the work of these textbooks.

## 4.1 Overview

Flexible multibody dynamics deals with modeling and analyzing constrained flexible bodies that undergo translational and rotational displacements with nonnegligible deformations. Several ways and formulations have been developed for modeling and simulation of effects due to the elasticity of flexible bodies. Comprehensive literature reviews in research and applications of multibody dynamics are found in [50, 52, 74, 138]. Popular formulations of flexible bodies in multibody dynamics are the floating frame of reference formulation (FFRF) and ANCF. The basic idea of FFRF is to decompose the motion of a flexible body into a rigid body motion and a superimposed deformation. Two sets of coordinates are used to identify the configuration of the flexible body. Absolute reference coordinates describe the location and rotation of a selected body coordinate system, while elastic coordinates describe the deformation field of the body relative to the body coordinate system. The FFRF is mainly used for modeling large rigid body motions and small deformations relative to the body coordinate system. In comparison, the ANCF is suitable for correct modeling large deformations and large rotations of flexible bodies. The basic idea is to describe the dynamics in terms of absolute nodal displacements and slopes at nodal points of finite elements. An arbitrary point of the element is defined by the nodal coordinates and a global shape function that consists of a complete set of rigid body modes. Detailed comparisons between FFRF and ANCF can be found in [44, 145, 146].

Flexible bodies are the main components in the rapidly growing field of soft robotics. Such

robots are made of highly deformable materials and undergo high deformation during motion. Current challenges in the design, modeling, and control of soft robotics are highlighted in [2, 127, 130, 164]. The authors agree that no consistent approach to modeling soft robots exists. Shabana [141] and Shabana and Eldeeb [144] pointed out that the challenges in modeling soft robots can be addressed by utilizing ANCF. The formulation is suitable for accurately representing large deformations, and nonlinear material behavior can be considered. Therefore, the ANCF is used in this dissertation to describe the flexibility of bodies. The thesis is not intended to introduce new approaches to ANCF, but focuses on using well-established ANCF formulations for adjoint-based sensitivity analysis in OCPs; see the examples in [101, 102].

## 4.2 Absolute Nodal Coordinate Formulation

The ANCF has been introduced in a technical report by Shabana [136] and become a popular formulation for flexible multibody dynamics. The ANCF is a nonlinear finite element formulation developed for correctly modeling large deformations and large rotations [137]. The general idea of the ANCF is to use absolute nodal position coordinates and absolute nodal slopes as degrees of freedom [61]. In contrast to conventional nonlinear finite element methods, the ANCF uses the absolute nodal slopes instead of rotational degrees of freedom to parameterize the cross-section and, therefore, does not necessarily suffer from singularities arising from angular parameterizations. In ANCF, the centrifugal and Coriolis forces are zero, and the mass matrix is constant for the degrees of freedom, which is particularly advantageous in dynamic simulations [55]. However, highly nonlinear stiffness terms for the formulation of elastic forces occur due to the use of absolute nodal position coordinates defined in a global coordinate system. Thus, even a linear elastic material model leads to nonlinear elastic forces. Comprehensive literature reviews of ANCF contributions are provided in [63, 118]. A detailed classification and description of the general requirements for ANCF finite elements can be found in [139]. For an overview and further research directions using ANCF elements, the reader is referred to Shabana [143].

A variety of ANCF elements have been developed to model beams/cables, plates/shells, and solid elements, and have been widely used in engineering applications. In this dissertation, the following two-dimensional standard ANCF elements are considered to examine effects due to the elasticity of flexible bodies:

- The formulation proposed by Berzeri and Shabana [11] has been used in [102]. This classical large deformation beam model only accounts for axial and bending deformation, while the effect of shear deformation is neglected. Moreover, the cross-section of the beam is assumed to remain plane and perpendicular to the beam center line. This element has been formulated based on the Euler-Bernoulli beam theory.

- The formulation proposed by Omar and Shabana [117] has been used in [101]. This classical large deformation beam model accounts for axial and bending deformation while considering shear deformation. Moreover, the cross-section of the beam does not remain perpendicular to the beam center line following the idea of Timoshenko's beam theory, but the cross-section in this element does not remain rigid.

Both elements have been tested extensively in the literature and are used in structural-optimization problems, e.g., [83, 150]. The shear deformable formulation proposed by Omar and Shabana [117] is briefly described in the following sections.

### 4.2.1 Kinematics

The geometric description of a planar two-noded ANCF element is based on nodal position vectors and nodal slope vectors; see Fig. 4.1 for an element in the undeformed reference configuration and in the deformed configuration. Elements are defined in the global coordinate system $(x, y)$ by position vectors $\mathbf{r}^{(j)} \in \mathbb{R}^2$ (–), axial slope vectors $\mathbf{r}_\xi^{(j)} \in \mathbb{R}^2$ (–), and transverse slope vectors $\mathbf{r}_\eta^{(j)} \in \mathbb{R}^2$ (–) of the $j$-th node (•). The two-dimensional element consists of six

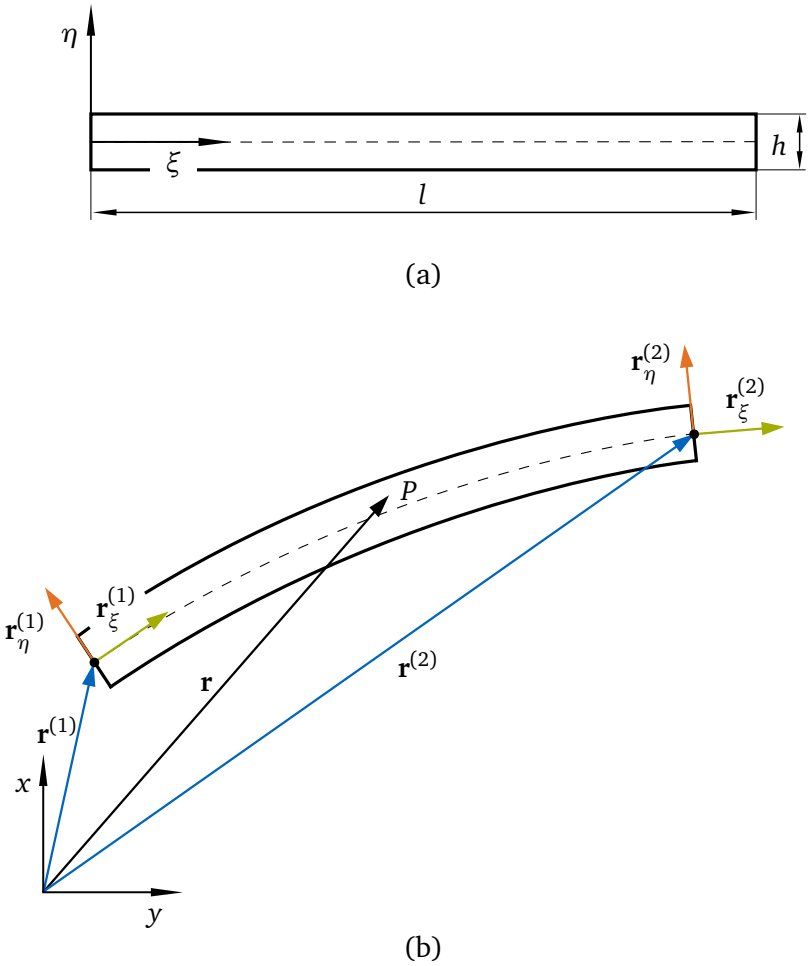



**Figure 4.1:** Position and slope vectors of an ANCF element: (a) is the undeformed reference configuration, and (b) is the deformed configuration

degrees of freedom at each node, where the nodal position is described with one vector, and the orientation of the cross-section is represented with two vectors. Such an element is often referred in literature as a fully-parameterized element since the full gradient information of the nodal position is included in the vector of degrees of freedom. For a clear distinction between fully-parameterized and gradient deficient elements and a comparison of advantages see [143]. Partial derivatives define the slope vectors for the $j$-th node as

$$\mathbf{r}_\xi^{(j)} = \frac{\partial \mathbf{r}^{(j)}}{\partial \xi} \quad \text{and} \quad \mathbf{r}_\eta^{(j)} = \frac{\partial \mathbf{r}^{(j)}}{\partial \eta}, \tag{4.1}$$

where the coordinates in the undeformed reference configuration $\xi$ and $\eta$ denote the axis of the beam and the axis along the cross-section, respectively. All vectors of an element are

concatenated in the vector of generalized coordinates

$$\mathbf{q} = \left(\mathbf{r}^{(1)}, \mathbf{r}_{\xi}^{(1)}, \mathbf{r}_{\eta}^{(1)}, \mathbf{r}^{(2)}, \mathbf{r}_{\xi}^{(2)}, \mathbf{r}_{\eta}^{(2)}\right)^{\mathsf{T}} \in \mathbb{R}^{12}. \tag{4.2}$$

A material point $\boldsymbol{\xi} = (\xi, \eta)^{\mathsf{T}}$ in the undeformed reference configuration can be mapped to the deformed configuration defined in the global coordinate system by

$$\mathbf{r}(\boldsymbol{\xi}, t) = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} a_0(t) + a_1(t)\xi + a_2(t)\eta + a_3(t)\xi\eta + a_4(t)\xi^2 + a_5(t)\xi^3 \\ b_0(t) + b_1(t)\xi + b_2(t)\eta + b_3(t)\xi\eta + b_4(t)\xi^2 + b_5(t)\xi^3 \end{pmatrix}, \tag{4.3}$$

where the position field is approximated using cubic polynomial functions with the time-dependent coefficients $a_0 \dots a_5$ and $b_0 \dots b_5$. The global position vector $\mathbf{r}(\boldsymbol{\xi}, t)$ can be decoupled into a local part $\mathbf{B}(\boldsymbol{\xi})$ related to the undeformed reference configuration and a time-dependent part $\boldsymbol{\gamma}(t)$ by the matrix representation

$$\mathbf{r}(\boldsymbol{\xi}, t) = \mathbf{B}(\boldsymbol{\xi})\boldsymbol{\gamma}(t) = \begin{pmatrix} 1 & \xi & \eta & \xi\eta & \xi^2 & \xi^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \xi & \eta & \xi\eta & \xi^2 & \xi^3 \end{pmatrix} \begin{pmatrix} a_0(t) \\ \vdots \\ a_5(t) \\ b_0(t) \\ \vdots \\ b_5(t) \end{pmatrix}. \tag{4.4}$$

The polynomial coefficients $\boldsymbol{\gamma}$ are defined such that the generalized coordinates of an element represent the nodal position vector of both nodes

$$q_1 = r_1|_{\xi=0,\eta=0}, \qquad q_2 = r_2|_{\xi=0,\eta=0}, \qquad q_7 = r_1|_{\xi=l,\eta=0}, \qquad q_8 = r_2|_{\xi=l,\eta=0}, \tag{4.5}$$

and the global slope vectors of both nodes

$$\begin{aligned} q_3 &= \left.\frac{\partial r_1}{\partial \xi}\right|_{\xi=0,\eta=0}, \quad q_4 = \left.\frac{\partial r_2}{\partial \xi}\right|_{\xi=0,\eta=0}, \quad q_5 = \left.\frac{\partial r_1}{\partial \eta}\right|_{\xi=0,\eta=0}, \quad q_6 = \left.\frac{\partial r_2}{\partial \eta}\right|_{\xi=0,\eta=0}, \\ q_9 &= \left.\frac{\partial r_1}{\partial \xi}\right|_{\xi=l,\eta=0}, \quad q_{10} = \left.\frac{\partial r_2}{\partial \xi}\right|_{\xi=l,\eta=0}, \quad q_{11} = \left.\frac{\partial r_1}{\partial \eta}\right|_{\xi=l,\eta=0}, \quad q_{12} = \left.\frac{\partial r_2}{\partial \eta}\right|_{\xi=l,\eta=0}. \end{aligned} \tag{4.6}$$

The components of the generalized coordinates in (4.5) and (4.6) are concatenated in the vector of generalized coordinates by

$$\mathbf{q}(t) = \mathbf{A}\boldsymbol{\gamma}(t), \tag{4.7}$$

where the coefficient matrix $\mathbf{A}$ is defined by

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & l & 0 & 0 & l^2 & l^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & l & 0 & 0 & l^2 & l^3 \\ 0 & 1 & 0 & 0 & 2l & 3l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2l & 3l^2 \\ 0 & 0 & 1 & l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & l & 0 & 0 \end{pmatrix}. \tag{4.8}$$

Substituting (4.7) into (4.4) yields

$$\mathbf{r}(\xi,t) = \underbrace{\mathbf{B}(\xi)\mathbf{A}^{-1}}_{\mathbf{S}(\xi)}\mathbf{q}(t), \tag{4.9}$$

in which the shape function matrix $\mathbf{S}$ has been introduced. The shape function matrix maps the generalized coordinates to the global position vector which reads

$$\mathbf{S} = \begin{pmatrix} s_1 & 0 & s_2 & 0 & s_3 & 0 & s_4 & 0 & s_5 & 0 & s_6 & 0 \\ 0 & s_1 & 0 & s_2 & 0 & s_3 & 0 & s_4 & 0 & s_5 & 0 & s_6 \end{pmatrix}, \tag{4.10}$$

where the components of the shape function matrix can be written as follows

$$s_1 = 1 - 3\alpha^2 + 2\alpha^3, \tag{4.11}$$
$$s_2 = l(\alpha - 2\alpha^2 + \alpha^3), \tag{4.12}$$
$$s_3 = h(1-\alpha)\beta, \tag{4.13}$$
$$s_4 = 3\alpha^2 - 2\alpha^3, \tag{4.14}$$
$$s_5 = l(-\alpha^2 + \alpha^3), \tag{4.15}$$
$$s_6 = h\alpha\beta. \tag{4.16}$$

Note that the components of the shape function matrix are formulated as a function of the normalized element coordinates $\alpha = \xi/l$ and $\beta = \eta/h$, where the Jacobian

$$\mathbf{J}_1 = \frac{\partial \xi}{\partial \boldsymbol{\alpha}} = \begin{pmatrix} l & 0 \\ 0 & h \end{pmatrix} \tag{4.17}$$

accounts for the transformation between the unit element coordinates $\boldsymbol{\alpha} = (\alpha, \beta)^\mathsf{T}$ and the element coordinates $\xi$. Normalizing the element coordinates aims to be compatible with standard numerical integration methods, e.g., using a Gaussian quadrature rule, to derive the equations of motion [62]. Analyzing the shape functions in (4.11)–(4.16), it is obvious that the centerline of the element is represented by a cubic interpolation in $\beta$ along the element axis, while the height of the element is represented by a linear interpolation in $\alpha$ along the cross-section [112].

To summarize the kinematics of the ANCF element proposed in [117], the position vector of an arbitrary point $P$ is defined in the global coordinate system $(x, y)$ using the shape function matrix and the generalized coordinates as

$$\mathbf{r}(\boldsymbol{\alpha},t) = \mathbf{S}(\boldsymbol{\alpha})\mathbf{q}(t), \tag{4.18}$$

in which the shape function matrix is a function of unit element coordinates $\boldsymbol{\alpha}$. In addition, the interpolation of the reference element is defined as follows

$$\mathbf{r}_0(\boldsymbol{\alpha}) = \mathbf{S}(\boldsymbol{\alpha})\mathbf{q}_0, \tag{4.19}$$

where the generalized coordinates in the reference configuration are denoted by $\mathbf{q}_0$.

### 4.2.2 Equations of Motion

The equations of motion for multibody systems can be derived using D'Alembert's principle and the principle of virtual work. D'Alembert's principle states that the virtual work of inertia

forces $\delta W_{\text{inert}}$ and elastic forces $\delta W_{\text{elast}}$ equals the virtual work of external forces $\delta W_{\text{ext}}$

$$\underbrace{\int_V \rho \ddot{\mathbf{r}}^{\mathsf{T}} \delta \mathbf{r} \, \mathrm{d}V}_{\delta W_{\text{inert}}} + \underbrace{\int_V \mathbf{S} : \delta \mathbf{E} \, \mathrm{d}V}_{\delta W_{\text{elast}}} = \underbrace{\int_V \mathbf{b}^{\mathsf{T}} \delta \mathbf{r} \, \mathrm{d}V + M \delta \theta}_{\delta W_{\text{ext}}}, \tag{4.20}$$

where $\rho$ is the mass density of the material, $V$ denotes the volume in the reference configuration, $\mathbf{S}$ is the second order Piola-Kirchhoff stress tensor, $\mathbf{E}$ is the nonlinear Green-Lagrange strain tensor, $\mathbf{b}$ denotes body forces, and $M$ is an external torque acting on the angle of rotation $\theta$ of the beam cross-section. The virtual work in D'Alembert's principle can be formulated in terms of the mass matrix $\mathbf{M}$, the elastic forces $\mathbf{Q}_{\text{elast}}$, and the external forces $\mathbf{Q}_{\text{ext}}$ as

$$\delta W_{\text{inert}} = \delta \mathbf{q}^{\mathsf{T}} \mathbf{M} \ddot{\mathbf{q}}, \quad \text{and} \quad \delta W_{\text{elast}} = \delta \mathbf{q}^{\mathsf{T}} \mathbf{Q}_{\text{elast}}, \quad \text{and} \quad \delta W_{\text{ext}} = \delta \mathbf{q}^{\mathsf{T}} \mathbf{Q}_{\text{ext}}. \tag{4.21}$$

For the derivation of the mass matrix, the elastic forces, and the external forces, see the following sections. Based on D'Alembert's principle and the terms in (4.21), the so-called weak form of the equations of motion for an ANCF element can be formulated as

$$\delta \mathbf{q}^{\mathsf{T}} (\mathbf{M} \ddot{\mathbf{q}} + \mathbf{Q}_{\text{elast}} - \mathbf{Q}_{\text{ext}}) = 0 \quad \forall \delta \mathbf{q}. \tag{4.22}$$

Since the element is not subjected to any constraints, the weak form holds for all $\delta \mathbf{q}$. Thus, the equations of motion for an ANCF element can be written as

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{Q}_{\text{elast}} = \mathbf{Q}_{\text{ext}}. \tag{4.23}$$

The second-order differential equations can be transformed into first-order differential equations to be compatible with the first-order differential equations of an OCP introduced in (3.7). Introducing generalized velocities $\mathbf{v} = \dot{\mathbf{q}}$ transforms the second-order system into

$$\dot{\mathbf{x}} = \mathbf{f} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{Q}_{\text{ext}} - \mathbf{Q}_{\text{elast}} \end{pmatrix}, \tag{4.24}$$

wherein the state variables are expressed by $\mathbf{x}^{\mathsf{T}} = \left(\mathbf{q}^{\mathsf{T}}, \mathbf{v}^{\mathsf{T}}\right)$.

#### Mass Matrix

The symmetric mass matrix of an element is obtained using the virtual work of inertia forces

$$\delta W_{\text{inert}} = \int_V \rho \ddot{\mathbf{r}}^{\mathsf{T}} \delta \mathbf{r} \, \mathrm{d}V = \ddot{\mathbf{q}}^{\mathsf{T}} \underbrace{\int_V \rho \mathbf{S}^{\mathsf{T}} \mathbf{S} \, \mathrm{d}V}_{\mathbf{M}} \, \delta \mathbf{q}, \tag{4.25}$$

in which the variation of the global position vector is computed by

$$\delta \mathbf{r} = \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \delta \mathbf{q} = \mathbf{S} \delta \mathbf{q}. \tag{4.26}$$

The mass matrix defined by (4.25) reads

$$\mathbf{M} = \int_V \rho \mathbf{S}^{\mathsf{T}} \mathbf{S} \, \mathrm{d}V = w \int_0^1 \int_{-1/2}^{1/2} \rho \mathbf{S}^{\mathsf{T}} \mathbf{S} \, |\det(\mathbf{J}_1)| \, \mathrm{d}\alpha \, \mathrm{d}\beta, \tag{4.27}$$

in which the volume integral is transformed to unit element coordinates using the determinant of the Jacobian of the unit element coordinate transformation $|\det(\mathbf{J}_1)| = lh$ and a constant width $w$ of the element. In general, the ANCF element leads to a constant mass matrix that depends only on the inertia and the dimension of the beam [142]. Thus, ANCF-based formulations are advantageous for dynamic simulations of multibody systems from a computational point of view.

#### Elastic Forces

The elastic forces of an element are obtained using the continuum mechanics-based formulation of the virtual work of elastic forces $\delta W_{\text{elast}}$, which is defined by using the nonlinear Green-Lagrange strain tensor and the second-order Piola-Kirchhoff stress tensor. The Green-Lagrange strain tensor reads

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^{\mathsf{T}}\mathbf{F} - \mathbf{I}), \tag{4.28}$$

where $\mathbf{F}$ denotes the deformation gradient and $\mathbf{I}$ is the identity matrix. The deformation gradient is defined by using partial derivatives of the position vector $\mathbf{r}$ with respect to the position vector in the reference configuration $\mathbf{r}_0$ as follows

$$\mathbf{F} = \frac{\partial \mathbf{r}}{\partial \mathbf{r}_0} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\alpha}} \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{r}_0}. \tag{4.29}$$

The transformation between the unit element coordinates $\boldsymbol{\alpha}$ and the position vector of the reference element is defined by the element Jacobian

$$\mathbf{J}_2 = \frac{\partial \mathbf{r}_0}{\partial \boldsymbol{\alpha}}, \tag{4.30}$$

which consistently considers precurved elements geometrically [62]. The element Jacobian simplifies to $\mathbf{J}_2 = \mathbf{J}_1 = \begin{pmatrix} l & 0 \\ 0 & h \end{pmatrix}$ for the case of a straight and undistorted reference configuration in which the local $\xi$-axis is parallel to the global $x$-axis. For linear-elastic material models, the second order Piola–Kirchhoff stress tensor is expressed by

$$\mathbf{S} = \mathbf{D}^4 : \mathbf{E}, \tag{4.31}$$

where $\mathbf{D}^4$ denotes the fourth order tensor of elasticities. For the planar formulation of the ANCF element, it is required to consider either plane stress or plane strain conditions. Thus, the stress-strain relation is rewritten in terms of the engineering strain vector $\boldsymbol{\varepsilon}$ and the stress vector $\boldsymbol{\sigma}$

$$\boldsymbol{\varepsilon} = \left(E_{xx}, E_{yy}, 2E_{xy}\right)^{\mathsf{T}} \quad \text{and} \quad \boldsymbol{\sigma} = \left(S_{xx}, S_{yy}, S_{xy}\right)^{\mathsf{T}}, \tag{4.32}$$

respectively. In case of plane stress, the elasticity matrix reads

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix}, \tag{4.33}$$

in which $E$ and $\nu$ represent the Young's modulus and the Poisson's ratio of the beam material, respectively. Using the elasticity matrix and the engineering strains, the stress-strain relation is given by

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}. \tag{4.34}$$

The stress-strain relation used follows the paper by Gerstmayr et al. [62], and is slightly updated from the original paper by Omar and Shabana [117] to be consistent with traditional planar finite elements.

The strain energy of an element with a rectangular cross-section is usually defined as

$$U = \frac{1}{2}\int_V \boldsymbol{\varepsilon}^{\mathsf{T}}\mathbf{D}\boldsymbol{\varepsilon}\,\mathrm{d}V = \frac{1}{2}w\int_{-1/2}^{1/2}\int_0^1 \boldsymbol{\varepsilon}^{\mathsf{T}}\mathbf{D}\boldsymbol{\varepsilon}\,|\det(\mathbf{J}_2)|\,\mathrm{d}\alpha\,\mathrm{d}\beta, \tag{4.35}$$

where the volume integral is transformed to unit element coordinates. Note that the integral transformation requires considering the determinate of the element Jacobian in (4.30). The virtual work of the elastic forces can be defined using the strain energy as

$$\delta W_{\text{elast}} = \delta \mathbf{q}^{\mathsf{T}} \frac{\partial U}{\partial \mathbf{q}}, \tag{4.36}$$

see [140]. Thus, the elastic forces are given by

$$\mathbf{Q}_{\text{elast}} = \frac{\partial U}{\partial \mathbf{q}}. \tag{4.37}$$

#### Generalized External Forces

Generalized external forces of an element are obtained using the principle of virtual work

$$\delta W_{\text{ext}} = \delta \mathbf{q}^{\mathsf{T}} \mathbf{Q}_{\text{ext}} = \int_V \delta \mathbf{r}^{\mathsf{T}} \mathbf{b} \, \mathrm{d}V + M \delta \theta \Rightarrow \mathbf{Q}_{\text{ext}}. \tag{4.38}$$

In this thesis, the vector of generalized forces $\mathbf{Q}_{\text{ext}}$ due to external forces and torques reads

$$\mathbf{Q}_{\text{ext}} = \mathbf{Q}_{\text{g}} + \mathbf{Q}_{\text{u}} + \mathbf{Q}_{\text{d}}, \tag{4.39}$$

where effects due to gravity $\mathbf{Q}_{\text{g}}$, an applied torque $\mathbf{Q}_{\text{u}}$, and viscous damping for joint friction $\mathbf{Q}_{\text{d}}$ are considered.

The generalized forces due to the distributed gravity of the element are obtained by the first term of the virtual work in (4.38) as

$$\begin{aligned} \delta W_{\text{g}} &= \int_V \delta \mathbf{r}^{\mathsf{T}} \mathbf{b} \, \mathrm{d}V = \delta \mathbf{q}^{\mathsf{T}} \int_V \rho \mathbf{S}^{\mathsf{T}} \, \mathrm{d}V \mathbf{g} \\ &= \delta \mathbf{q}^{\mathsf{T}} \underbrace{w \int_{-1/2}^{1/2} \int_0^1 \rho \mathbf{S}^{\mathsf{T}} \, |\det(\mathbf{J}_1)| \, \mathrm{d}\alpha \, \mathrm{d}\beta \mathbf{g}}_{\mathbf{Q}_{\text{g}}}, \end{aligned} \tag{4.40}$$

in which the body force $\mathbf{b} = \rho \mathbf{g} = \rho(0, -g)^{\mathsf{T}}$ is given by gravity field and the mass density of the material. Based on the virtual work, the generalized distributed gravity force reads

$$\mathbf{Q}_{\text{g}} = -mg \left( 0, \frac{1}{2}, 0, \frac{l}{12}, 0, 0, 0, \frac{1}{2}, 0, -\frac{l}{12}, 0, 0 \right)^{\mathsf{T}}, \tag{4.41}$$

where the mass of an element is denoted by $m = \rho lhw$.

In addition to the generalized gravity forces, the principle of virtual work is utilized to derive the generalized forces when an external torque $M$ is applied at the cross-section of the beam. The virtual work of the external torque is given by

$$\delta W_{\text{M}} = M \delta \theta, \tag{4.42}$$

where the generalized forces can be obtained by replacing the variation of the rotation angle $\delta\theta$ as a function of the variation of the generalized coordinates $\delta\mathbf{q}$. The orientation of the cross-section can be described by attaching a coordinate system to the cross-section; see Fig. 4.2. The rotation matrix of the attached coordinate system is defined by

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} = \frac{1}{f} \begin{pmatrix} \underbrace{-\frac{\partial r_1}{\partial \eta}}_{\mathbf{r}_{\eta,\perp}} & \underbrace{\frac{\partial r_2}{\partial \eta}}_{\mathbf{r}_{\eta}} \\ \frac{\partial r_2}{\partial \eta} & \frac{\partial r_1}{\partial \eta} \end{pmatrix}, \tag{4.43}$$

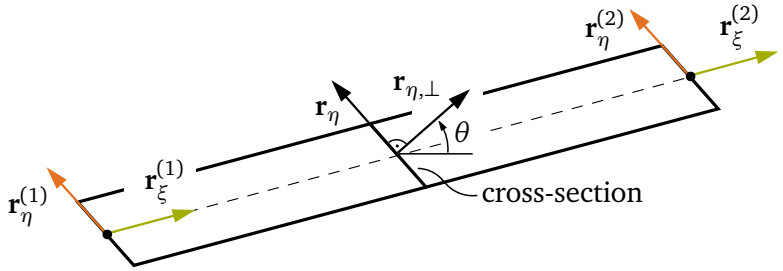



**Figure 4.2:** Orientation of the cross-section for a shear deformable ANCF element

where the orientation of the cross-section is expressed by the transverse slope vector $\mathbf{r}_\eta$ and the orthogonal vector $\mathbf{r}_{\eta,\perp}$ normalized with

$$f = \|\mathbf{r}_\eta\|_2 = \sqrt{\left(\frac{\partial r_1}{\partial \eta}\right)^2 + \left(\frac{\partial r_2}{\partial \eta}\right)^2}. \tag{4.44}$$

From the rotation matrix (4.43), the angle of rotation of the cross-section can be expressed by trigonometric functions as

$$\sin(\theta) = -\frac{1}{f}\frac{\partial r_1}{\partial \eta} \quad \text{and} \quad \cos(\theta) = \frac{1}{f}\frac{\partial r_2}{\partial \eta}. \tag{4.45}$$

The trigonometric functions can be used to formulate the variation of the rotation angle given by

$$\delta\theta = \frac{\frac{\partial r_2}{\partial \eta}\delta\left(\frac{\partial r_1}{\partial \eta}\right) - \frac{\partial r_1}{\partial \eta}\delta\left(\frac{\partial r_2}{\partial \eta}\right)}{f^2}. \tag{4.46}$$

In case a concentrated torque $M$ is applied at the first node $\mathbf{r} = \mathbf{r}^{(1)}$ and/or second node $\mathbf{r} = \mathbf{r}^{(2)}$, the variation of the rotation angle can be formulated as

$$\delta\theta = \frac{\frac{\partial r_2^{(1)}}{\partial \eta}\delta\left(\frac{\partial r_1^{(1)}}{\partial \eta}\right) - \frac{\partial r_1^{(1)}}{\partial \eta}\delta\left(\frac{\partial r_2^{(1)}}{\partial \eta}\right)}{\left(f^{(1)}\right)^2} \quad \text{and/or} \quad \delta\theta = \frac{\frac{\partial r_2^{(2)}}{\partial \eta}\delta\left(\frac{\partial r_1^{(2)}}{\partial \eta}\right) - \frac{\partial r_1^{(2)}}{\partial \eta}\delta\left(\frac{\partial r_2^{(2)}}{\partial \eta}\right)}{\left(f^{(2)}\right)^2}, \tag{4.47}$$

respectively. The variations of the rotation angle evaluated at the nodes can be formulated as a function of the generalized coordinates. Therefore, it follows that

$$\delta\theta = \mathbf{c}^\mathsf{T}\delta\mathbf{q}, \tag{4.48}$$

where $\mathbf{c}$ maps the variation of the generalized coordinates to the variation of the rotation angle. In case a torque is applied at the first node $\mathbf{r} = \mathbf{r}^{(1)}$, the coefficient vector reads

$$\mathbf{c} = \frac{1}{q_5^2 + q_6^2}\left(0, 0, 0, 0, q_6, -q_5, 0, 0, 0, 0, 0, 0\right)^\mathsf{T}, \tag{4.49}$$

and the coefficient vector $\mathbf{c}$ for a torque applied at the second node $\mathbf{r} = \mathbf{r}^{(2)}$ reads

$$\mathbf{c} = \frac{1}{q_{11}^2 + q_{12}^2}\left(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, q_{12}, -q_{11}\right)^\mathsf{T}. \tag{4.50}$$

Substituting the variation of the rotation angle (4.48) into the virtual work (4.42) leads to the generalized forces due to a torque $M$ given by

$$\delta W_{\mathrm{M}} = \delta \mathbf{q}^{\mathrm{T}} \underbrace{M\mathbf{c}}_{\mathbf{Q}_{\mathrm{M}}}. \tag{4.51}$$

Hence, generalized forces due to an applied control $M = u$ and a damping $M = -d\dot{\theta}$ are given by

$$\mathbf{Q}_{\mathrm{u}} = u\mathbf{c} \quad \text{and} \quad \mathbf{Q}_{\mathrm{u}} = -d\dot{\theta}\mathbf{c}, \tag{4.52}$$

respectively. The applied control $u$ drives the ANCF element, and the viscous damping is considered to model joint friction. When modeling a flexible body with multiple ANCF elements, the generalized forces due to the control and damping are considered only at a node kinematically constrained to the ground or at nodes representing mechanical joints for coupling two bodies; see the examples in [101, 102].

## 4.3 Element Assembly

The equations of motion formulated in the previous section are derived for a single ANCF element. The continuum of flexible bodies is modeled by finite elements, which must be properly connected at the nodes. Following [140], the element connectivity can be obtained by eliminating redundant generalized coordinates at the connection nodes. To this end, the element-specific Boolean transformation matrix $\mathbf{T}^{(e)}$ maps the independent generalized coordinates $\mathbf{q}$ of all nodes to its local element representation of an element $(e)$

$$\mathbf{q}^{(e)} = \mathbf{T}^{(e)}\mathbf{q}. \tag{4.53}$$

The equations of motion for the assembled multibody system are obtained in analogy to a single finite element described in Section 4.2.2. Based on the formulation in (4.22), the weak form of the assembled multibody system reads

$$\sum_{(e)} \delta \mathbf{q}^{(e)^{\mathrm{T}}} \left( \mathbf{M}^{(e)} \ddot{\mathbf{q}}^{(e)} + \mathbf{Q}^{(e)}_{\mathrm{elast}} - \mathbf{Q}^{(e)}_{\mathrm{ext}} \right) = 0 \quad \forall \delta \mathbf{q}^{(e)}. \tag{4.54}$$

The equations of motion can not be obtained directly from the weak form in (4.54) since the generalized coordinates $\mathbf{q}^{(e)}$ of an element are not independent. Thus, the variations of the generalized coordinates and the generalized accelerations of an element are expressed in terms of the independent generalized coordinates using (4.53) as

$$\delta \mathbf{q}^{(e)} = \mathbf{T}^{(e)} \delta \mathbf{q} \quad \text{and} \quad \ddot{\mathbf{q}}^{(e)} = \mathbf{T}^{(e)} \ddot{\mathbf{q}}, \tag{4.55}$$

respectively. Substituting the expressions in (4.55) into the weak form in (4.54) yields

$$\delta \mathbf{q}^{\mathrm{T}} \left[ \sum_{(e)} \left( \mathbf{T}^{(e)^{\mathrm{T}}} \mathbf{M}^{(e)} \mathbf{T}^{(e)} \ddot{\mathbf{q}} + \mathbf{T}^{(e)^{\mathrm{T}}} \mathbf{Q}^{(e)}_{\mathrm{elast}} - \mathbf{T}^{(e)^{\mathrm{T}}} \mathbf{Q}^{(e)}_{\mathrm{ext}} \right) \right] = 0 \quad \forall \delta \mathbf{q}, \tag{4.56}$$

which holds for all $\delta \mathbf{q}$ if the body motion is unconstrained. Consequently, the global mass matrix is assembled by

$$\mathbf{M} = \sum_{(e)} \mathbf{T}^{(e)^{\mathrm{T}}} \mathbf{M}^{(e)} \mathbf{T}^{(e)}, \tag{4.57}$$

and the assembled vector of generalized forces due to elastic and external forces reads

$$\mathbf{Q}_{\text{elast}} = \sum_{(e)} \mathbf{T}^{(e)^{\mathsf{T}}} \mathbf{Q}_{\text{elast}}^{(e)} \quad \text{and} \quad \mathbf{Q}_{\text{ext}} = \sum_{(e)} \mathbf{T}^{(e)^{\mathsf{T}}} \mathbf{Q}_{\text{ext}}^{(e)}, \tag{4.58}$$

respectively. The dynamics of the assembled flexible multibody system is defined by second-order differential equations given by

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Q}_{\text{elast}} = \mathbf{Q}_{\text{ext}}, \tag{4.59}$$

Similar to a single ANCF element, the assembled system can be transformed into first-order differential equations by introducing the generalized velocities $\mathbf{v} = \dot{\mathbf{q}}$. Hence, the first-order differential equations reads

$$\dot{\mathbf{x}} = \mathbf{f} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{Q}_{\text{ext}} - \mathbf{Q}_{\text{elast}} \end{pmatrix}, \tag{4.60}$$

wherein the state variables of all nodes are expressed by $\mathbf{x}^{\mathsf{T}} = \left(\mathbf{q}^{\mathsf{T}}, \mathbf{v}^{\mathsf{T}}\right)$.

Chapter 5

# Scientific Contributions

This dissertation aims to provide novel approaches for efficient and accurate sensitivity analysis for first-order differential equations. The scientific contributions focus on deriving advanced formulations in sensitivity analysis using the adjoint variable method, highlighting their efficiency and applicability to large-scale optimization problems.

This chapter summarizes the scientific contributions of this dissertation based on three peer-reviewed publications. Two publications [101, 102] are published in the journal Multibody System Dynamics, and one publication [99] is published in the IUTAM Bookseries Optimal Design and Control of Multibody Systems. A brief outline and highlights of the scientific contributions are given in the next section. In addition, each publication is summarized, and the author's contributions are presented in the following sections.

## 5.1 Outline of Publications

**Publication I** [102] and **Publication II** [99] are concerned with time-optimal control problems. The focus is on using adjoint variables in the context of an NLP problem. An example of the publications include the time-optimal control problem of a flexible robotic system. The promising results motivate the combination of optimal control and structural optimization of flexible multibody systems in a combined optimization. To this end, **Publication III** proposes a discrete adjoint gradient approach to efficiently compute sensitivities of equality and inequality constraints in dynamics, e.g., final constraints on state variables and/or stress restrictions of the flexible components.

Note that the arrangement of **Publication I-III** follows a chronological and methodological order. A reprint of the publications can be found in Appendix B; see **Publication I** in Appendix B.1, **Publication II** in Appendix B.2, and **Publication III** in Appendix B.3.

## 5.2 Summary of Publications

This section summarizes the scientific contributions to the dissertation. In addition, the author's contributions to the **Publications I-III** are provided according to the Contributor Roles Taxonomy (CRediT) defined in [1].

### 5.2.1 Publication I

#### Bibliographic Information

Lichtenecker, D., Rixen, D., Eichmeir, P., and Nachbagauer, K. “On the use of adjoint gradients for time-optimal control problems regarding a discrete control parameterization”. In: *Multibody System Dynamics* 59.3 (2023), pp. 313–334. DOI: 10.1007/s11044-023-09898-5

#### CRediT Author’s Contributions Statement

**Lichtenecker, D.:** Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft and Review & Editing, Visualization, Project Administration. **Rixen, D.:** Resources, Writing - Review & Editing, Supervision, Funding Acquisition. **Eichmeir, P.:** Conceptualization, Methodology, Validation, Writing - Review & Editing. **Nachbagauer, K.:** Conceptualization, Methodology, Validation, Writing - Review & Editing, Supervision, Project Administration, Funding Acquisition.

#### Summary

**Publication I** proposes an adjoint gradient approach for time-optimal control problems regarding parameterization of the control. Time-optimal control problems aim to manipulate a mechanical system from an initial state to a final state in the shortest possible time. This paper focuses on efficient and accurate computation of first-order gradients in time-optimal control problems using adjoint gradients. In addition, the role of adjoint variables in direct optimization methods is analyzed to reveal a new perspective on the optimality conditions in time-optimal control problems of dynamical systems considering final constraints.

In this paper, the time-optimal control problem is formulated similarly to the problem definition discussed in Section 3.2.2. Upper and lower bounds of the control are considered with a penalty function $P$, while the final state is enforced by an equality constraint $\boldsymbol{\phi}$ at the final time. As proposed by Eichmeir et al. [53], such time-optimal control problems can be solved by using an indirect gradient-based approach which relates the control $\mathbf{u}$ with final constraints $\boldsymbol{\phi}$. **Publication I** extends the indirect gradient method to incorporate adjoint gradients regarding a parameterization of the control. Therein, the continuous control is parameterized by $\mathbf{u}(t) = \mathbf{C}(t)\bar{\mathbf{u}}$, where $\mathbf{C}$ is a time-dependent interpolation function and $\bar{\mathbf{u}}$ is a set of control variables. The proposed adjoint gradients are utilized in the context of direct and indirect solution approaches.

Using the control parameterization, the infinite-dimensional time-optimal control problem is transcribed into a finite-dimensional NLP problem using a direct single shooting approach as discussed in Section 3.3.1. The resulting NLP problem is solved using the SQP implementation provided by the MATLAB optimization toolbox fmincon. As pointed out in Section 3.3.4, an efficient and accurate computation of first-order gradients is essential. The proposed adjoint gradients are provided to the third-party solver, which speeds up the optimization procedure. In addition, the adjoint variables are used to discuss the optimality of the converged NLP solution by introducing a switching function to the corresponding time-optimal control problem. As shown in Fig. 3.4, the switching function can be used for a graphical interpretation of the optimality concerning Pontryagin’s minimum principle.

Two examples of a time-optimal rest-to-rest motion of a Selective Compliance Assembly Robot Arm (SCARA) are analyzed to demonstrate the proposed adjoint gradient approach. In the first example, the SCARA is modeled with two rigid bodies in the two-dimensional space. The second example is intended to demonstrate the applicability of the proposed method to problems where the SCARA is modeled with flexible bodies in the two-dimensional space.

Flexible multibody systems tend to be underactuated, and the OCP becomes more complicated than the OCP of rigid multibody systems. However, the flexible bodies are formulated using the ANCF element proposed by Berzeri and Shabana [11]. A brief introduction to the kinematics and kinetics of the ANCF element used is given in **Publication I**. In both examples studied, the interpolation function $\mathbf{C}$ is formulated to represent a cubic spline interpolation for the set of control variables $\bar{\mathbf{u}}$. Smooth control functions are especially relevant to reduce vibrations of the flexible multibody system in order to fulfill final state constraints.

As discussed in Section 3.2.2, the evolution of the optimal control for time-optimal control problems is of the bang-bang type. An approximation of the bang-bang behavior can be recognized in the results of both examples. In addition, the switching function derived from an indirect optimization perspective is used to relate the results obtained by fmincon to the optimality conditions of an indirect optimization approach. The results obtained with fmincon agree with the optimality conditions of the indirect approach. A comparison of the number of function evaluations required for converged solutions considering either finite-differences or adjoint gradients shows the tremendous advantage of using the proposed approach. The computational effort to obtain a converged solution can be reduced significantly, which is especially relevant for underactuated time-optimal control problems of flexible multibody systems.

### 5.2.2 Publication II

#### Bibliographic Information

Lichtenecker, D., Eichmeir, P., and Nachbagauer, K. “On the usage of analytically computed adjoint gradients in a direct optimization for time-optimal control problems”. In: *Optimal Design and Control of Multibody Systems*. Ed. by Nachbagauer, K. and Held, A. Vol. 42. IUTAM Bookseries. Springer, Cham, 2024, pp. 153–164. DOI: 10.1007/978-3-031-50000-8_14

#### CRediT Author’s Contributions Statement

**Lichtenecker, D.:** Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft and Review & Editing, Visualization, Project Administration. **Eichmeir, P.:** Conceptualization, Methodology, Validation, Writing - Review & Editing. **Nachbagauer, K.:** Conceptualization, Methodology, Writing - Review & Editing, Supervision, Project Administration, Funding Acquisition.

#### Summary

**Publication II** extends the adjoint gradient approach for time-optimal control problems proposed in **Publication I**. The structure of first-order gradients computed by direct differentiation and the adjoint method is addressed to emphasize the advantage of using the proposed adjoint gradients. In addition, the adjoint variables are used to discuss the sensitivity of NLP solutions concerning the refinement of the control parameterization. Furthermore, the paper outlines the workflow for integrating the adjoint gradients into a third-party direct optimization solver.

According to **Publication I**, the time-optimal control problem is formulated by a direct single shooting approach, where the control is parameterized by $\mathbf{u}(t) = \mathbf{C}(t)\bar{\mathbf{u}}$. Thus, the time-optimal NLP problem contains the control variables $\bar{\mathbf{u}}$ and the final time $t_f$ as optimization variables, i.e., $\mathbf{z}^\mathsf{T} = (t_f, \bar{\mathbf{u}}^\mathsf{T})$. However, the used NLP solver provides an optimal set of optimization variables $\mathbf{z} = \mathbf{z}^*$ in case the KKT conditions are sufficiently fulfilled. The KKT

conditions correspond to the finite-dimensional NLP problem and do not provide any information on the optimality of the original infinite-dimensional OCP. Consequently, the KKT conditions for the time-optimal control problem can be satisfied even when considering a relatively low number of control variables, in which the continuous control function can not sufficiently approximate the bang-bang structure for a time-optimal control as discussed in Section 3.2.2. To overcome the latter issue, the basic idea of **Publication I** is used to relate converged NLP solutions to Pontryagin's minimum principle for further discussion on the optimality of an optimal point $\mathbf{z}^*$.

The rigid two-arm SCARA analyzed in **Publication I** is used in this publication to discuss the sensitivity of converged NLP solutions concerning a different number of grid nodes in the range of $k \in \{5, 10, 20, 30, 40, 50\}$ per control. For each discretization $\bar{\mathbf{u}}_k$ with a different number of grid nodes, the NLP problem is solved using the SQP implementation provided by the MATLAB optimization toolbox fmincon. In addition, the proposed adjoint gradients are provided to the solver to speed up the optimization procedure. It can be observed that the optimized time-optimal control functions become a bang-bang structure by increasing the number of control grid nodes. In addition, the final time $t_f^*$ decreases by increasing the number of control grid nodes. Note that the KKT conditions for $k = 5$ control grid nodes per control are fulfilled, but the result is not time-optimal regarding Pontryagin's minimum principle. In contrast, using $k = 50$ control grid nodes per control satisfies Pontryagin's minimum principle sufficiently.

The computational efficiency using the proposed adjoint gradient method to compute first-order sensitivities is highlighted in **Publication II**. Therein, a graphical interpretation of the dimensions using the adjoint method and the direct differentiation method is given. Both approaches require the solution of linear differential equations before the first-order gradients can be computed. The dimensions of the underlying differential equations for the adjoint method do not scale when the number of optimization variables is changed. In contrast, the dimensions of the differential equations for the direct differentiation method are proportional to the number of optimization variables. Thus, the computational effort to compute first-order gradients using the direct differentiation method depends strongly on solving the differential equations, especially for problems with a high number of optimization variables. For the study in **Publication II** regarding a different number of grid nodes $k$ per control, the computational effort to compute the first-order gradients using the proposed adjoint method is almost independent of the number of grid nodes. A slight increase in the computational effort can be observed due to the changing dimension of matrix multiplications when the number of grid nodes per control is increased.

**Publication II** proposes a procedure for using adjoint variables in a direct optimization approach. The adjoint variables are used to compute first-order gradients during the iterative optimization procedure. In addition, the adjoint variables are used to evaluate the optimality of the results obtained from the NLP solver regarding Pontryagin's minimum principle.

### 5.2.3 Publication III

#### Bibliographic Information

Lichtenecker, D. and Nachbagauer, K. "A discrete adjoint gradient approach for equality and inequality constraints in dynamics". In: *Multibody System Dynamics* 61.1 (2024), pp. 103–130. DOI: 10.1007/s11044-024-09965-5

#### CRediT Author's Contributions Statement

**Lichtenecker, D.:** Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft and Review & Editing, Visualization, Project Administration. **Nachbagauer, K.:** Conceptualization, Methodology, Validation, Writing - Review & Editing, Supervision, Project Administration, Funding Acquisition.

#### Summary

**Publication III** proposes a discrete adjoint gradient approach for efficient and accurate computation of sensitivities in dynamic simulations. The presented approach incorporates the computation of first-order gradients of equality and inequality constraints in dynamics. The focus of this paper is to provide an efficient framework to combine an OCP with structural optimization of mechanical systems. Combining both optimization tasks is promising to obtain the best possible mechanical structure regarding an OCP. This paper contributes to solving sophisticated and challenging optimization problems of flexible multibody systems efficiently, e.g., the design and control of soft robotics.

Similar to **Publication I** and **Publication II**, the OCP is formulated using a direct single shooting approach, where the control is parameterized by $\mathbf{u}(t) = \mathbf{C}(t)\bar{\mathbf{u}}$. In addition, a set of parameters $\xi$ is considered to parameterize the mechanical system. Considering an OCP and a structural optimization leads to a combined set of optimization variables $\mathbf{z}^\mathsf{T} = (\xi^\mathsf{T}, \bar{\mathbf{u}}^\mathsf{T})$. The resulting NLP problem is hard to handle, and therefore, an efficient approach to compute first-order gradients becomes even more essential. In this paper, NLP problems are solved using the software package IPOPT, where the proposed discrete adjoint gradients are provided to the solver via an interface.

The discrete adjoint method constructs a set of algebraic equations for the adjoint variables directly from the time-integration scheme applied to the state equations of the mechanical system. In **Publication III**, the discrete adjoint variables are derived for systems formulated by first-order differential equations. The derivation of adjoint-based sensitivities of equality and inequality equality constraints leads to discretized matrix differential equations, which depend on the forward time-integration scheme. **Publication III** demonstrates the proposed discrete adjoint gradient approach for two different time-integration schemes (explicit and implicit Euler method) and highlights efficiency and straightforward applicability. The discrete adjoint gradients are especially suitable for optimization problems involving large-scale models or high-dimensional optimization spaces.

Three numerical examples are investigated to demonstrate the use and the advantage of the proposed discrete adjoint gradient approach. In the first example, the role of discrete adjoint variables in sensitivity analysis is discussed using an academic one-mass oscillator. This simple example enables a deep insight into the proposed adjoint method. The sensitivity analysis shows that the evolution of discrete adjoint variables is not necessarily a smooth function. The second example highlights the computational efficiency using the adjoint gradients within the energy optimal control problem of a nonlinear spring pendulum. The third example exploits the efficient discrete adjoint approach to enable a combined optimal control and design problem of a SCARA modeled with flexible bodies. The components of the SCARA are formulated using the ANCF element proposed by Omar and Shabana [117]. A brief introduction to the kinematics and kinetics of the ANCF element used is given in Section 4.2. The efficiency of the discrete adjoint method is demonstrated by a comparison of the runtime required to compute first-order gradients. The proposed discrete adjoint method significantly reduces the runtime compared to the finite-difference method.

Summarizing, **Publication III** discusses adjoint-based sensitivity analysis for dynamic systems in gradient-based optimization problems. Deriving the discrete adjoint gradients

is mathematically more laborious than using finite differences or the direct differentiation method. However, the significant reduction of the runtime when using discrete adjoint gradients for the sensitivity analysis justifies the considerable preprocessing effort to derive the adjoint gradients. This paper presents a novel discrete adjoint gradient approach for the sensitivity analysis of equality and inequality constraints in dynamics. Moreover, the paper demonstrates the application of two different time-integration schemes, highlighting their efficiency and applicability to large-scale optimization problems.

Chapter 6

# Closure

This thesis focuses on OCPs of flexible multibody systems using the adjoint method for efficient and accurate computation of sensitivities. The treatment of this subject requires a solid mathematical background regarding the general concept of the adjoint method, numerical optimization, and flexible multibody formulation. Sufficient details of these subjects are provided to understand the scientific contributions of this thesis.

In this thesis, OCPs are treated using direct optimization methods. Direct optimization methods include the computation of sensitivities in an iterative solution process. Therein, the sensitivities are crucial for determining a search direction to minimize the cost functional while satisfying equality and inequality constraints. Therefore, an efficient and accurate approach for the computation of sensitivities is advantageous, especially for large-scale optimization problems. To address the requirements of efficiency and accuracy in sensitivity analysis, the adjoint theory is exploited in this thesis to derive analytical approaches for sensitivity analysis concerning different OCP formulations.

To demonstrate the efficiency and applicability of the proposed approaches, optimal trajectories are computed for rest-to-rest problems of mechanical systems modeled with rigid and flexible bodies. Initial and final state conditions are specified, and equality and inequality constraints are considered during the motion of the mechanical system. For example, the optimal motion of a flexible two-arm robot is studied regarding a time-optimal control problem, and the energy-optimal control problem of a nonlinear spring pendulum is analyzed. These practical examples demonstrate the theoretical concepts presented in this thesis and highlight the advantages regarding a significant reduction of the computational cost for solving OCPs.

## 6.1 Conclusion and Discussion

Optimal control of flexible multibody systems is of major concern in the emerging and advanced field of soft robotics that aims to go beyond traditional robot models with rigid components. The compliance and flexibility of these systems often result in highly nonlinear dynamics, which complicates the design and control of such robots. This thesis aims to contribute to the challenging field of optimal control of flexible multibody systems by developing efficient and accurate control strategies, e.g., for applications in soft robotics.

Conclusions on the scientific contributions to this thesis are given below for each publication:

- **Publication I** proposes an adjoint gradient approach for time-optimal control problems formulated as a direct optimization problem. Therein, the continuous control functions are parameterized using a set of control grid nodes, which serve as optimization variables along with the required time to manipulate a mechanical system from an initial

state to a final state. The adjoint variables are exploited to compute first-order gradients of the optimization problem. In addition, the adjoint variables are utilized to discuss the solution of an NLP solver in terms of Pontryagin's minimum principle. The interpretation of the optimality concerning Pontryagin's minimum principle provides essential information on whether the parameterization of the control is sufficient to approximate the solution of the original infinite-dimensional OCP. The proposed approach is demonstrated on the time-optimal control problem of a SCARA modeled with rigid and flexible bodies. The computationally efficient approach to computing first-order gradients allows to study the challenging time-optimal control problem of flexible multibody systems in a time-efficient manner.

- **Publication II** extends the proposed adjoint gradient approach in **Publication I** for time-optimal control problems. It highlights the structure of first-order gradients computed via direct differentiation and the proposed adjoint variable method, emphasizing the benefits of using the adjoint gradient approach. Additionally, the paper examines the sensitivity of converged NLP solutions concerning the refinement of control parameterization by using the adjoint variables.

- **Publication III** presents a discrete adjoint gradient approach for (in)equality constraints in dynamics. The proposed approach provides an efficient and accurate opportunity for sensitivity analysis. The time-integration scheme used to compute the evolution of the state variables implies the backward integration of the adjoint variables. The straightforward application of different time-integration schemes is shown in the paper using the explicit and implicit Euler method. In addition, the paper highlights the applicability of the proposed approach to large-scale optimization problems. The adjoint-based sensitivity analysis enables the computation of a solution to challenging optimization problems concerning numerous (in)equality constraints, such as the combination of optimal control and design in a simultaneous optimization in a time-efficient manner.

The developed approaches for adjoint-based sensitivities are compared with the sensitivities computed by the finite-difference method for verification of the derivations and their implementation. The results of the sensitivity analysis obtained by both approaches are in good agreement, which demonstrates the correctness and validation of the presented adjoint gradients. In terms of computational effort, the new approaches presented in this thesis significantly outperform the numerically computed gradients usually used in standard implementations. Thus, using the developed adjoint-based sensitivities within an optimization framework for solving an OCP significantly reduces the required runtime for a converged solution.

## 6.2 Outlook

This thesis demonstrates the potential of using adjoint variables in OCPs of flexible multibody systems for future developments, e.g., in soft robotics. Future investigations could address the following research directions:

- The proposed adjoint gradient approaches can be extended to incorporate multibody systems formulated by a set of redundant coordinates instead of using a minimal set

of coordinates similar to [113]. The set of redundant coordinates leads to differential-algebraic equations to describe the dynamics of the mechanical system. However, considering differential-algebraic equations requires consistent boundary conditions for the adjoint variables. A similar approach as proposed by Gear et al. [60] can be used to define consistent boundary conditions.

- The studied OCPs in this thesis are formulated as a direct single shooting approach. A natural extension of the proposed approaches is to formulate the OCP as a direct multiple shooting approach. Using the multiple shooting approach decouples the shooting interval from its predecessors with initial conditions for the state variables at the boundaries of the shooting intervals. Consequently, the adjoint variables are also decoupled from preceding shooting intervals with final conditions at the boundaries of the shooting intervals. Thus, this approach enables parallel computing to reduce the runtime for the sensitivity analysis. Kirches et al. [94] and Blonigan and Wang [20] motivate to extend the proposed approaches for a direct multiple shooting approach.

- Similar to the adjoint-based feedback-feedforward control presented in [106], the proposed methods of this thesis can be embedded in a feedback control loop algorithm. The offline solution of the OCP provides the feedforward signal, which can be utilized in a control loop. Using a closed loop control system is especially relevant for practical and experimental implementations on the hardware to increase reliability.

- The adjoint method is a popular approach in machine learning for the training of neural networks, commonly referred to as back propagation [88]. The scientific contributions of this thesis can be utilized for efficient sensitivity analysis of neural ordinary differential equations concerning the training parameters; see [86] for a discussion on the adjoint method for a neural ordinary differential equation network. Using efficient solvers in sensitivity analysis is particularly advantageous for high-dimensional machine learning models [128].

Appendix A

# Notation

This appendix provides some basics of the notational conventions in linear algebra used in this thesis.

## A.1 Vectors and Matrices

In this thesis, the components of vectors and matrices are real numbers. In general, scalar quantities are denoted by lowercase non-bold characters, vectors are denoted by lowercase bold characters, and matrices are denoted by uppercase bold characters. An exception in the notation is, e.g., the scalar cost functional denoted by the uppercase character $J$ to be consistent with the scientific community of optimal control. However, the dimension of vectors of length $N_\mathrm{n}$ is given by $\mathbb{R}^{N_\mathrm{n}}$, and the dimension of $N_\mathrm{n} \times N_\mathrm{m}$ matrices is given by $\mathbb{R}^{N_\mathrm{n} \times N_\mathrm{m}}$.

A vector $\mathbf{f} \in \mathbb{R}^{N_\mathrm{n}}$ is usually considered as a column vector given by

$$\mathbf{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_{N_\mathrm{n}} \end{pmatrix}, \tag{A.1}$$

in which the $i$-th component $f_i,\ i \in \{1, \ldots, N_\mathrm{n}\}$ is a real number. The transpose of the vector $\mathbf{f}$ is the row vector

$$\mathbf{f}^\mathsf{T} = \begin{pmatrix} f_1 & \ldots & f_{N_\mathrm{n}} \end{pmatrix}. \tag{A.2}$$

A matrix $\mathbf{F} \in \mathbb{R}^{N_\mathrm{n} \times N_\mathrm{m}}$ is given by

$$\mathbf{F} = \begin{pmatrix} f_{1,1} & \cdots & f_{1,N_\mathrm{m}} \\ \vdots & \ddots & \vdots \\ f_{N_\mathrm{n},1} & \cdots & f_{N_\mathrm{n},N_\mathrm{m}} \end{pmatrix}, \tag{A.3}$$

in which its components are denoted by the double subscripts as $f_{i,j},\ i \in \{1, \ldots, N_\mathrm{n}\},\ j \in \{1, \ldots, N_\mathrm{m}\}$. The transpose of the matrix $\mathbf{F}$ is denoted by $\mathbf{F}^\mathsf{T}$.

## A.2 Nabla Operator

Let $\mathcal{X} \subseteq \mathbb{R}^{N_\mathrm{x}}$ be an open subset and $f : \mathcal{X} \to \mathbb{R}$ is a differentiable scalar-valued function, i.e., $f \in \mathcal{C}^1$, then the differential operator $\nabla_\mathbf{x}$ defines the first-order gradient with respect to the

vector $\mathbf{x} \in \mathbb{R}^{N_\mathrm{n}}$ given as

$$\nabla_{\mathbf{x}} f = \left(\frac{\partial f}{\partial \mathbf{x}}\right)^{\mathsf{T}} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_{N_\mathrm{n}}} \end{pmatrix} \in \mathbb{R}^{N_\mathrm{n}}, \tag{A.4}$$

where the components $\partial f / \partial x_i$ represent the partial derivatives of $f$ with respect to $x_i$. The first-order gradient of a differentiable vector-valued function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^{N_\mathrm{m}}$ with respect to the vector $\mathbf{x}$ is defined by

$$\nabla_{\mathbf{x}} \mathbf{f} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{\mathsf{T}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_{N_\mathrm{m}}}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_{N_\mathrm{n}}} & \cdots & \frac{\partial f_{N_\mathrm{m}}}{\partial x_{N_\mathrm{n}}} \end{pmatrix} \in \mathbb{R}^{N_\mathrm{n} \times N_\mathrm{m}}, \tag{A.5}$$

which is usually called the Jacobian matrix.

Appendix B

# Publications

## B.1 Publication I

### Bibliographic Information

Lichtenecker, D., Rixen, D., Eichmeir, P., and Nachbagauer, K. “On the use of adjoint gradients for time-optimal control problems regarding a discrete control parameterization”. In: *Multibody System Dynamics* 59.3 (2023), pp. 313–334. DOI: 10.1007/s11044-023-09898-5

### Copyright Information

Multibody System Dynamics (2023) 59:313–334
https://doi.org/10.1007/s11044-023-09898-5

ORIGINAL RESEARCH

# On the use of adjoint gradients for time-optimal control problems regarding a discrete control parameterization

**Daniel Lichtenecker[1] · Daniel Rixen[1] · Philipp Eichmeir[2,3] · Karin Nachbagauer[4,3]**

Received: 29 June 2022 / Accepted: 14 March 2023 / Published online: 4 April 2023


## Abstract

In this paper, we discuss time-optimal control problems for dynamic systems. Such problems usually arise in robotics when a manipulation should be carried out in minimal operation time. In particular, for time-optimal control problems with a high number of control parameters, the adjoint method is probably the most efficient way to calculate the gradients of an optimization problem concerning computational efficiency. In this paper, we present an adjoint gradient approach for solving time-optimal control problems with a special focus on a discrete control parameterization. On the one hand, we provide an efficient approach for computing the direction of the steepest descent of a cost functional in which the costs and the error in the final constraints reduce within one combined iteration. On the other hand, we investigate this approach to provide an exact gradient for other optimization strategies and to evaluate necessary optimality conditions regarding the Hamiltonian function. Two examples of the time-optimal trajectory planning of a robot demonstrate an easy access to the adjoint gradients and their interpretation in the context of the optimality conditions of optimal control solutions, e.g., as computed by a direct optimization method.

✉ D. Lichtenecker
daniel.lichtenecker@tum.de

D. Rixen
rixen@tum.de

P. Eichmeir
philipp.eichmeir@fh-wels.at

K. Nachbagauer
karin.nachbagauer@fh-wels.at

1 TUM School of Engineering and Design, Department of Mechanical Engineering, Chair of Applied Mechanics, Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich, Munich, Germany

2 Institute of Mechanics and Mechatronics, Vienna University of Technology, Getreidemarkt 9/E325, 1060 Wien, Austria

3 Faculty of Engineering and Environmental Sciences, University of Applied Sciences Upper Austria, Stelzhamerstraße 23, 4600 Wels, Austria

4 Institute for Advanced Study, Technical University of Munich, Lichtenbergstraße 2a, 85748 Garching, Germany

**Keywords** Time-optimal control · Adjoint gradient method · Cubic spline parameterization · Hamiltonian function · Nonlinear programming · Sequential quadratic programming

## 1 Introduction

Improving the performance of mechanical systems requires sophisticated optimization strategies to fulfill the high demands of current and future product requirements. In general, two problem formulations can be considered to describe various optimization applications: structural optimization of mechanical components and/or finding an optimal control for nonlinear dynamical systems [37]. The focus of this paper is on the latter problem and relates particularly with time-optimal controls.

Bobrow et al. [4] solved the time-optimal control problem for the case where the path is specified and the actuator torque limitations for the robot control are known. They computed the optimal control torques using a conventional linear feedback control system. Shin and McKay [35] used dynamic programming considering parametric functions to reduce the state space. Regarding the application to industrial robots, smooth trajectory planning is essential and has been presented, e.g., by Constantinescu and Craft [8] by using cubic splines to parameterize the state-space trajectory. Reiter et al. [31] proposed a time-optimal path tracking problem formulated as a nonlinear programming (NLP) problem solved by a multiple shooting method to account for the continuity required to respect the technological limits of real robots. Moreover, methods for the fast computation of optimal solutions to planning problems with changing parameters based on B-spline parameterization are presented in [30].

An alternative to the methods mentioned above is the use of indirect optimization methods avoiding the solution of a boundary value problem suffering from poor initial controls. The fundamental work by Bryson and Ho [6] shows how the gradient in an indirect optimization approach can be computed in a straightforward manner using adjoint variables. Optimal control problems can be solved iteratively with adjoint gradients using nonlinear optimization routines, as described in the sense of optimal control or parameter identification in multibody systems, e.g., in [23].

The adjoint method has been used by various authors [2, 14, 17] in different research areas. In the last few decades, the sensitivity analysis based on the adjoint method has become increasingly important [7, 26]. An extensive literature review in a more recent work [15] presents various gradient-based optimization methods, especially in design optimization of flexible multibody systems.

Moreover, since the adjoint method is computationally efficient, real-time applications and neural network applications can be addressed with the adjoint approach. For instance, physics-informed neural networks use partial differential equations in the cost functions to incorporate prior scientific knowledge. Previous research has shown that the discrete adjoint approach is efficient in application of neural networks [29]. Solvers capable of building efficient gradients are beneficial for training machine learning embedded cost functionals. Moreover, Gholami et al. [12] proposed an adjoint-based neural ordinary differential equation framework that provides unconditionally accurate gradients. Johnston and Patel [18] stated that adjoint methods are used in both control theory and machine learning to efficiently compute gradients of functionals.

In this paper, we concentrate on the efficient computation of gradients in optimal control problems and the role of adjoint variables in the optimization strategy. The proposed ideas

can also be used in neural network approaches for the interpretation and evaluation of optimal solutions. In particular, we discuss a particular class of time-optimal control problems for dynamic systems involving final constraints. For such problems, Eichmeir et al. [10] recently presented an indirect gradient method to relate the control variations to the error of the given final constraints. In this paper, we extend the method to provide an exact gradient for discrete control parameterizations for direct or indirect optimization methods. For both methods, the Hamiltonian of the system can be determined and can then be considered for classical statements about optimality. To be more precise, we investigate the role of the adjoint variables in the verification of the optimality conditions of a solution derived by an arbitrary optimization approach, e.g., computed by a direct optimization method.

To this end, we discuss the time-optimal control problem of a two-arm robot. In a first example, the robot is formulated with rigid bodies. The advantage of the proposed approach, in particular, concerning computational effort, will be exploited in a second example of a flexible robotic system using the absolute nodal coordinate formulation (ANCF) for describing large deformations. Both examples are solved by a direct optimization method, and the evaluation of the optimality criteria regarding a Hamiltonian leads to an interpretation of the adjoint gradients. In addition, we performed a comparison of the number of function evaluations in a local minimum considering either numerical or analytical gradients, which shows the clear advantage of using adjoint gradients.

## 2 Time-optimal control problem

The aim of an optimal control problem is to find a control of a dynamical system to minimize certain performance measures. Let us consider the nonlinear dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad \text{with} \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{R}^m$ are the vectors of state and control variables, respectively. A performance measure might be the energy consumption or the operation time from initial to final state. The focus in this work will be on the latter.

We briefly discuss the time-optimal control problem [3, 6, 21]. The goal is to determine a final time $t_f = t_f^*$ and a control $\mathbf{u}(t) = \mathbf{u}^*$ such that the scalar cost functional

$$J(\mathbf{x}(t), \mathbf{u}(t), t_f) = \int_{t_0}^{t_f} \Big[1 + P(\mathbf{x}(t), \mathbf{u}(t))\Big] \, \mathrm{d}t \tag{2}$$

is minimized while satisfying the final constraints

$$\boldsymbol{\phi}(\mathbf{x}(t_f), t_f) = \mathbf{0} \in \mathbb{R}^q. \tag{3}$$

Note that the state variables $\mathbf{x}(t)$ and control $\mathbf{u}(t)$ are functions over the time interval $t \in [t_0, t_f]$ in the cost functional and that inequality constraints for the state variables and control are accounted for using the penalty function $P(\mathbf{x}, \mathbf{u})$. This means that a penalty term will be added to the final time $t_f$ in case of violating inequality constraints; see Eq. (2). The optimal control problem is defined by Eqs. (1)–(3) and can be solved by two different approaches. *Direct* methods transform the original infinite-dimensional optimization problem into a finite-dimensional one. The resulting NLP problem can be treated by well-known methods, e.g., the sequential quadratic programming (SQP) approach [24]. An optimal point

is in accordance with the Karush–Kuhn–Tucker (KKT) conditions [19, 22], which are necessary optimality conditions for direct optimization methods. Contrary, *indirect* methods are based on the calculus of variations and the derived necessary optimality conditions. These explicit expressions result from Pontryagin's minimum principle and are in the form of a two-point boundary value problem, which is usually hard to solve. Shooting methods, collocation methods, or gradient-based approaches can be used to solve the underlying two-point boundary value problem.

### 2.1 Direct optimization method

Direct methods transform the original infinite-dimensional optimization problem into a finite-dimensional one by a parameterization of the control $\mathbf{u}(t)$. In this paper, the control is discretized by a set of grid nodes $\bar{\mathbf{u}}$. Hence the set of optimization variables in time-optimal control problems is defined by $\mathbf{z}^\mathsf{T} = \left(t_f, \bar{\mathbf{u}}^\mathsf{T}\right) \in \mathbb{R}^z$. The discretization of the control leads to an NLP problem of the form

$$\begin{aligned} &\min_{\mathbf{z}} \; J(\mathbf{z}) \\ &\text{s.t. } \boldsymbol{\phi}(\mathbf{z}) = \mathbf{0}. \end{aligned} \tag{4}$$

Note that the NLP problem is defined by the optimization variables $\mathbf{z}$. To evaluate the cost functional and final constraints, the state variables $\mathbf{x}$ have to be computed with respect to the optimization variables $\mathbf{z}$. For solving the state equations in this step, a classical ordinary differential equation (ODE) solver can be used, e.g., an explicit Runge–Kutta solver. The NLP problem in Eq. (4) can be solved by classical direct optimization methods. The local optimality of direct methods is investigated by introducing the Lagrangian function

$$\mathcal{L}(\mathbf{z}, \boldsymbol{\xi}) = J(\mathbf{z}) - \boldsymbol{\xi}^\mathsf{T} \boldsymbol{\phi}(\mathbf{z}), \tag{5}$$

where $\boldsymbol{\xi} \in \mathbb{R}^q$ is the Lagrange multiplier. The necessary first-order or KKT conditions of the optimization problem in Eq. (4) are formulated as

$$\begin{pmatrix} \nabla_\mathbf{z} \mathcal{L}(\mathbf{z}^*, \boldsymbol{\xi}^*) \\ \nabla_{\boldsymbol{\xi}} \mathcal{L}(\mathbf{z}^*, \boldsymbol{\xi}^*) \end{pmatrix} = \mathbf{0}. \tag{6}$$

One of the most powerful methods for finding a KKT point $\left(\mathbf{z}^*, \boldsymbol{\xi}^*\right)$ of the NLP problem is an SQP approach [16, 25, 28]. The basic idea of this approach is to replace the original NLP problem with a quadratic subproblem. The solution of this subproblem is then used in an iterative method to determine a KKT point satisfying Eq. (6). The quadratic approximation of the cost function and the linear approximation of the constraints lead to

$$\begin{aligned} &\min_{\mathbf{d}_k} \; J(\mathbf{d}_k) \approx J(\mathbf{z}_k) + \nabla_\mathbf{z} J(\mathbf{z}_k)^\mathsf{T} \mathbf{d}_k + \frac{1}{2} \mathbf{d}_k^\mathsf{T} \nabla_{\mathbf{zz}}^2 J(\mathbf{z}_k) \mathbf{d}_k \\ &\text{s.t. } \boldsymbol{\phi}(\mathbf{d}_k) \approx \boldsymbol{\phi}(\mathbf{z}_k) + \nabla_\mathbf{z} \boldsymbol{\phi}(\mathbf{z}_k)^\mathsf{T} \mathbf{d}_k, \end{aligned} \tag{7}$$

where $\mathbf{d}_k = \mathbf{z} - \mathbf{z}_k$ is the minimizer of the quadratic subproblem. In analogy to Eq. (6), the KKT conditions regarding the quadratic-linear model result in the linear system

$$\begin{pmatrix} \nabla_{\mathbf{zz}}^2 \mathcal{L} & -\nabla_\mathbf{z} \boldsymbol{\phi}(\mathbf{z}_k) \\ \nabla_\mathbf{z} \boldsymbol{\phi}(\mathbf{z}_k)^\mathsf{T} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{d}_k \\ \boldsymbol{\xi}_k \end{pmatrix} = \begin{pmatrix} -\nabla_\mathbf{z} J(\mathbf{z}_k) \\ -\boldsymbol{\phi}(\mathbf{z}_k) \end{pmatrix}, \tag{8}$$

which is called the KKT system. The solution of this system is used to update the optimization variables

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_k \mathbf{d}_k \tag{9}$$

for the $(k+1)$th iteration. The method is made more robust by using a step length control parameter $\alpha_k$. The factor is obtained by minimizing a proper merit function. For a detailed description and different variants of the basic SQP approach, we refer to [3, 24], which is quite standard and implemented in various codes in this way. One goal of the present paper is to derive adjoint gradients providing analytically given gradients for an SQP approach. Hence the following section is devoted to the derivation of analytically computed adjoint gradients of the Jacobians in Eq. (8). This approach is especially efficient in the case of a high number of optimization variables. Moreover, the indirect approach is used for the evaluation of the chosen parameterization of the control. The parameterization of the control is correctly chosen with respect to the problem formulation if the necessary optimality conditions based on indirect methods are sufficiently small; see Fig. 2 for a graphical validation in the scope of an example.

## 2.2 Adjoint gradient approach

In this section, we exploit the adjoint gradient approach to evaluate the necessary first-order conditions regarding the Hamiltonian function in an indirect optimization method. For this purpose, we briefly summarize the key idea and theory for determining time-optimal controls for dynamic systems regarding final constraints using an indirect method. Generally, indirect methods are based on optimality conditions and lead to solving the underlying two-point boundary value problem. Instead of solving the two-point boundary value problem, the optimization problem can also be addressed by a gradient-based method, e.g., the Kelley–Bryson method [5, 20]. The key idea is to find a variation of the control to produce the maximum local decrease of the cost functional. To find a minimum of the cost functional, we can simply walk a short distance along the negative gradient of the cost functional. We pursue this method to fulfill the necessary optimality conditions derived by the calculus of variations following the basic ideas in [3, 6, 21].

### 2.2.1 Combined adjoint gradient approach regarding final constraints

The adjoint gradient computation in the presence of final constraints has been presented in a recent work [10], and here we briefly summarize the main steps. To derive the adjoint and influence equations, the cost functional in Eq. (2) is extended by the state equations in Eq. (1) leading to

$$\bar{J} = \int_{t_0}^{t_f} \Big[1 + P(\mathbf{x}, \mathbf{u}) + \mathbf{p}^{\mathsf{T}}\big(\mathbf{f}(\mathbf{x}, \mathbf{u}) - \dot{\mathbf{x}}\big)\Big] \mathrm{d}t \tag{10}$$

for any choice of the adjoint variables $\mathbf{p} \in \mathbb{R}^n$ in the cases where the state equations are satisfied. To compute the variation of the extended cost functional $\bar{J}$, we perform infinitesimal variations of the final time $\delta t_f$ and of the control $\delta \mathbf{u}$, which also cause a variation of the states $\delta \mathbf{x}$ due to the state equations. The resulting variation of $\bar{J}$ leads to a variation $\delta \dot{\mathbf{x}}$, which can be eliminated by integration by parts. Finally, the cost functional can be reduced to

$$\delta \bar{J} = \int_{t_0}^{t_f} \Big(P_{\mathbf{u}} + \mathbf{p}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}}\Big) \delta \mathbf{u} \, \mathrm{d}t + (1 + P_f) \delta t_f \tag{11}$$

in the case where the adjoint variables fulfill the linear time-variant final value problem

$$\dot{\mathbf{p}} = -P_{\mathbf{x}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\mathbf{p} \quad \text{with} \quad \mathbf{p}(t_f) = \mathbf{0}, \tag{12}$$

which is solved backward in time. Note that to compute the adjoint system, the state equations must first be solved forward in time. The subscripts $\mathbf{u}$ and $\mathbf{x}$ denote the partial derivatives with respect to $\mathbf{u}$ and $\mathbf{x}$, and the subscript $f$ indicates the evaluation of quantities at time $t = t_f$.

The optimization problem in Eqs. (1)–(3) does not only consist of minimizing the cost functional, but in addition, the final constraints must be fulfilled. The variations of the cost functional and final constraints will lead to a combined gradient-based approach for updating the final time and controls. As a first step, the final constraints are extended with the state equations, leading to

$$\bar{\boldsymbol{\phi}} = \int_{t_0}^{t_f} \mathbf{R}^{\mathsf{T}}\big(\mathbf{f}(\mathbf{x},\mathbf{u}) - \dot{\mathbf{x}}\big)\mathrm{d}t + \boldsymbol{\phi}(\mathbf{x}(t_f), t_f). \tag{13}$$

Proceeding exactly the same way as above, the variation for the extended final constraints is defined by

$$\delta\bar{\boldsymbol{\phi}} = \int_{t_0}^{t_f} \mathbf{R}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\delta\mathbf{u}\,\mathrm{d}t + \dot{\boldsymbol{\phi}}_f\delta t_f, \tag{14}$$

where $\dot{\boldsymbol{\phi}}_f$ denotes the total time derivative of Eq. (3) evaluated at time $t = t_f$. The influence adjoint variables $\mathbf{R} \in \mathbb{R}^{n\times q}$ have to fulfill the matrix differential equation

$$\dot{\mathbf{R}} = -\mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\mathbf{R} \quad \text{with} \quad \mathbf{R}(t_f) = \boldsymbol{\phi}_{\mathbf{x}}^{\mathsf{T}}(\mathbf{x}(t_f), t_f), \tag{15}$$

in which one set of $n$ ordinary differential equations for each component of the final constraints $\boldsymbol{\phi}(\mathbf{x}(t_f), t_f) = \mathbf{0}$ is solved backward in time.

A linear combination of the scalar $\delta\bar{J}$ and vector $\delta\bar{\boldsymbol{\phi}}$ needs the introduction of vector multipliers $\boldsymbol{\nu} \in \mathbb{R}^q$ resulting in

$$\delta\bar{J} + \boldsymbol{\nu}^{\mathsf{T}}\delta\bar{\boldsymbol{\phi}} = \int_{t_0}^{t_f} \underbrace{\Big[P_{\mathbf{u}} + \big(\mathbf{p}^{\mathsf{T}} + \boldsymbol{\nu}^{\mathsf{T}}\mathbf{R}^{\mathsf{T}}\big)\mathbf{f}_{\mathbf{u}}\Big]}_{:=-\delta\mathbf{u}^{\mathsf{T}}}\delta\mathbf{u}\,\mathrm{d}t + \underbrace{\Big(1 + P_f + \boldsymbol{\nu}^{\mathsf{T}}\dot{\boldsymbol{\phi}}_f\Big)}_{:=-\delta t_f}\delta t_f, \tag{16}$$

where $\boldsymbol{\nu}$ is a vector of multipliers to combine both sets of adjoint variables and is computed in such a way that the variations in the control and final time lead to a better approximation of the final constraints. The largest possible decrease of the combined variation is obtained if the variations $\delta\mathbf{u}$ and $\delta t_f$ are chosen in the appropriate descent directions in Eq. (16) for the optimal updates $\mathbf{u}_{\text{new}} = \mathbf{u} + \delta\mathbf{u}$ and $t_{f,\text{new}} = t_f + \delta t_f$. These updates always reduce the cost functional and final constraints within one iteration.

Moreover, the Hamiltonian for the time-optimal control problem in Eqs. (1)–(3) was formulated by

$$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) := 1 + P(\mathbf{x}(t), \mathbf{u}(t)) + \boldsymbol{\lambda}(t)^{\mathsf{T}}\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{17}$$

in which $\boldsymbol{\lambda}(t) = \mathbf{p}(t) + \mathbf{R}(t)\boldsymbol{\nu}$ exploits the decoupling of boundary conditions of the state and adjoint equations by introducing a set of adjoint variables $\mathbf{p}$ and the so-called influence

adjoint variables $\mathbf{R}$ for $q$ final constraints. The decoupling within the multiplier $\boldsymbol{\lambda}$ enables sequential integration of a new set of canonical equations forward and backward in time, depending on a putative optimal control history. The solution of the canonical (adjoint and influence) equations for $\mathbf{p}$ and $\mathbf{R}$ can be combined to determine the Hamiltonian in Eq. (17). A more elaborate derivation can be found in [10], supplemented with examples and convergence analysis.

### 2.2.2 Parameterization of control

The following adjoint gradient approach allows the consideration of final constraints and enables the use of various control parameterizations. Hence the infinite-dimensional optimal control problem is reduced to a finite-dimensional problem to reduce the complexity of the optimization task. As proposed for instance in [10], the variation $\delta\mathbf{u}$ can be defined as an explicit function in time by

$$\delta\mathbf{u} := -P_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\big(\mathbf{p} + \mathbf{R}\boldsymbol{\nu}\big), \tag{18}$$

and only depends on the time discretization of the forward and backward solution. It has to be emphasized that this update $\delta\mathbf{u}$ is not yet dependent on the type of parameterization. To meet the restrictions of industrial applications, feasible control parameterizations have to be specified. However, the control function $\mathbf{u}(t)$ can be parameterized by grid nodes $\bar{\mathbf{u}}$ (see the Appendix for a cubic spline parameterization of the control) in the form

$$\mathbf{u}(t) = \mathbf{C}(t)\,\bar{\mathbf{u}}, \tag{19}$$

leading to the variation of the control

$$\delta\mathbf{u}(t) = \mathbf{C}(t)\,\delta\bar{\mathbf{u}}. \tag{20}$$

It has to be emphasized that any parameterization according to Eq. (19) can be used for the proposed gradient approach. Following the theory introduced in Section 2.2.1, the variation in Eq. (20) is inserted into Eq. (16). Since the variation $\delta\bar{\mathbf{u}}$ does not depend on time, we can rewrite the combined variation in the following form:

$$\delta J + \boldsymbol{\nu}^{\mathsf{T}}\delta\bar{\boldsymbol{\phi}} = \underbrace{\int_{t_0}^{t_f}\Big[\Big(P_{\mathbf{u}} + \boldsymbol{\lambda}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\Big)\mathbf{C}\Big]\,\mathrm{d}t}_{:=-\delta\bar{\mathbf{u}}^{\mathsf{T}}}\,\delta\bar{\mathbf{u}} + \underbrace{\Big(1 + P_f + \boldsymbol{\nu}^{\mathsf{T}}\dot{\boldsymbol{\phi}}_f\Big)}_{:=-\delta t_f}\,\delta t_f. \tag{21}$$

Similarly to the previous section, the variations of the grid nodes $\delta\bar{\mathbf{u}}$ and final time $\delta t_f$ are defined by

$$\delta\bar{\mathbf{u}} = -\kappa\int_{t_0}^{t_f}\mathbf{C}^{\mathsf{T}}\Big(P_{\mathbf{u}}^{\mathsf{T}} + \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\boldsymbol{\lambda}\Big)\,\mathrm{d}t, \tag{22}$$

$$\delta t_f = -\kappa\alpha\Big(1 + P_f + \dot{\boldsymbol{\phi}}_f^{\mathsf{T}}\boldsymbol{\nu}\Big), \tag{23}$$

so that the combined variation leads to the largest possible decrease. The scalar $\kappa$ is a sufficiently small step size, and the factor $\alpha$ can be used as a scaling factor if the magnitudes of the two latter variations differ dramatically. The choice of the vector of multipliers $\boldsymbol{\nu}$ is

based on the reduction of the final constraints in every iteration with the updates $\delta\bar{\mathbf{u}}$ and $\delta t_f$, i.e.,

$$\delta\bar{\boldsymbol{\phi}} := -\varepsilon\,\boldsymbol{\phi}_f, \tag{24}$$

choosing an appropriate update parameter $\varepsilon > 0$, as shown in detail in [10]. This approach finally results in

$$\boldsymbol{\nu} = \mathbf{A}^{-1}\left(\varepsilon\,\boldsymbol{\phi}_f - \mathbf{b}\right), \tag{25}$$

where we use the abbreviations

$$\mathbf{A} := \kappa \int_{t_0}^{t_f} \mathbf{R}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\mathbf{C}\,\mathrm{d}t \int_{t_0}^{t_f} \mathbf{C}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{R}\,\mathrm{d}t + \kappa\alpha\dot{\boldsymbol{\phi}}_f\dot{\boldsymbol{\phi}}_f^{\mathsf{T}}, \tag{26}$$

$$\mathbf{b} := \kappa \int_{t_0}^{t_f} \mathbf{R}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\mathbf{C}\,\mathrm{d}t \int_{t_0}^{t_f} \mathbf{C}^{\mathsf{T}}\left(P_{\mathbf{u}}^{\mathsf{T}} + \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{p}\right)\mathrm{d}t + \kappa\alpha\dot{\boldsymbol{\phi}}_f(1 + P_f). \tag{27}$$

This combined gradient approach enables the use of various parameterizations of the control, shown, e.g., in [10] for a bang-bang control. This paper investigates the combined gradient approach to evaluate necessary optimality conditions regarding the Hamiltonian function.

In case of using a direct optimization method, the corresponding analytically adjoint gradients can be used instead of numerically computed gradients because of the computational burden. To derive these adjoint gradients, the discrete control parameterization from Eq. (20) is inserted into the variation of the cost functional in Eq. (11) and into the variation of the final constraints in Eq. (14). Hence the analytically adjoint gradients are given by

$$\nabla_{\bar{\mathbf{u}}}\, J^{\mathsf{T}} = \int_{t_0}^{t_f} \left(P_{\mathbf{u}} + \mathbf{p}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\right)\mathbf{C}\,\mathrm{d}t, \tag{28}$$

$$\nabla_{\bar{\mathbf{u}}}\, \boldsymbol{\phi}^{\mathsf{T}} = \int_{t_0}^{t_f} \mathbf{R}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\mathbf{C}\,\mathrm{d}t, \tag{29}$$

and can be used instead of the numerical gradients in Eq. (8) in a direct optimization method. Note that the analytical gradients of $J$ and $\boldsymbol{\phi}$ with respect to $t_f$ cannot be simply given in a proper form and are therefore computed numerically in this case. It has to be emphasized that the size of the adjoint system does not increase with a larger number of grid nodes, which is not the case in the direct differentiation method or a numerical differentiation method. If (forward or backward) numerical differentiation is used, then the equations of motion have to be solved $(1 + z)$ times to evaluate the numerical gradients, where $z$ is the number of optimization variables. Hence, the number of forward simulations grows linearly with the number of optimization variables.

Note that the variables $\kappa$, $\alpha$, and $\varepsilon$ are introduced here in accordance to the prework paper [10] but are not relevant in the discussion of the present paper, and therefore the variables are set to $\kappa = \alpha = \varepsilon = 1$. In this paper, the gradients in Eq. (28) and (29) are provided to a direct optimization method without using any user-defined scaling factors. Furthermore, the presented approach provides a new view of adjoint gradients in terms of the evaluation of optimality criteria using the Hamiltonian function.

## 3 Interpretation of the adjoint variables for optimality conditions

The decoupling of the gradients in Section 2.2 can lead to a new perspective on the adjoint variables in the evaluation and interpretation of Pontryagin's principle. Bryson and Ho [6] derive the necessary optimality conditions using the calculus of variations. The minimization of the Hamiltonian must be addressed for restricted optimal controls problems according to Pontryagin's minimal principle [21]. In this paper, we state the Hamiltonian as

$$\mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) := 1 + P(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{30}$$

in which the multiplier $\boldsymbol{\lambda}$ is computed by a linear combination of the adjoint variables

$$\boldsymbol{\lambda} = \mathbf{p} + \mathbf{R}\boldsymbol{v}. \tag{31}$$

Using the introduced Hamiltonian, we can formulate the necessary optimality conditions for time-optimal control problems. The derived variation in Eq. (22) can be rewritten in terms of the Hamiltonian as

$$\delta \bar{\mathbf{u}} = -\int_{t_0}^{t_f} \mathcal{H}_{\bar{\mathbf{u}}}^{\mathsf{T}} \, \mathrm{d}t, \tag{32}$$

where the partial derivative of the Hamiltonian with respect to the grid nodes $\bar{\mathbf{u}}$ is given by

$$\frac{\partial \mathcal{H}}{\partial \bar{\mathbf{u}}} = \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \bar{\mathbf{u}}} = \left( P_{\mathbf{u}} + \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}} \right) \mathbf{C}. \tag{33}$$

In a similar manner, the variation in Eq. (23) can be reformulated as

$$\delta t_f = -\left( \mathcal{H} + \boldsymbol{\phi}_t^{\mathsf{T}} \boldsymbol{v} \right) \Big|_{t=t_f}, \tag{34}$$

where the total time derivative of the final constraints is given by

$$\dot{\boldsymbol{\phi}}_f = \left( \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} + \frac{\partial \boldsymbol{\phi}}{\partial t} \right) \Big|_{t=t_f} = \left( \mathbf{R}^{\mathsf{T}} \mathbf{f} + \boldsymbol{\phi}_t \right) \Big|_{t=t_f}. \tag{35}$$

Note that the partial derivative of the constraints with respect to the states is expressed in terms of the influence adjoint variables given in the final condition in Eq. (15). In this paper, we use the adjoint gradients within the optimality conditions for other optimization methods relating to a Hamiltonian function. An optimal control parameterized by $\mathbf{u} = \mathbf{C}\bar{\mathbf{u}}$ must satisfy the necessary optimality conditions

$$\dot{\mathbf{x}}^* = \quad \mathcal{H}_{\boldsymbol{\lambda}}^{\mathsf{T}}(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*), \qquad t \in [t_0, \, t_f], \tag{36}$$

$$\dot{\boldsymbol{\lambda}}^* = -\mathcal{H}_{\mathbf{x}}^{\mathsf{T}}(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*), \qquad t \in [t_0, \, t_f], \tag{37}$$

$$\mathbf{0} = \int_{t_0}^{t_f} \mathcal{H}_{\bar{\mathbf{u}}}^{\mathsf{T}} \, \mathrm{d}t, \qquad t \in [t_0, \, t_f], \tag{38}$$

$$0 = \mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*) + \boldsymbol{\phi}_t^{\mathsf{T}} \boldsymbol{v}^*, \quad t = t_f, \tag{39}$$

fulfilling the boundary conditions

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{40}$$

$$\mathbf{x}(t_f) = \mathbf{x}_f, \tag{41}$$

$$\boldsymbol{\lambda}(t_f) = \boldsymbol{\lambda}_f. \tag{42}$$

However, since the defined Hamiltonian is an autonomous system, $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*) = 0$ only if an infinite number of grid nodes is used. In addition to the optimality conditions in Eq. (36)–(39), the optimal control history $\mathbf{u}^*$ can be evaluated in the particular case where the control appears linearly in the underlying state equation by introducing the switching function [27]

$$h_i^*(t) = {\mathbf{f}_{u_i}^*}^{\mathsf{T}} \boldsymbol{\lambda}^*, \quad i = 1, \ldots, m. \tag{43}$$

Following Pontryagin's principle, three cases can be observed to express the optimal control

$$u_i^*(t) := \begin{cases} u_{i,\max} & \text{for} \quad h_i^* < 0, \\ u_{i,\min} & \text{for} \quad h_i^* > 0, \\ \text{unspecified} & \text{for} \quad h_i^* = 0, \end{cases} \tag{44}$$

where an infinite number of grid nodes $\hat{u}_i$ is assumed, and the dynamic behavior of the control is of the bang-bang type. Hence Eq. (43) and (44) can be used in a postprocessing step to relate the optimal control $u_i^*$ generated by a direct method with the switching functions $h_i$ to evaluate the solution in terms of an indirect method. If an infinite number of grid nodes is used, then the roots of the switching function exactly match the switching times of the control. Note that the optimization results of the direct approach may yield to so-called singular intervals, where the switching function is zero for a finite time interval, and Pontryagin's principle does not provide any information on the optimal control. However, gradient-based optimization methods are appropriate to identify the control history of singular intervals [9].

### 3.1 Procedure for the use of the adjoint gradients

1. Select an initial final time $t_f$ and initial grid nodes $\bar{\mathbf{u}}$. In time-optimal control problems, it is numerically advantageous to use a normalized time domain $\tau = t/t_f \in [0,\ 1]$. Derivatives with respect to the original time coordinate are given by $\mathrm{d}(\cdot)/\mathrm{d}t = 1/t_f\,(\cdot)'$, where the variable $(\cdot)$ is a function of the normalized time domain $\tau$.
2. Solve the state equations related to initial conditions

$$\mathbf{x}' = t_f\,\mathbf{f}(\mathbf{x}, \mathbf{u}) \quad \text{with} \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{45}$$

   using an ODE solver.
3. Compute the adjoint variables $\mathbf{p}(\tau)$ by solving the linear time-variant final value problem backward in time

$$\mathbf{p}' = -t_f \left( P_{\mathbf{x}}^{\mathsf{T}} + \mathbf{f}_{\mathbf{x}}^{\mathsf{T}} \mathbf{p} \right) \quad \text{with} \quad \mathbf{p}(1) = \mathbf{0}. \tag{46}$$

4. Compute the adjoint variables $\mathbf{R}(\tau)$ using the matrix differential equation

$$\mathbf{R}' = -t_f\,\mathbf{f}_{\mathbf{x}}^{\mathsf{T}} \mathbf{R} \quad \text{with} \quad \mathbf{R}(1) = \boldsymbol{\phi}_{\mathbf{x}}^{\mathsf{T}}(\mathbf{x}(1), 1). \tag{47}$$

5. Compute the adjoint gradients of the cost functional and the final constraints

$$\nabla_{\bar{\mathbf{u}}} J^{\mathsf{T}} = t_f \int_0^1 \left( P_{\mathbf{u}} + \mathbf{p}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}} \right) \mathbf{C} \, \mathrm{d}\tau, \qquad \nabla_{\bar{\mathbf{u}}} \boldsymbol{\phi}^{\mathsf{T}} = t_f \int_0^1 \mathbf{R}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}} \mathbf{C} \, \mathrm{d}\tau. \tag{48}$$

6. Use these analytical gradients for solving the constrained optimization problem by a direct method, e.g., as shown in Section 2.1 by an SQP approach. Repeat steps (2)–(5) until the KKT conditions are satisfied and hence until an optimal solution $\mathbf{z}^{*^{\mathsf{T}}} = \left( t_f^*, \, \bar{\mathbf{u}}^{*^{\mathsf{T}}} \right)$ is found.
7. The states $\mathbf{x}^*$ according to the optimal control grid nodes $\bar{\mathbf{u}}^*$ have to be computed for evaluation of the Hamiltonian in Eq. (30). Moreover, the corresponding $\boldsymbol{\lambda}^*$ is obtained based on Eq. (31).
8. Finally, the optimality conditions in Eqs. (36)–(39) can be evaluated in terms of the Hamiltonian function.

## 4 Numerical examples

To show the use of the adjoint gradient approach in a direct optimization method and to evaluate the optimality conditions in a typical time-optimal control problem, we present two examples of a Selective Compliance Assembly Robot Arm (SCARA) in a rest-to-rest motion. The serial robot consists of two bodies connected with revolute joints and an additional mass is attached to the tool center point (TCP). Firstly, we only consider structural components that are modeled as rigid bodies. Secondly, the rigid bodies are replaced with flexible components. For both examples, the goal is to identify controls $u_1^*$ and $u_2^*$ ($m = 2$) with a continuity requirement up to $\mathcal{C}^2$ such that the TCP moves from a prescribed initial state to a final state in minimal time. Minimizing the cost functional

$$\min_{\mathbf{z}} J = t_f \int_0^1 \Big[ 1 + P(u_1, u_2) \Big] \mathrm{d}\tau \tag{49}$$

$$\text{s.t. } \boldsymbol{\phi}(\mathbf{x}) = \mathbf{0} \tag{50}$$

leads to the shortest operation period $t_f^*$ with respect to physical bounds of the controls, i.e., $-u_{i,\max} \leq u_i \leq u_{i,\max}$. The physical bounds are considered with penalty approach $P(u_1, u_2) := \mu_1 P_1 + \mu_2 P_2$, where the penalty function is given by

$$P_i(u_i) := \begin{cases} 0 & \text{for} \quad |u_i| < u_{i,\max}, \\ \frac{1}{2}(|u_i| - u_{i,\max})^2 & \text{otherwise.} \end{cases} \tag{51}$$

To ensure a continuity requirement up to $\mathcal{C}^2$ of the control $\mathbf{u}$ and to transform the original infinite optimization problem to a finite dimensional one, the interpolation scheme in Eq. (81) in the Appendix is used to parameterize the control. Hence the vector of optimization variables $\mathbf{z}$ consists of the final time $t_f$ and discrete grid nodes $\hat{\mathbf{u}}_i$, i.e., $\mathbf{z}^{\mathsf{T}} = \left( t_f, \, \hat{\mathbf{u}}_1^{\mathsf{T}}, \, \hat{\mathbf{u}}_2^{\mathsf{T}} \right)$. The NLP problem is formulated as direct single shooting and solved by a standard SQP method implemented in the MATLAB function *fmincon*, in which the gradients of the Lagrange function in Eq. (28) and (29) are computed with the proposed adjoint method. In addition, the Hessian of the Lagrange function is computed by a BFGS method.

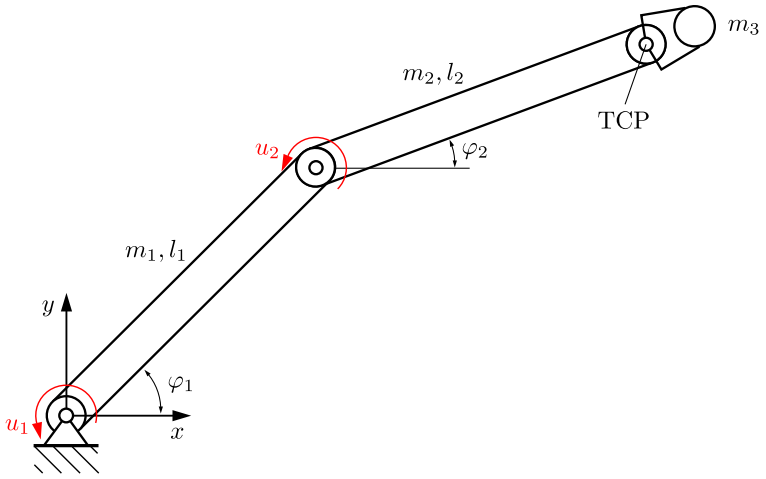



**Fig. 1** Planar two-arm robot with rigid bodies in a general configuration

## 4.1 Rigid SCARA

### 4.1.1 Task description and optimization problem

In the following example, the robot depicted in Fig. 1 is considered in the time-optimal control problem. All structural components are modeled as rigid bodies. The robot is described with a minimal set of generalized coordinates $\varphi_1$ and $\varphi_2$. The system dynamics are obtained with a coupled first-order differential equation by introducing the state variables

$$\mathbf{x} = \left(\varphi_1,\ \varphi_2,\ \omega_1,\ \omega_2\right)^{\mathsf{T}}, \tag{52}$$

where $\dot{\varphi}_i = \omega_i$. This model has been studied by several authors for time-optimal control problems, e.g., in [10]. The mass of the first body and the mass of the TCP is given by $m_1 = m_3 = 1$ kg, the mass of the second body is $m_2 = 0.5$ kg, and the length of both links is $l_1 = l_2 = 1$ m. The moment of inertia of both bodies around their centers of gravity is defined as $J_i = m_i l_i^2/12$.

The cost functional of the optimization problem is given in Eq. (49), and the final constraints read

$$\boldsymbol{\phi}(\varphi_1, \varphi_2, \omega_1, \omega_2) := \left.\begin{pmatrix} l_1\cos(\varphi_1) + l_2\cos(\varphi_2) - x_f \\ l_1\sin(\varphi_1) + l_2\sin(\varphi_2) - y_f \\ \omega_1 \\ \omega_2 \end{pmatrix}\right|_{t=t_f}, \tag{53}$$

where $x_f = 1$ m and $y_f = 1$ m denote the desired final configuration of the TCP in the workspace $\mathcal{W}$. Physical limitations of the controls are considered with the upper bounds $u_{1,\max} = 4$ Nm and $u_{2,\max} = 2$ Nm. Moreover, the weights for the penalty approach are chosen as $\mu_1 = \mu_2 = 10$. The initial conditions of the states are defined by $\mathbf{x}_0 = (-\pi/4,\ 0,\ 0,\ 0)^{\mathsf{T}}$. The control $u_i$ is discretized with $k = 50$ grid nodes and uniform intervals in the normalized time domain $\tau$, i.e., the number of optimization variables is $z = 101$. As an initial guess, the final time is taken as $t_f = 3$ s, and the grid nodes are set to zero, $\bar{\mathbf{u}} = \mathbf{0}$.

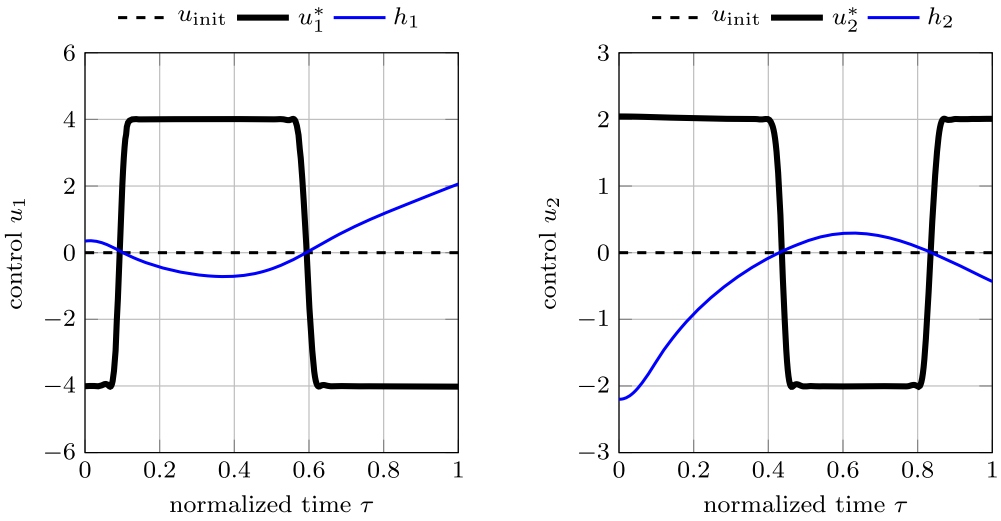



**Fig. 2** Initial controls, optimal controls, and switching functions for a time-optimal rest-to-rest motion of the rigid SCARA model

**Table 1** Comparison of two different approaches to provide gradients in the SQP routine for converged solutions

| Number of grid nodes $k$ | Type of gradient computation | Number of function evaluations |
|---|---|---|
| 5 | forward finite differences | 579 |
| | adjoint gradient method | 107 |
| 50 | forward finite difference | 38584 |
| | adjoint gradient method | 811 |

### 4.1.2 Results

Figure 2 shows the time-optimal control history for both controls in the normalized time domain, where the optimized final time is given by $t_f^* = 1.8294$ s. The results are in accordance with the defined final constraints in Eq. (53), and the controls converge to bang-bang solutions with respect to the control parameterization. Note that the switching function $h_i$ in Eq. (43) is derived from an indirect method and evaluated in terms of the optimal SQP solution. However, the resulting switching function agrees well with the defined control in Eq. (44), and the Hamiltonian of the system is sufficiently small. The optimization result is robust with respect to initial guesses, which implies that the proposed approach converges even if the initial guess is far away from the optimal solution.

Table 1 compares the number of function evaluations when numerical or adjoint gradients are used in the direct optimization method for converged solutions $\mathbf{z}^*$. In the table, the controls are discretized with $k = 5$ and $k = 50$ grid nodes each. We can see that the number of function evaluations in the case of numerical gradients is in general higher when compared to the case where analytical gradients are used. This becomes even more obvious when the number of optimization variables is increased. In addition, the computational cost is reduced for a large number of grid nodes by using adjoint gradients, since the adjoint system does not depend on the number of optimization variables. Note that the number of iterations does not depend on the way the gradients are computed, i.e., both gradients point in the same direction. To be more precise, the number of function evaluations counts the number of evaluations of the cost functional and the number of evaluations of the constraints.

In the case of using finite differences, note that for each evaluation of the constraints, the state equations must be solved. Hence we observe (in general) a high computational effort. In comparison, the function evaluations for analytically derived adjoint gradients are much fewer. Here, although the calculation of the gradient is complicated, the state equations have to solved only once, and just one system of ordinary differential equations (the adjoint system) needs to be solved, which is independent of the number of grid nodes $k$. The numbers in the table are of course problem-dependent and also dependent on the solver settings, thus also on the number of iterations.

### 4.2 Flexible SCARA

Modern robot design includes innovative lightweight techniques to reduce mass and energy consumptions in production lines. Therefore optimal control problems have to be defined for flexible multibody systems, in which the flexible components have to be able to describe large deformations during motion. Multibody systems with flexible components are often underactuated systems, and the optimal control problem becomes more complicated in comparison to fully actuated systems [32].

In this paper, we use the ANCF in the second example examining the effects due to elasticity of the SCARA. The ANCF has been developed particularly for solving large deformation problems in multibody dynamics [34]. Contrary to classical nonlinear finite element approaches used in the literature, the ANCF does not use rotational degrees of freedom and therefore does not necessarily suffer from singularities emerging from angular parameterizations. The most essential advantage of the ANCF is the fact that the mass matrix is constant with respect to the generalized coordinates. The following example is intended to show the applicability of the proposed method for solving time-optimal control problems of highly flexible multibody systems. Therefore we use a standard ANCF element, which is available and has been tested extensively in the literature; see, e.g., [1].

#### 4.2.1 Equations of motion

Based on the proposed ANCF formulation of Berzeri and Shabana [1], a two-node element is described in the global coordinate system with the generalized coordinates

$$\mathbf{q}^{\mathsf{T}} = \left(\mathbf{r}^{(1)^{\mathsf{T}}}, \mathbf{r}_{\chi}^{(1)^{\mathsf{T}}}, \mathbf{r}^{(2)^{\mathsf{T}}}, \mathbf{r}_{\chi}^{(2)^{\mathsf{T}}}\right), \tag{54}$$

where $\mathbf{r}^{(i)}$ denotes the nodal position vector, and $\mathbf{r}_{\chi}^{(i)}$ represents the nodal slope vector of the $i$th node. Figure 3 shows the used ANCF element in a deformed configuration including the generalized coordinates.

An arbitrary point on the undeformed configuration is expressed as $\chi \in [0,\, l]$, where $l$ is the original length of the beam. The position vector of the beam model is defined as

$$\mathbf{r} = \mathbf{S}\mathbf{q}, \tag{55}$$

where the shape function matrix $\mathbf{S}$ maps the generalized coordinates into the global position vector in the reference frame of the workspace $\mathcal{W}$.

The governing equations of a single beam element require the mass matrix and generalized forces. The mass matrix of an element is defined by using the kinetic energy

$$T = \frac{1}{2} m \int_0^1 \dot{\mathbf{r}}^{\mathsf{T}} \dot{\mathbf{r}} \, \mathrm{d}\xi = \frac{1}{2} \dot{\mathbf{q}}^{\mathsf{T}} m \int_0^1 \mathbf{S}^{\mathsf{T}} \mathbf{S} \, \mathrm{d}\xi \, \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^{\mathsf{T}} \mathbf{M} \dot{\mathbf{q}} \tag{56}$$

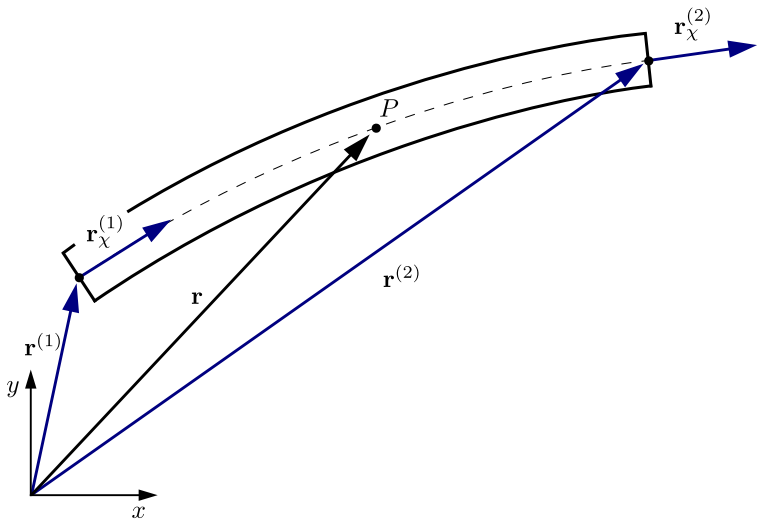



**Fig. 3** ANCF element in a deformed configuration

and the normalized beam length in the undeformed configuration $\xi = \chi / l \in [0, 1]$. Note that the ANCF mass matrix is constant, which leads to numerical advantages. Additionally, the elastic forces vector $\mathbf{Q}_k$, external applied torques $\mathbf{Q}_u$, and damping forces $\mathbf{Q}_d$ have to be defined. The elastic forces vector is defined as

$$\mathbf{Q}_k = \left( \frac{\partial U}{\partial \mathbf{q}} \right)^{\mathsf{T}}, \tag{57}$$

where the strain energy due to longitudinal and bending deformations

$$U = \frac{1}{2} l \int_0^1 E\, A \varepsilon^2 \, \mathrm{d}\xi + \frac{1}{2} l \int_0^1 E\, I \kappa^2 \, \mathrm{d}\xi \tag{58}$$

is used. Here $E$ represents the Young modulus, $A$ is the cross-sectional area, and $I$ is the second moment of area. The curvature $\kappa$ is defined with the Serret–Frenet formulas [13]

$$\kappa = \frac{|\mathbf{r}' \times \mathbf{r}''|}{|\mathbf{r}'|^3}, \tag{59}$$

and the longitudinal strain $\varepsilon$ is formulated with the nonlinear Green–Lagrange strain measure

$$\varepsilon = \frac{1}{2} \left( \mathbf{r}'^{\mathsf{T}} \mathbf{r}' - 1 \right), \tag{60}$$

where

$$\mathbf{r}' = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\chi} = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\xi} \frac{\mathrm{d}\xi}{\mathrm{d}\chi} = \frac{1}{l} \frac{\mathrm{d}\mathbf{S}}{\mathrm{d}\xi} \mathbf{q}. \tag{61}$$

In addition to elastic forces, the principal of virtual work due to a torque $M_i$ acting on the angle of rotation of the cross-section $\theta$

$$\delta W_i = M_i \delta\theta = M_i \frac{\partial \theta}{\partial \mathbf{q}} \delta\mathbf{q} = \mathbf{Q}_i^{\top} \delta\mathbf{q}, \tag{62}$$

leads to the generalized force $\mathbf{Q}_i$. Hence generalized forces associated with control $u$ and damping $f_d = -d\dot{\theta}$ in the revolute joint are given by

$$\mathbf{Q}_u = u \left( \frac{\partial \theta}{\partial \mathbf{q}} \right)^\top , \tag{63}$$

$$\mathbf{Q}_d = f_d \left( \frac{\partial \theta}{\partial \mathbf{q}} \right)^\top , \tag{64}$$

where $d$ is the viscous damping coefficient. Finally, the equations of motion for a single beam element can be obtained as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Q}_k = \mathbf{Q}_u + \mathbf{Q}_d . \tag{65}$$

Introducing the generalized velocities $\mathbf{v} = \dot{\mathbf{q}}$ as additional variables transforms the second-order differential equation for $\mathbf{q}$ into a first-order system

$$\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{v}} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{Q}_u + \mathbf{Q}_d - \mathbf{Q}_k \end{pmatrix} . \tag{66}$$

Instead of using the augmented formulation of Eq. (65) to obtain the equations of motion for $N$ connected elements, it is possible to define an independent set of generalized coordinates to obtain the equations of motion for a constrained multibody system in the form of Eq. (66). Remark that the system Jacobians are calculated with a symbolic toolbox, simplified and factorized to reduce the complex expressions for efficient use. Note that in the two-arm SCARA example, the revolute joint between the first arm and the ground and the revolute joint between the two arms reduce the number of generalized coordinates. The following set of parameters is used in the optimization procedure: the mass of beams $m_1 = m_2 = 2$ kg, the beam length in undeformed configuration $l_1 = l_2 = 1$ m, the viscous damping coefficient $d_1 = d_2 = 0.1$ Nm/rad, the axial stiffness $E_1 A_1 = E_2 A_2 = 300$ N, and the bending stiffness $E_1 I_1 = E_2 I_2 = 3$ Nm$^2$. Moreover, an additional mass attached to the TCP $m_3 = 0.5$ kg is considered, which has to be taken into account in Eq. (56) for the kinetic energy of the second beam.

### 4.2.2 Optimization problem

Similarly to the example in Section 4.1, the cost functional is given in Eq. (49), and the final constraints for the TCP read

$$\boldsymbol{\phi}(\mathbf{x}) := \left. \begin{pmatrix} \mathbf{r}^{(2)} - \mathbf{x}_f \\ \dot{\mathbf{r}}^{(2)} \end{pmatrix} \right|_{t=t_f} , \tag{67}$$

where $\mathbf{x}_f = \left(x_f,\, y_f\right)^\mathsf{T}$ with $x_f = 1$ m and $y_f = 1$ m denotes the desired final configuration of the TCP in the workspace $\mathcal{W}$. The state variables of the remaining nodes are not specified and therefore free. Physical limitations of the controls are considered with the upper bounds $u_{1,\max} = u_{2,\max} = 1$ Nm. The weights for the penalty approach are chosen as $\mu_1 = \mu_2 = 50$. The initial state of the flexible SCARA is defined in the undeformed configuration as in the rigid example with $\varphi_1 = -\pi/4$ rad and $\varphi_2 = 0$ rad. In this example, the control $u_i$ is discretized with $k = 10$ grid nodes and uniform intervals in the normalized time domain $\tau$, i.e., the number of optimization variables is $z = 21$. As an initial guess, the assumption for the final time is $t_f = 5$ s, and the grid nodes are set to zero, $\bar{\mathbf{u}} = \mathbf{0}$.

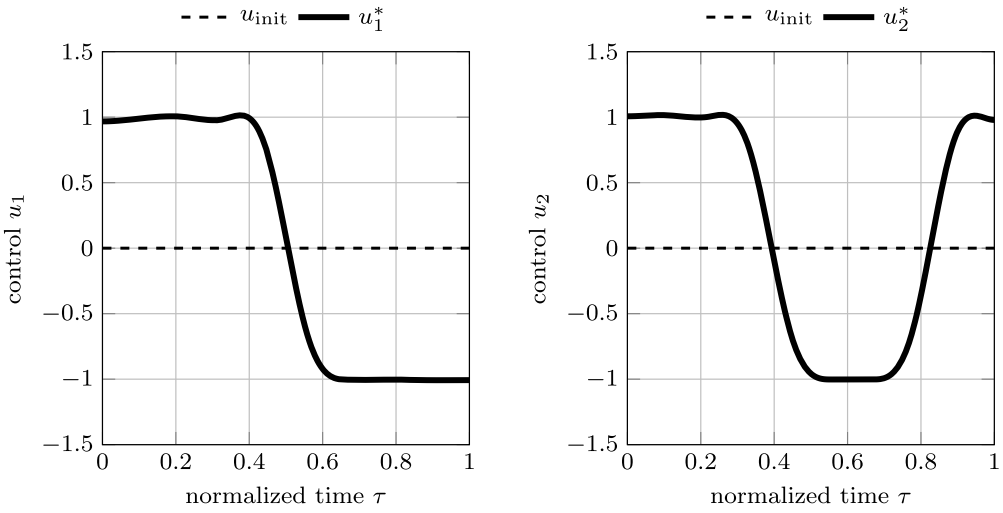



**Fig. 4** Initial and optimal controls for a time-optimal rest-to-rest motion of a flexible SCARA model

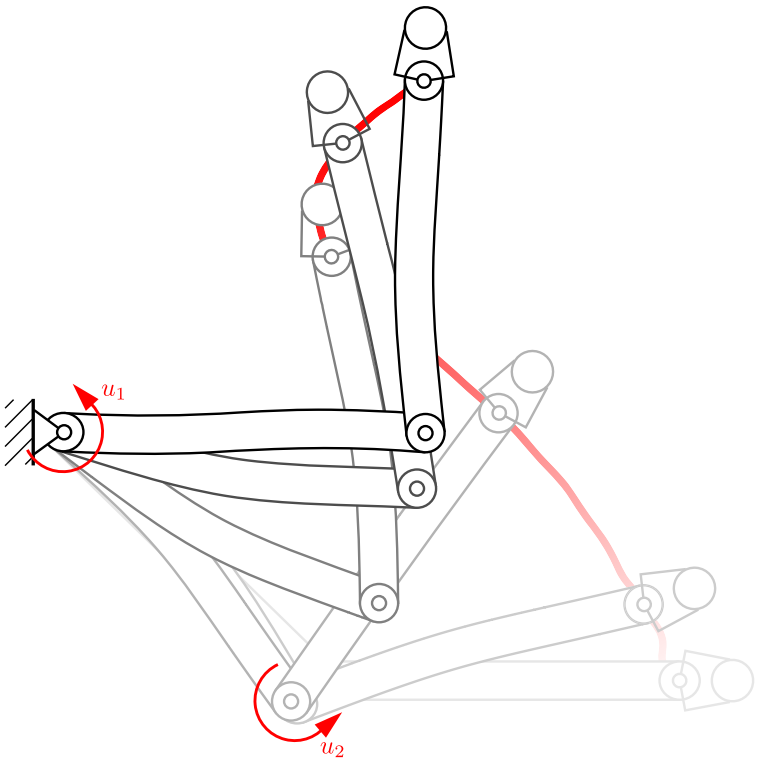



**Fig. 5** Time-optimal control of a flexible two-arm robot for a rest-to-rest maneuver

### 4.2.3 Results

Figure 4 shows the time-optimal control history for both controls in the normalized time domain $\tau$, where the optimized final time is given by $t_f^* = 3.7289$ s. The results are in accordance with the defined final constraints in Eq. (67), and the optimality conditions in Eqs. (36)–(39) are sufficiently small. Snapshots of the time-optimal motion are illustrated in Fig. 5, where the nodal slope vector $\mathbf{r}_\chi^{(i)}$ is scaled for improved representation of the structural flexibility.

## 5 Conclusion and outlook

In this paper, we presented a gradient-based technique to show a new perspective on the optimality conditions in time-optimal control problems of dynamical systems considering final

constraints. Conventional solutions based on a nonlinear programming approach/SQP approach utilize the adjoint variables to assess the optimality conditions regarding the Hamiltonian function. The use of adjoint gradients for a discrete control parameterization is presented in two examples. The gradients are used to replace numerical gradients in a direct optimization method and to evaluate the optimality conditions in terms of a Hamiltonian. The present work illustrates the computational advantage especially by application of the adjoint gradients for discrete control parameterizations in time-optimal control problems of flexible multibody systems including a high number of degrees of freedom. Moreover, the comparison of function evaluations of numerical and adjoint gradients can provide a future perspective for significantly reducing the computational burden when applied to highly dimensional complex multibody system applications.

In a future work, the proposed approach can be extended to formulate the multibody system by a set of redundant coordinates similarly to [23]. Considering differential-algebraic equations requires consistent boundary conditions for the adjoint variables. A similar approach as proposed by Gear, Gupta, and Leimkuhler [11] can be used to overcome this issue.

## Appendix: Formulation of cubic spline parameterization

The original infinite-dimensional optimization problem in Eqs. (1)–(3) has to be transformed into a finite-dimensional one to carry out a direct optimization method. This procedure is usually denoted as a direct transcription method [3]. In general, the literature provides various formulations that can be pursued to perform such a transformation. All methods result in a vector $\mathbf{z}$ to describe the control history. One common method is to carry out a time discretization of the control and an interpolation between the resulting subintervals; e.g., Steiner and Reichl [36] used a linear dependency between the subintervals to minimize a cost functional. A higher-order interpolation scheme is obtained using cubic splines, e.g., in [33]. In the present work, we use a cubic interpolation scheme of the control history $u(t)$:

$$\begin{aligned} u(t) := s_i(t) &= \hat{u}_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3 \\ \text{for} \quad t &\in [t_i,\, t_{i+1}] \quad \text{with} \quad i = 0,\, 1,\, \ldots,\, s-1, \end{aligned} \tag{68}$$

where $s_i$ is the $i$th cubic spline segment for $t \in [t_i,\, t_{i+1}]$, and $s \in \mathbb{N}$ represents the number of piecewise defined spline functions. A given grid node is expressed with $\hat{u}_i$, and $\{b_i,\, c_i,\, d_i\}$ are constant spline parameters associated with the $i$th segment $s_i$. To determine a spline with $\mathcal{C}^2$ continuity, we require that

$$\begin{aligned} s_i(t_{i+1}) &= \hat{u}_{i+1} && \text{with} \quad i = 0,\, 1,\, \ldots,\, s-1, \\ \dot{s}_i(t_{i+1}) &= \dot{s}_{i+1}(t_{i+1}) && \text{with} \quad i = 0,\, 1,\, \ldots,\, s-2, \\ \ddot{s}_i(t_{i+1}) &= \ddot{s}_{i+1}(t_{i+1}) && \text{with} \quad i = 0,\, 1,\, \ldots,\, s-2. \end{aligned} \tag{69}$$

This set of equations leads to a linear system with $h_i := t_{i+1} - t_i$:

$$\begin{aligned} \hat{u}_i + b_i h_i + c_i h_i^2 + d_i h_i^3 &= \hat{u}_{i+1} && \text{with} \quad i = 0,\, 1,\, \ldots,\, s-1, \\ b_i + 2c_i h_i + 3d_i h_i^2 &= b_{i+1} && \text{with} \quad i = 0,\, 1,\, \ldots,\, s-2, \\ 2c_i + 6d_i h_i &= 2c_{i+1} && \text{with} \quad i = 0,\, 1,\, \ldots,\, s-2, \end{aligned} \tag{70}$$

for the unknown spline parameters collected in

$$\mathbf{p}_{\mathrm{b,c,d}} = \left(b_0,\, \ldots, b_{s-1},\, c_0,\, \ldots, c_{s-1},\, d_0,\, \ldots, d_{s-1}\right)^{\mathsf{T}} \in \mathbb{R}^{3s}. \tag{71}$$

Since the number of linear equations $x = 3s - 2$ in Eq. (70) is lower than the number of unknowns $y = 3s$, the linear system is underdetermined. The first and second time derivatives of the splines $s_0(t_0)$ and $s_{s-1}(t_s)$ are still undefined and can be used to determine a unique solution of the spline parameters $\mathbf{p}_{\mathrm{b,c,d}}$. One option is to prescribe the velocity of the first and last spline segment $s_0(t_0) = s_{s-1}(t_s) = 0$:

$$b_0 = 0, \tag{72}$$

$$b_{s-1} = -2c_{s-1}h_{s-1} - 3d_{s-1}h_{s-1}^2. \tag{73}$$

Now the number of linear equations is equal to the number of unknowns, and Eqs. (70)–(73) can be written in the compact form

$$\mathbf{K}\hat{\mathbf{u}} + \mathbf{A}\mathbf{p}_{\mathrm{b,c,d}} = 0, \tag{74}$$

where the vector $\hat{\mathbf{u}} = \left(\hat{u}_0,\, \hat{u}_1,\, \ldots,\, \hat{u}_s\right)^{\mathsf{T}} \in \mathbb{R}^k$ collects all grid nodes with $k = s + 1$. The coefficient matrices $\mathbf{K} \in \mathbb{R}^{3s\times k}$ and $\mathbf{A} \in \mathbb{R}^{3s\times 3s}$ can be simply determined with the underlying Eqs. (70)–(73). However, it is also possible to transform the control history in Eq. (68) into

$$u(t) = \bar{\tau}\,\mathbf{P}\mathbf{p}_{\mathrm{b,c,d}} + \bar{\delta}\hat{\mathbf{u}}, \tag{75}$$

where we use the abbreviations

$$\bar{\tau} = \left(t - t_i,\, (t - t_i)^2,\, (t - t_i)^3\right) \in \mathbb{R}^3, \qquad \mathbf{P} = \begin{pmatrix} \delta & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \delta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \delta \end{pmatrix} \in \mathbb{R}^{3\times 3s}, \tag{76}$$

and

$$\bar{\delta} = (\delta,\, 0) \in \mathbb{R}^k. \tag{77}$$

The Boolean vector $\delta \in \mathbb{R}^s$ picks a certain quantity corresponding to the time interval $t \in [t_i,\, t_{i+1}]$, and the components are defined by

$$\delta_i := \begin{cases} 1 & \text{for} \quad t \in [t_i,\, t_{i+1}] \quad \text{with} \quad i = 0,\, 1,\, \ldots,\, s-1, \\ 0 & \text{otherwise}, \end{cases} \tag{78}$$

using the Kronecker delta, e.g., $\delta = (0,\, 1,\, 0,\, 0)$ for $s = 4$ spline segments in the second time interval. In this sense, the Boolean matrix $\mathbf{P}$ maps all $3s$ spline parameters into those active in the $i$th time interval $t \in [t_i,\, t_{i+1}]$. Now using Eq. (74), the control history can be expressed as a simple vector multiplication

$$u(t) = \mathbf{c}\,\hat{\mathbf{u}} \in \mathbb{R} \tag{79}$$

with the time-dependent vector

$$\mathbf{c}(t) := -\bar{\tau}\mathbf{P}\mathbf{A}^{-1}\mathbf{K} + \bar{\delta} \in \mathbb{R}^k. \tag{80}$$

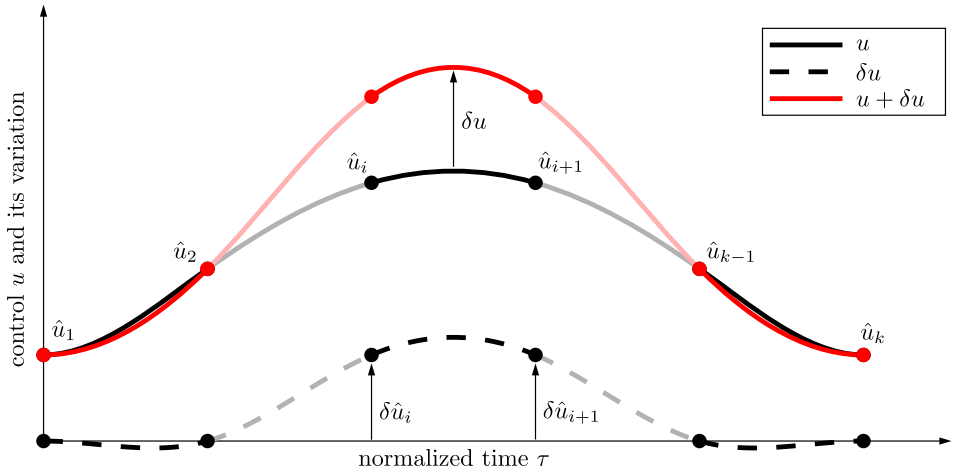



**Fig. 6** Effect of updating grid nodes $\hat{u}_i$ and $\hat{u}_{i+1}$ on the continuous control function $u(t)$

Note that all the information of all spline segments is given in $\mathbf{c}$. Instead of using a single control $u$, generalizations of Eq. (79) for $m$ controls are readily given by

$$\mathbf{u}(t) = \mathbf{C}\,\bar{\mathbf{u}} \in \mathbb{R}^m, \tag{81}$$

where $\bar{\mathbf{u}}^\mathsf{T} = \left(\hat{\mathbf{u}}_1^\mathsf{T}, \ldots, \hat{\mathbf{u}}_m^\mathsf{T}\right) \in \mathbb{R}^{m \cdot k}$ collects all grid nodes $k$ of the $m$ control inputs, and the sparse block diagonal matrix $\mathbf{C}$ reads

$$\mathbf{C}(t) := \begin{pmatrix} \mathbf{c} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{c} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{c} \end{pmatrix} \in \mathbb{R}^{m \times m \cdot k}. \tag{82}$$

The interpolation scheme in Eq. (81) is used in Section 2.2.2 to describe a continuous control history. The variation of grid nodes leads to

$$\delta\mathbf{u}(t) = \mathbf{C}\,\delta\bar{\mathbf{u}}, \tag{83}$$

i.e., an update of the grid nodes $\delta\bar{\mathbf{u}}$ in the optimization procedure leads to an update of the control function $u(t)$, shown in Fig. 6 for a single control. It must be emphasized that all $m$ controls have to be discretized with the same number of grid nodes $k$ and time intervals $[t_i,\, t_{i+1}]$. The discrete control parameterization can be applied for optimal control problems in direct optimization methods, but also in the same manner in indirect optimization methods.

**Acknowledgements** Daniel Lichtenecker and Karin Nachbagauer acknowledge support from the Technical University of Munich – Institute for Advanced Study.

**Funding Note** Open Access funding enabled and organized by Projekt DEAL.

## Declarations

**Competing Interests** The authors declare that they have no conflict of interest.

## References

1. Berzeri, M., Shabana, A.A.: Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation. J. Sound Vib. **235**(4), 539–565 (2000). https://doi.org/10.1006/jsvi.1999.2935
2. Bestle, D., Eberhard, P.: Analyzing and optimizing multibody systems. Mech. Struct. Mach. **20**(1), 67–92 (1992). https://doi.org/10.1080/08905459208905161
3. Betts, J.T.: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2nd edn. SIAM, Philadelphia (2010). https://doi.org/10.1137/1.9780898718577
4. Bobrow, J.E., Dubowsky, S., Gibson, J.S.: Time-optimal control of robotic manipulators along specified paths. Int. J. Robot. Res. **4**(3), 3–17 (1985). https://doi.org/10.1177/027836498500400301
5. Bryson, A.E., Denham, W.F.: Optimal programming problems with inequality constraint II: solution by steepest ascent. AIAA J. **2**(1), 23–34 (1964). https://doi.org/10.2514/3.2209
6. Bryson, A.E., Ho, Y.C.: Applied Optimal Control: Optimization, Estimation, and Control. Taylor & Francis, New York (1975). https://doi.org/10.1201/9781315137667
7. Cao, Y., Li, S., Petzold, L., Serban, R.: Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution. SIAM J. Sci. Comput. **24**(3), 1076–1089 (2003). https://doi.org/10.1137/S1064827501380630
8. Constantinescu, D., Croft, E.A.: Smooth and time–optimal trajectory planning for industrial manipulators along specified paths. J. Robot. Syst. **17**(5), 233–249 (2000). https://doi.org/10.1002/(SICI)1097-4563(200005)17:5<233::AID-ROB1>3.0.CO;2-Y
9. Eichmeir, P., Lauß, T., Oberpeilsteiner, S., Nachbagauer, K., Steiner, W.: The adjoint method for time-optimal control problems. J. Comput. Nonlinear Dyn. **16**(2), 021003 (2021). https://doi.org/10.1115/1.4048808
10. Eichmeir, P., Nachbagauer, K., Lauß, T., Sherif, K., Steiner, W.: Time-optimal control of dynamic systems regarding final constraints. J. Comput. Nonlinear Dyn. **16**(3), 031003 (2021). https://doi.org/10.1115/1.4049334
11. Gear, C.W., Gupta, G.K., Leimkuhler, B.: Automatic integration of Euler-Lagrange equations with constraints. J. Comput. Appl. Math. **12–13**, 77–90 (1985). https://doi.org/10.1016/0377-0427(85)90008-1
12. Gholami, A., Keutzer, K., Biros, G.: ANODE: unconditionally accurate memory-efficient gradients for neural ODEs. ArXiv preprint (2019) https://doi.org/10.48550/arXiv.1902.10298. arXiv:1902.10298
13. Goetz, A.: Introduction to Differential Geometry. Addison Wesley, London (1970)
14. Graichen, K., Petit, N.: A continuation approach to state and adjoint calculation in optimal control applied to the reentry problem. In: Proceedings of the 17th IFAC World Congress, Seoul, Korea, July 6–11, 2008, pp. 14307–14312. (2008). https://doi.org/10.3182/20080706-5-KR-1001.02424
15. Gufler, V., Wehrle, E., Zwölfer, A.: A review of flexible multibody dynamics for gradient-based design optimization. Multibody Syst. Dyn. **53**(4), 379–409 (2021). https://doi.org/10.1007/s11044-021-09802-z
16. Han, S.P.: Superlinearly convergent variable metric algorithms for general nonlinear programming problems. Math. Program. **11**(1), 263–282 (1976). https://doi.org/10.1007/BF01580395
17. Held, A., Seifried, R.: Gradient-based optimization of flexible multibody systems using the absolute nodal coordinate formulation. In: Proceedings of the ECCOMAS Thematic Conference Multibody Dynamics, Zagreb, Croatia, July 1–4 (2013)
18. Johnston, L., Patel, V.: Second-order sensitivity methods for robustly training recurrent neural network models. IEEE Control Syst. Lett. **5**(2), 529–534 (2021). https://doi.org/10.1109/LCSYS.2020.3001498
19. Karush, W.: Minima of functions of several variables with inequalities as side constraints. Master's thesis, Department of Mathematics, University of Chicago (1939)
20. Kelley, H.J.: Method of gradients: optimization techniques with applications to aerospace systems. Math. Sci. Eng. **5**, 205–254 (1962)
21. Kirk, D.E.: Optimal Control Theory: An Introduction. Dover, New York (2004)

22. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492 (1951)
23. Nachbagauer, K., Oberpeilsteiner, S., Sherif, K., Steiner, W.: The use of the adjoint method for solving typical optimization problems in multibody dynamics. J. Comput. Nonlinear Dyn. **10**(6), 061011 (2015). https://doi.org/10.1115/1.4028417
24. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5
25. Ober-Blöbaum, S.: Discrete mechanics and optimal control. PhD thesis, University of Paderborn (2008)
26. Petzold, L., Li, S., Cao, Y., Serban, R.: Sensitivity analysis of differential-algebraic equations and partial differential equations. Comput. Chem. Eng. **30**(10–12), 1553–1559 (2006). https://doi.org/10.1016/j.compchemeng.2006.05.015
27. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mischchenko, E.F.: The Mathematical Theory of Optimal Processes. Wiley, New York (1962)
28. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: Numerical Analysis. Lecture Notes in Mathematics, vol. 630, pp. 144–157 (1978). https://doi.org/10.1007/BFb0067703
29. Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., Edelman, A.: Universal differential equations for scientific machine learning. ArXiv preprint (2020) https://doi.org/10.48550/arXiv.2001.04385. arXiv:2001.04385
30. Reiter, A.: Optimal Path and Trajectory Planning for Serial Robots: Inverse Kinematics for Redundant Robots and Fast Solution of Parametric Problems. Springer, Wiesbaden (2020). https://doi.org/10.1007/978-3-658-28594-4
31. Reiter, A., Müller, A., Gattringer, H.: On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators. IEEE Trans. Ind. Inform. **14**(4), 1681–1690 (2018). https://doi.org/10.1109/TII.2018.2792002
32. Seifried, R.: Dynamics of Underactuated Multibody Systems: Modeling, Control and Optimal Design. Springer, Switzerland (2014). https://doi.org/10.1007/978-3-319-01228-5
33. Seiwald, P., Rixen, D.: Fast approximation of over-determined second-order linear boundary value problems by cubic and quintic spline collocation. Robotics **9**(2), 48 (2020). https://doi.org/10.3390/robotics9020048
34. Shabana, A.A.: Definition of the slopes and the finite element absolute nodal coordinate formulation. Multibody Syst. Dyn. **1**(3), 339–348 (1997). https://doi.org/10.1023/A:1009740800463
35. Shin, K., McKay, N.: A dynamic programming approach to trajectory planning of robotic manipulators. IEEE Trans. Autom. Control **31**(6), 491–500 (1986). https://doi.org/10.1109/TAC.1986.1104317
36. Steiner, W., Reichl, S.: The optimal control approach to dynamical inverse problems. J. Dyn. Syst. Meas. Control **134**(2), 021010 (2012). https://doi.org/10.1115/1.4005365
37. Tromme, E., Held, A., Duysinx, P., Brüls, O.: System-based approaches for structural optimization of flexible mechanisms. Arch. Comput. Methods Eng. **25**(3), 817–844 (2018). https://doi.org/10.1007/s11831-017-9215-6

## B.2 Publication II

### Bibliographic Information

Lichtenecker, D., Eichmeir, P., and Nachbagauer, K. “On the usage of analytically computed adjoint gradients in a direct optimization for time-optimal control problems”. In: *Optimal Design and Control of Multibody Systems*. Ed. by Nachbagauer, K. and Held, A. Vol. 42. IUTAM Bookseries. Springer, Cham, 2024, pp. 153–164. DOI: 10.1007/978-3-031-50000-8_14

### Copyright Information

# On the Usage of Analytically Computed Adjoint Gradients in a Direct Optimization for Time-Optimal Control Problems

Daniel Lichtenecker[1(✉)], Philipp Eichmeir[2], and Karin Nachbagauer[2,3]

[1] Technical University of Munich, Germany; TUM School of Engineering and Design, Department of Mechanical Engineering, Chair of Applied Mechanics, Munich Institute of Robotics and Machine Intelligence (MIRMI), Boltzmannstraße 15, 85748 Garching, Germany
daniel.lichtenecker@tum.de

[2] University of Applied Sciences Upper Austria, Campus Wels, Stelzhamerstraße 23, 4600 Wels, Austria
{philipp.eichmeir,karin.nachbagauer}@fh-wels.at

[3] Institute for Advanced Study, Technical University of Munich, Lichtenbergstraße 2a, 85748 Garching, Germany

**Abstract.** This paper discusses time-optimal control problems and describes a workflow for the use of analytically computed adjoint gradients considering a discrete control parameterization. The adjoint gradients are used here to support a direct optimization method, such as Sequential Quadratic Programming (SQP), by providing analytically computed gradients and avoiding the elaborate numerical differentiation. In addition, the adjoint variables can be used to evaluate the necessary first-order optimality conditions regarding the Hamiltonian function and gives an opportunity to discuss the sensitivity of a solution with respect to the refinement of the discretization of the control. To further emphasize the advantages of adjoint gradients, there is also a discussion of the structure of analytical gradients computed by a direct differentiation method, and the difference in the dimensions compared to the adjoint approach is addressed. An example of trajectory planning for a robot shows application scenarios for the adjoint variables in a cubic spline parameterized control.

## 1 Introduction

Optimal control theory is based on the calculus of variations and deals with finding optimal trajectories for nonlinear dynamical systems, e.g. spacecrafts or multibody systems like robots. The works by Kelley [4] and Bryson and Ho [1] have to be mentioned as groundbreaking in the field of optimal control theory and serve as basis for extensive subsequent research, also in the field of time-optimal control.


K. Nachbagauer and A. Held (Eds.): IUTAM 2022, IUTAM Bookseries 42, pp. 153–164, 2024.
https://doi.org/10.1007/978-3-031-50000-8_14

As a special class of time-optimal control problems considering final constraints, one can cite the control of a robot arm designed in such a way that the operation time for a rest-to-rest maneuver becomes minimal. Following an indirect optimization approach, such problems can be transformed into a two-point boundary value problem, which can usually be solved by shooting or full collocation methods. Alternatively, a direct optimization approach can be pursued, in which the boundary value problem is posed as a nonlinear programming problem method, see e.g. [12] for the time-optimal trajectory planning considering the continuity required to respect technological limits of real robots.

An alternative to the mentioned methods is offered by indirect gradient methods, which are considered to be particularly robust with respect to initial controls. The work by Bryson and Ho [1] shows how the gradient can be computed straightforward using adjoint variables. With this gradient information optimal control problems can be solved iteratively by the use of nonlinear optimization routines, as described in the sense of optimal control or parameter identification in multibody systems e.g. in [8]. The work by Eichmeir et al. [2] extends the theory for time-optimal control problems to dynamic systems under final constraints. Such problems arise e.g. in space vehicle dynamics during minimum time moon ascending/descending maneuvers or in robotics in the case that the time for a rest-to-rest maneuver should become a minimum. Such problems can be considered as two-point boundary value problems, with the major drawback of requiring an initial guess close to the optimal solution. Otherwise, the optimal control problem could be solved via the adjoint method which is an efficient way to compute the direction of the steepest descent of a cost functional. However, when using such indirect methods to solve optimal control problems, a major drawback appears in the computation of the Hamiltonian and the required derivatives: they may be complex and furthermore need to be recomputed often during the simulation. Moreover, depending on the variables or parameters to be identified in the optimal control strategy, it is difficult to assign a physical meaning to the adjoint variables.

This paper focuses on solving time-optimal control problems with a classical direct optimization method and then evaluating the respective optimality conditions based on an indirect optimization approach. The adjoint variables can be investigated to efficiently compute the gradients of the cost functional and the constraints. Moreover, the adjoint variables can be investigated to exploit the optimality conditions regarding the Hamiltonian function. To demonstrate the use of analytically computed adjoint gradients, the time-optimal trajectory planning of a Selective Compliance Assembly Robot Arm (SCARA) is solved by an SQP method and the optimality conditions regarding the Hamiltonian function are evaluated by the adjoints. The application shows the easy access to the adjoint gradients and discusses the latter mentioned role of the adjoint variables in the optimality conditions.

## 2 Use of Adjoint Variables in Direct Optimization Approaches

The aim of this paper is to determine a control $\mathbf{u}(t) = \mathbf{u}^*$ and a final time $t_f = t_f^*$ of a dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad \text{with} \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{1}$$

such that the scalar performance measure

$$J(\mathbf{x}(t), \mathbf{u}(t), t_f) = \int_{t_0}^{t_f} \Big[1 + P(\mathbf{x}(t), \mathbf{u}(t))\Big] \, \mathrm{d}t \tag{2}$$

becomes a minimum with respect to a final constraint

$$\boldsymbol{\phi}(\mathbf{x}(t_f), t_f) = \mathbf{0} \in \mathbb{R}^q. \tag{3}$$

Inequality constraints on the state $\mathbf{x} \in \mathbb{R}^n$ and the control $\mathbf{u} \in \mathbb{R}^m$ are considered by the scalar penalty function $P$. To be specific, violations of inequality constraints within the time interval $t \in [t_0, t_f]$ are taken into account as an additional term in the cost functional in Eq. (2). The above optimal control problem can generally be solved by a direct or indirect optimization approach. In this paper, the original infinite dimensional optimization problem is transformed into a finite dimensional one by parameterizing the control with a finite set of optimization variables $\mathbf{z} \in \mathbb{R}^z$ including the final time and the control parameterization. Thus, the resulting nonlinear programming (NLP) problem can be solved with classical direct optimization approaches such as the SQP method [9].

**How to Interpret the Results from a Direct Optimization Algorithm** An optimal point $\mathbf{z} = \mathbf{z}^*$ fulfills the well-known Karush-Kuhn-Tucker (KKT) conditions [3,5], but these conditions do not provide any information about the quality of the control parameterization with respect to the infinite dimensional optimization problem. The basic idea to interpret an optimal point $\mathbf{z}^*$ is to relate the direct optimization approach to Pontryagin's minimum principle [11]. The optimality conditions based on an indirect optimization approach [2] can be used for this idea. Figure 1 illustrates a rough flowchart for the interpretation of results obtained by a direct optimization approach. This approach requires the Hamiltonian of the system to evaluate Pontryagin's minimum principle. The Hamiltonian for time-optimal control problems related to the cost functional in Eq. (2) can be formulated as

$$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) := 1 + P(\mathbf{x}(t), \mathbf{u}(t)) + \boldsymbol{\lambda}(t)^{\mathsf{T}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{4}$$

in which the multiplier $\boldsymbol{\lambda}(t) = \mathbf{p}(t) + \mathbf{R}(t)\boldsymbol{\nu}$ is computed by a linear combination of the adjoint variables $\mathbf{p} \in \mathbb{R}^n$ and $\mathbf{R} \in \mathbb{R}^{n \times q}$. The vector $\boldsymbol{\nu} \in \mathbb{R}^q$ is a multiplier to combine both adjoint variables. A deep insight into the combination of both adjoint variables is presented in [2]. The Hamiltonian in Eq. (4) is used in Sect. 4 to interpret the results of a time-optimal control problem obtained by a direct optimization approach as depicted in Fig. 1.

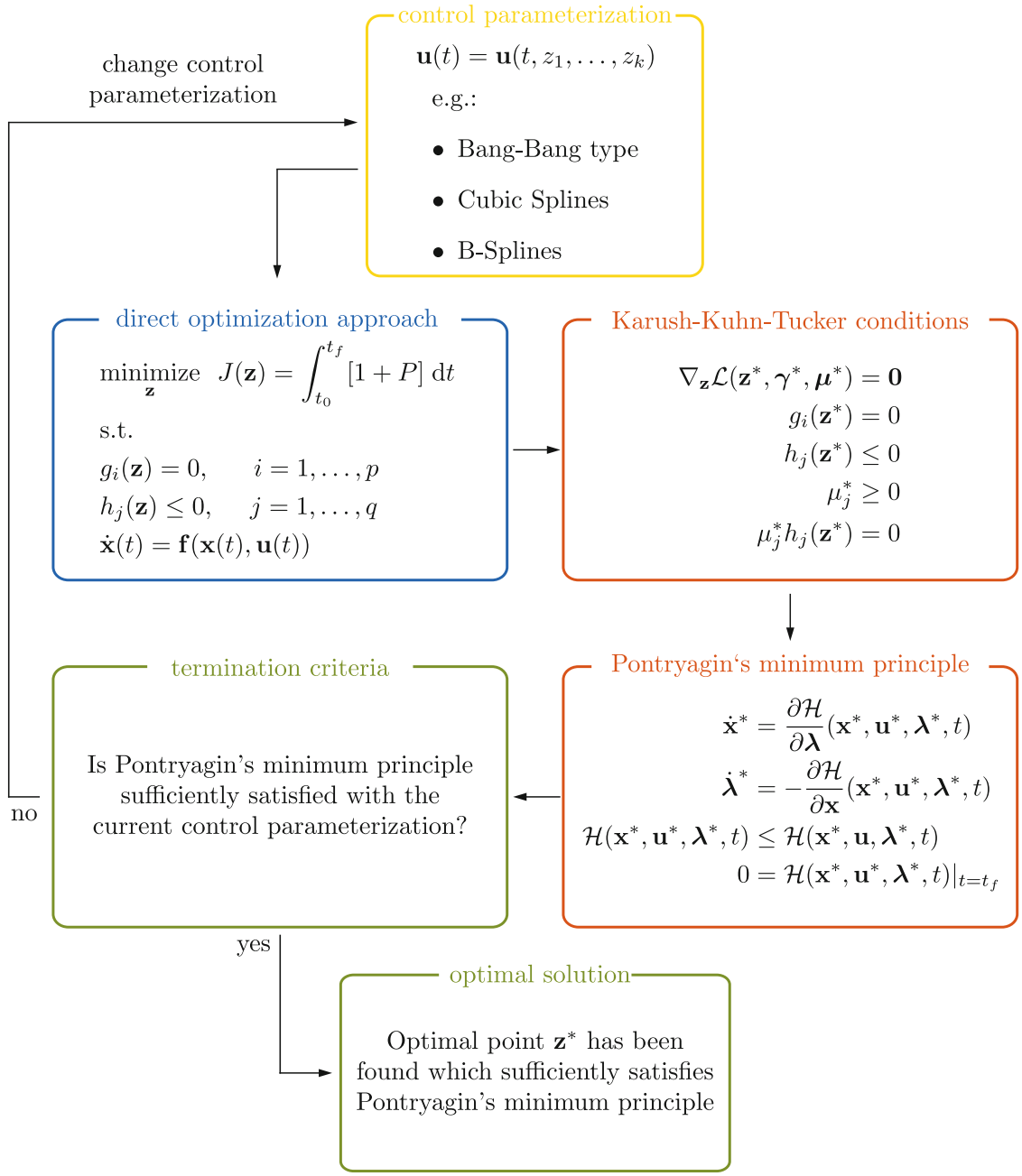



**Fig. 1.** Flowchart to interpret the results from a direct optimization algorithm with Pontryagin's minimum principle

## 3 Computation of First-Order Derivatives

Classical gradient-based optimization algorithms rely on the derivatives of the cost functional and the constraints with respect to the optimization variables $\mathbf{z}$. The computation of these gradients takes a key role in such optimization algorithms and the convergence of the optimization depends on the accuracy of the gradients. In addition to accuracy, efficient computation of gradients is especially important for large numbers of optimization variables. Thus, the computational effort to solve the optimization problem depends significantly on the efficient

computation of gradients. Figure 2 summarizes the most common approaches for the computation of first-order gradients. The *finite differences* method is the easiest approach to code, but suffers in terms of computational effort especially for a large number of optimization variables. In case of using (forward or backward) finite differences, the state equations have to be solved $(1 + z)$ times in order to evaluate the numerical gradients with respect to $z$ optimization variables. Thus, the number of forward simulations grows linearly with the number of optimization variables. In contrast to this numerical approach, the *direct differentiation* and the *adjoint method* are referred as analytical approaches to compute gradients. Both approaches lead to exact gradient information and using them in an optimization scheme leads to an increase in efficiency. The characteristics of the analytical approaches are discussed in the following sections.

### 3.1 Direct Differentiation Approach for Discrete Control Parameterization

The direct differentiation approach is based on the sensitivity of the state equations and is briefly discussed in this section. In this paper, the control is described by $\mathbf{u}(t) = \mathbf{C}\,\bar{\mathbf{u}}$, in which the vector $\bar{\mathbf{u}}^\mathsf{T} = \left(\hat{\mathbf{u}}_1^\mathsf{T}, \ldots, \hat{\mathbf{u}}_m^\mathsf{T}\right) \in \mathbb{R}^{m \cdot k}$ collects $k$ grid nodes of the $m$ equidistant time-discretized controls and the matrix $\mathbf{C}(t) \in \mathbb{R}^{m \times m \cdot k}$ maps the grid nodes to a time dependent function. The interpolation matrix $\mathbf{C}$ has to be determined once *a priori* and depends on the chosen interpolation order [6].

By using this control parameterization, the gradient of the cost functional is directly obtained by differentiating it with respect to the grid nodes as

$$\nabla_{\bar{\mathbf{u}}} J^\mathsf{T} = \int_{t_0}^{t_f} \left[\frac{\partial P}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \bar{\mathbf{u}}} + \frac{\partial P}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \bar{\mathbf{u}}}\right] \mathrm{d}t \tag{5}$$

$$= \int_{t_0}^{t_f} \left[P_\mathbf{x} \mathbf{x}_{\bar{\mathbf{u}}} + P_\mathbf{u} \mathbf{C}\right] \mathrm{d}t, \tag{6}$$

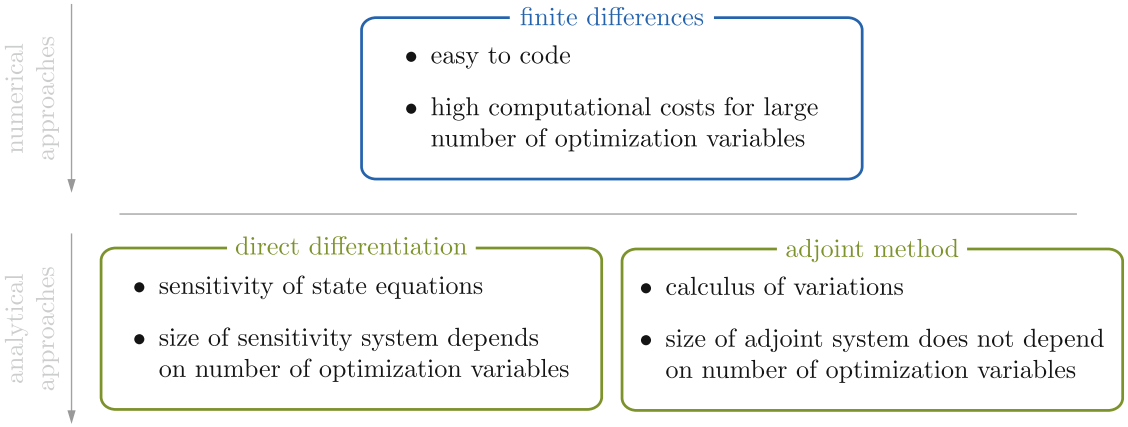



**Fig. 2.** Overview of approaches to compute first-order derivatives

in which the partial derivative of the parameterized control with respect to the grid nodes

$$\frac{\partial \mathbf{u}}{\partial \bar{\mathbf{u}}} = \mathbf{C} \tag{7}$$

has been utilized. Partial derivatives of an arbitrary function $f$ with respect to $x$ are denoted with subscripts, i.e. $f_x$. Similar to the gradient of the cost functional, the gradient of the final constraints in Eq. (3) can be calculated by direct differentiation as

$$\nabla_{\bar{\mathbf{u}}} \boldsymbol{\phi}^{\mathsf{T}} = \boldsymbol{\phi}_{\mathbf{x}} \mathbf{x}_{\bar{\mathbf{u}}}. \tag{8}$$

The resulting gradients in Eq. (6) and Eq. (8) involve the system sensitivity $\mathbf{x}_{\bar{\mathbf{u}}} \in \mathbb{R}^{n \times m \cdot k}$ which is obtained by differentiating the state equations with respect to the grid nodes as

$$\dot{\mathbf{x}}_{\bar{\mathbf{u}}} = \mathbf{f}_{\mathbf{x}} \mathbf{x}_{\bar{\mathbf{u}}} + \mathbf{f}_{\mathbf{u}} \mathbf{u}_{\bar{\mathbf{u}}} \tag{9}$$

$$= \mathbf{f}_{\mathbf{x}} \mathbf{x}_{\bar{\mathbf{u}}} + \mathbf{f}_{\mathbf{u}} \mathbf{C}. \tag{10}$$

Initial conditions of the system sensitivity are defined as

$$\mathbf{x}_{\bar{\mathbf{u}}}(0) = \mathbf{0}, \tag{11}$$

since the initial conditions of the state equations do not depend on the grid nodes, i.e. $\mathbf{x}(0) = \mathbf{x}_0$. The system Jacobian $\mathbf{f}_{\mathbf{x}} \in \mathbb{R}^{n \times n}$ and $\mathbf{f}_{\mathbf{u}} \in \mathbb{R}^{n \times m}$ have to be calculated *a priori*, e.g., by analytical differentiation, in order to solve the matrix differential system in Eq. (10). Remark that the differential system depends on the number of grid nodes. Thus, the computational effort increases with the number of grid nodes.

### 3.2 Adjoint Gradient Approach for Discrete Control Parameterization

A large number of grid nodes leads to a large solution space and, therefore, the gradient computation leads to a high computational effort resulting from finite differences or direct differentiation. An efficient alternative to compute gradients analytically is the adjoint variable method which is based on the calculus of variations. Following the basic idea presented in the seventies by Bryson and Ho [1], an adjoint gradient approach for discrete control parameterizations is utilized. Lichtenecker et al. [6] derived the adjoint gradients for time-optimal control problems defined in Eqs. (1)–(3) for spline control parameterizations in the following form:

$$\nabla_{\bar{\mathbf{u}}} J^{\mathsf{T}} = \int_{t_0}^{t_f} \left( \mathbf{p}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}} + P_{\mathbf{u}} \right) \mathbf{C} \, \mathrm{d}t, \tag{12}$$

$$\nabla_{\bar{\mathbf{u}}} \boldsymbol{\phi}^{\mathsf{T}} = \int_{t_0}^{t_f} \mathbf{R}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}} \mathbf{C} \, \mathrm{d}t, \tag{13}$$

in which the adjoint variables fulfill the (adjoint) system of differential equations

$$\dot{\mathbf{p}} = -P_{\mathbf{x}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\mathbf{p} \qquad \text{with} \qquad \mathbf{p}(t_f) = \mathbf{0}, \tag{14}$$

$$\dot{\mathbf{R}} = -\mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\mathbf{R} \qquad \text{with} \qquad \mathbf{R}(t_f) = \boldsymbol{\phi}_{\mathbf{x}}^{\mathsf{T}}(\mathbf{x}(t_f), t_f). \tag{15}$$

Due to the final conditions, they have to be solved backward in time to compute the adjoint gradients. Moreover, it has to be emphasized that the size of the adjoint system does not grow with the number of grid nodes which is not the case for direct differentiation, see Sect. 3.3. The adjoint gradients in Eqs. (12) and (13) prove to be preferable regarding computational effort and accuracy in gradient based optimization strategies. For further details on adjoint gradients, the reader is referred to [2,6].

**How to Compute the Adjoint Gradients**

The adjoint gradients in Eqs. (12) and (13) can be used for direct and indirect optimization algorithms. Both approaches are iterative methods and, therefore, the gradients have to be recomputed in each iteration. In this paper, we use a direct optimization method in order to compute the optimal control. Similar as shown in [10], Fig. 3 illustrates the application of adjoint gradients provided to a direct optimization method and is summarized with the following steps:

1. Select a direct optimization method which is able to use user-defined gradients, e.g. a classical SQP method or an Interior Point (IP) method.
2. The optimization algorithm proposes values $\mathbf{z}_i$ for the optimization variables associated to the current $i$-th iteration. Starting from this view, the gradients have to be computed for the $(i+1)$-th iteration.
3. Solve the state equations related to the actual optimization variables and initial conditions using an ODE solver.
4. The cost functional and the final constraints can be evaluated.
5. Compute the adjoint variables $\mathbf{p}$ and $\mathbf{R}$ backward in time using Eqs. (14) and (15).
6. Finally, the adjoint gradients of the cost functional and the final constraints are computed by a time integration and provided to the optimization algorithm for the next iteration.
7. Steps (2) through (6) are repeated until the KKT conditions are fulfilled with respect to the optimal solution $\mathbf{z}^*$.

### 3.3 Discussion on Duality of Gradients

McNamara et al. [7] pointed out that the adjoint approach can be interpreted as a special case of linear duality and that the core of this method is based on a substitution of variables. This can be seen by considering the first term of the gradients of the cost functional in Eqs. (6) and (13), i.e.,

$$\int_{t_0}^{t_f} P_{\mathbf{x}}\mathbf{x}_{\bar{\mathbf{u}}}\,\mathrm{d}t \quad \text{with} \quad \dot{\mathbf{x}}_{\bar{\mathbf{u}}} = \mathbf{f}_{\mathbf{x}}\mathbf{x}_{\bar{\mathbf{u}}} + \mathbf{f}_{\mathbf{u}}\mathbf{C} \quad \text{and} \quad \mathbf{x}_{\bar{\mathbf{u}}}(0) = \mathbf{0}, \tag{a}$$

$$\int_{t_0}^{t_f} \mathbf{p}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\mathbf{C}\,\mathrm{d}t \quad \text{with} \quad \dot{\mathbf{p}} = -P_{\mathbf{x}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\mathbf{p} \quad \text{and} \quad \mathbf{p}(t_f) = \mathbf{0}. \tag{b}$$

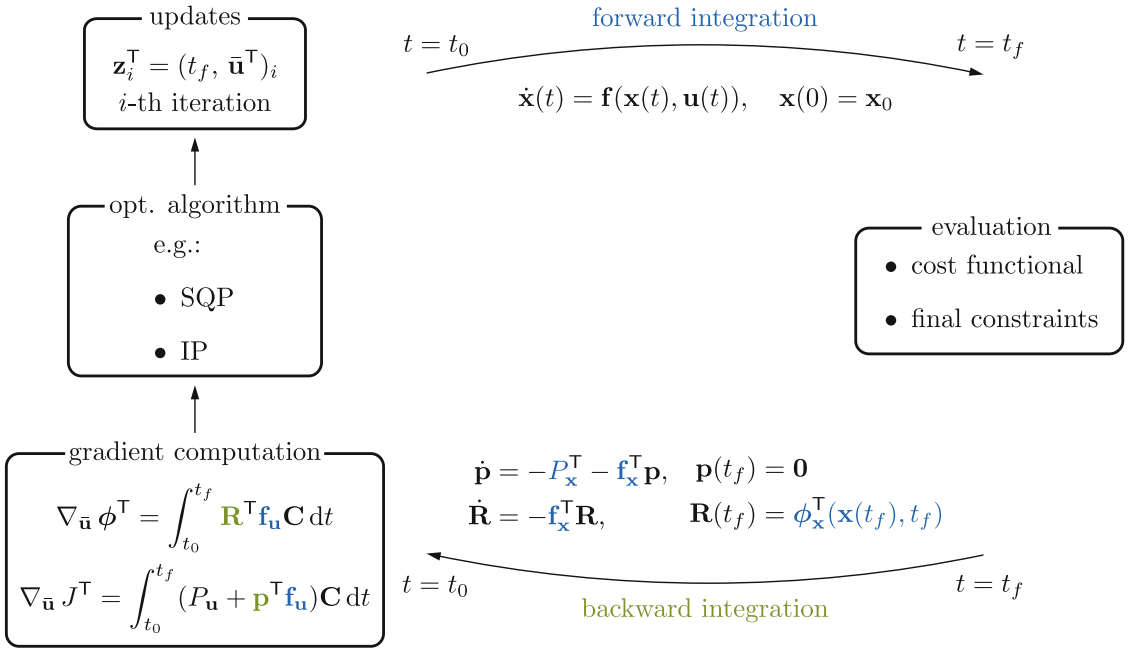



**Fig. 3.** Procedure for the use of adjoint gradients in direct optimization approaches

Both terms require the solution of a linear differential system, but it has to be emphasized that the size of the systems is different. The size of the system sensitivity depends on the number of states $n$, the number of controls $m$ and on the number of grid nodes $k$, while the size of the adjoint system depends only on $n$. To compute the gradients, one can solve either the primal system (a) with dimension $(n \times m \cdot k)$ or the dual system (b) with dimension $(n \times 1)$. Thus, the adjoint approach is an efficient technique to incorporate especially a large number of grid nodes. A graphical interpretation of the dimensions occurring in the gradients of the cost functional is shown in Fig. 4, with a special focus on increasing the number of grid nodes.

## 4 Numerical Example

### 4.1 Task Description and Optimization Problem

The analytically derived adjoint gradients in [6] are used for a direct optimization method in a time-optimal control problem of a SCARA with two rigid bodies. The goal is to manipulate the tool center point (TCP) of the robot depicted in Fig. 5 from an initial state to a final state in minimal operation time $t_f^*$ with a discrete control parameterization. To meet industrial requirements, the control is forced to be $\mathcal{C}^2$ continuous. Hence, the matrix $\mathbf{C}$ is chosen such that the interpolation of each discretized control subinterval is performed by a cubic spline function. The state equations are obtained by introducing the state variables

$$\mathbf{x} = (\varphi_1,\, \varphi_2,\, \omega_1,\, \omega_2)^\mathsf{T}, \tag{16}$$

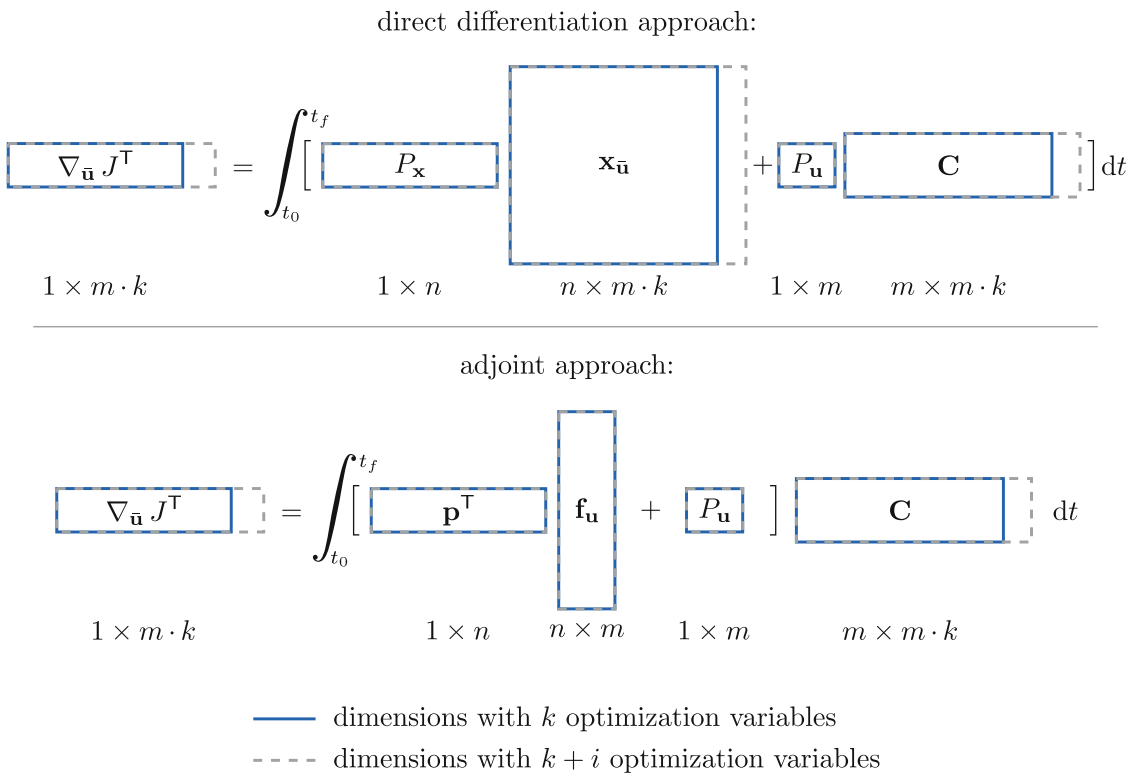



**Fig. 4.** Graphical interpretation of the dimensions occurring in the gradient of the cost functional with respect to the direct differentiation approach in Eq. (6) and the adjoint approach in Eq. (13)

in which $\dot{\varphi}_i = \omega_i$. The model parameters for the simulation are set as follows: $m_1 = m_3 = 1\,\mathrm{kg}$, $m_2 = 0.5\,\mathrm{kg}$, $l_i = 1\,\mathrm{m}$ and $J_i = m_i l_i^2/12$, in which $i \in \{1, 2\}$. The mass $m_3$ is considered as a point mass attached to the TCP.

The cost functional of the optimization problem is given in Eq. (2), in which the penalty term $P(\mathbf{u}) = 10(P_1(u_1) + P_2(u_2))$ is used with

$$P_i(u_i) := \begin{cases} 0 & \text{for } |u_i| < u_{i,\max}, \\ \frac{1}{2}(|u_i| - u_{i,\max})^2 & \text{otherwise.} \end{cases} \tag{17}$$

The final constraints of the system are defined as

$$\boldsymbol{\phi}(\varphi_1, \varphi_2, \omega_1, \omega_2) := \begin{pmatrix} l_1 \cos(\varphi_1) + l_2 \cos(\varphi_2) - x_f \\ l_1 \sin(\varphi_1) + l_2 \sin(\varphi_2) - y_f \\ \omega_1 \\ \omega_2 \end{pmatrix} \Bigg|_{t=t_f}, \tag{18}$$

in which $x_f = 1\,\mathrm{m}$ and $y_f = 1\,\mathrm{m}$ denote the desired final configuration of the TCP. Physical bounds of the controls are given by $u_{1,\max} = 4\,\mathrm{Nm}$ and $u_{2,\max} = 2\,\mathrm{Nm}$.

The NLP contains the optimization variables $\mathbf{z}^\mathsf{T} = (t_f, \bar{\mathbf{u}}^\mathsf{T})$ and is solved with an SQP method. As an initial guess, the assumption for the final time is $t_f = 2\,\mathrm{s}$ and the grid nodes are set to $\bar{\mathbf{u}} = \mathbf{0}$. Initial conditions of the state

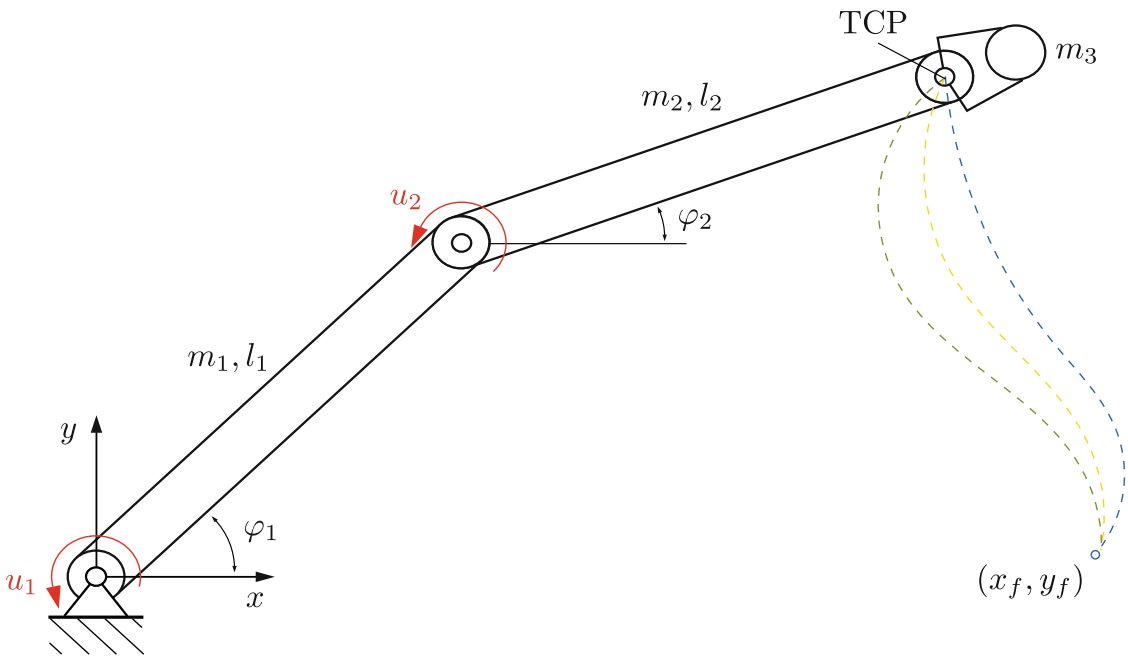



**Fig. 5.** SCARA with two rigid bodies in a general configuration

variables are set to $\mathbf{x}_0 = (-\pi/4,\, 0,\, 0,\, 0)^\mathsf{T}$. In order to analyze the sensitivity of the solution to the refinement of the discretization of the control, both controls are equidistantly discretized in the time interval $t \in [0, t_f]$ with a set of grid nodes with various number $k \in \{5, 10, 20, 30, 40, 50\}$.

### 4.2 Results

Figure 6 shows the optimal control history $\mathbf{u}_k^*$ and the resulting trajectory of the TCP with respect to the defined number of grid nodes $k$. One can observe that the control becomes a bang-bang type control by increasing the number of grid nodes. It can also be seen that the TCP trajectory with $k = 5$ grid nodes is noticeably different compared to controls in which the number of grid nodes is higher. This is due to the fact that in this case the optimal control cannot be represent a bang-bang structure. Theoretically, an infinite number of grid nodes will lead to the shortest possible final time. The final times for the six independent optimizations are $(k = 5, t_f^* = 1.9439\,\mathrm{s})$, $(10, 1.8633\,\mathrm{s})$, $(20, 1.8391\,\mathrm{s})$, $(30, 1.8325\,\mathrm{s})$, $(40, 1.8303\,\mathrm{s})$ and $(50, 1.8294\,\mathrm{s})$.

The optimal control with $k = 50$ grid nodes and the corresponding switching functions, as defined in [2] for bang-bang controls, are shown in Fig. 7. The zero values of the control agree well with those of the switching functions $h_i$ and the Hamiltonian of the system is sufficiently small. Thus, the termination criteria shown in Fig. 1 is satisfied and a bang-bang control can be approximated with a sufficient number of grid nodes.

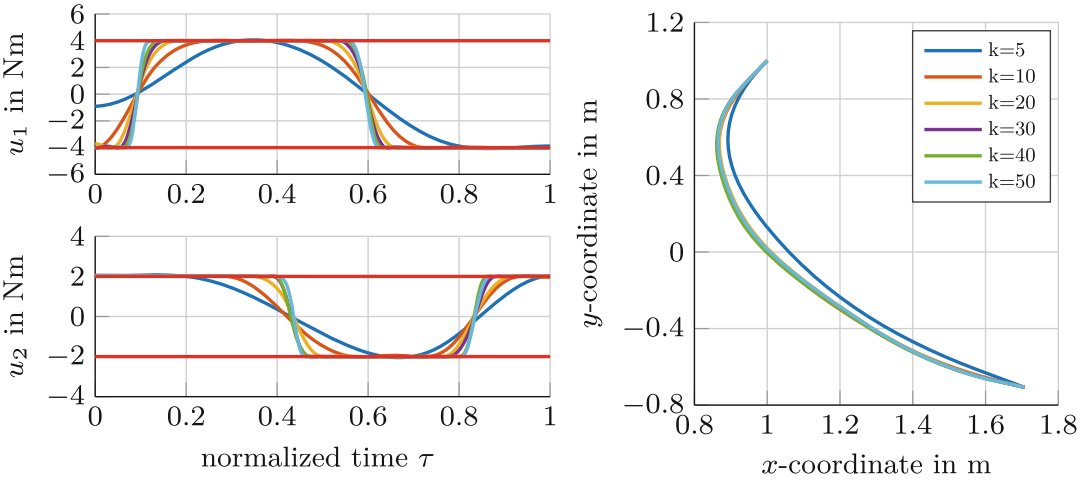



**Fig. 6.** Optimal control history and TCP trajectory for various number of grid nodes

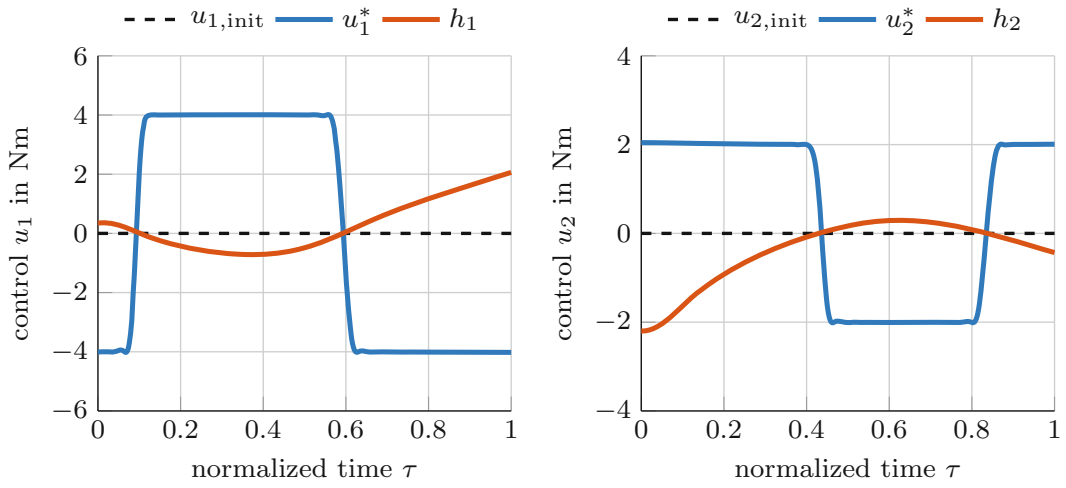



**Fig. 7.** Initial controls, optimal controls and switching functions considering a cubic spline parameterization of the control

## 5 Conclusions

This paper presents a procedure for using adjoint variables in a direct optimization approach. The adjoint variables are examined in the context of two scenarios: The adjoint variables are used to compute the gradients during the optimization. In addition, the adjoint variables are used to evaluate Pontryagin's minimum principle in order to discuss the optimization results obtained by an SQP method. A time-optimal control problem of a SCARA shows the versatile application of adjoint variables. Moreover, the computational effort for the computation of gradients can be reduced by considering adjoint gradients, especially when the number of grid nodes is large or the mechanical system is difficult to solve forward in time.

**Acknowledgements.** Daniel Lichtenecker and Karin Nachbagauer acknowledge support from the Technical University of Munich - Institute for Advanced Study.

## References


1. Bryson, A.E., Ho, Y.C.: Applied Optimal Control: Optimization, Estimation, and Control. Taylor & Francis, New York (1975). https://doi.org/10.1201/9781315137667
2. Eichmeir, P., Nachbagauer, K., Lauß, T., Sherif, K., Steiner, W.: Time-optimal control of dynamic systems regarding final constraints. J. Comput. Nonlinear Dynam. **16**(3), 031003 (2021). https://doi.org/10.1115/1.4049334
3. Karush, W.: Minima of functions of several variables with inequalities as side constraints. Master's thesis, Department of Mathematics, University of Chicago (1939)
4. Kelley, H.J.: Method of gradients: optimization techniques with applications to aerospace systems. Math. Sci. Eng. **5**, 205–254 (1962)
5. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492 (1951)
6. Lichtenecker, D., Rixen, D., Eichmeir, P., Nachbagauer, K.: On the use of adjoint gradients for time-optimal control problems regarding a discrete control parameterization. Multibody Sys. Dyn. **59**(3), 313–334 (2023). https://doi.org/10.1007/s11044-023-09898-5
7. McNamara, A., Treuille, A., Popović, Z., Stam, J.: Fluid control using the adjoint method. ACM Trans. Graph. **23**(3), 449–456 (2004). https://doi.org/10.1145/1015706.1015744
8. Nachbagauer, K., Oberpeilsteiner, S., Sherif, K., Steiner, W.: The use of the adjoint method for solving typical optimization problems in multibody dynamics. J. Comput. Nonlinear Dynam. **10**(6), 061011 (2015). https://doi.org/10.1115/1.4028417
9. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5
10. Pikuliński, M., Malczyk, P.: Adjoint method for optimal control of multibody systems in the Hamiltonian setting. Mech. Mach. Theory **166**, 104473 (2021). https://doi.org/10.1016/j.mechmachtheory.2021.104473
11. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mischchenko, E.F.: The Mathematical Theory of Optimal Processes. John Wiley & Sons, New York (1962)
12. Reiter, A., Müller, A., Gattringer, H.: On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators. IEEE Trans. Ind. Inf. **14**(4), 1681–1690 (2018). https://doi.org/10.1109/TII.2018.2792002

## B.3 Publication III

### Bibliographic Information

Lichtenecker, D. and Nachbagauer, K. “A discrete adjoint gradient approach for equality and inequality constraints in dynamics”. In: *Multibody System Dynamics* 61.1 (2024), pp. 103–130. DOI: 10.1007/s11044-024-09965-5

### Copyright Information

Multibody System Dynamics (2024) 61:103–130
https://doi.org/10.1007/s11044-024-09965-5

RESEARCH

# A discrete adjoint gradient approach for equality and inequality constraints in dynamics

**Daniel Lichtenecker**[1] · **Karin Nachbagauer**[2,3]

Received: 2 October 2023 / Accepted: 15 January 2024 / Published online: 29 January 2024


## Abstract

The optimization of multibody systems requires accurate and efficient methods for sensitivity analysis. The adjoint method is probably the most efficient way to analyze sensitivities, especially for optimization problems with numerous optimization variables. This paper discusses sensitivity analysis for dynamic systems in gradient-based optimization problems. A discrete adjoint gradient approach is presented to compute sensitivities of equality and inequality constraints in dynamic simulations. The constraints are combined with the dynamic system equations, and the sensitivities are computed straightforwardly by solving discrete adjoint algebraic equations. The computation of these discrete adjoint gradients can be easily adapted to deal with different time integrators. This paper demonstrates discrete adjoint gradients for two different time-integration schemes and highlights efficiency and easy applicability. The proposed approach is particularly suitable for problems involving large-scale models or high-dimensional optimization spaces, where the computational effort of computing gradients by finite differences can be enormous. Three examples are investigated to validate the proposed discrete adjoint gradient approach. The sensitivity analysis of an academic example discusses the role of discrete adjoint variables. The energy optimal control problem of a nonlinear spring pendulum is analyzed to discuss the efficiency of the proposed approach. In addition, a flexible multibody system is investigated in a combined optimal control and design optimization problem. The combined optimization provides the best possible mechanical structure regarding an optimal control problem within one optimization.

**Keywords** Discrete adjoint method · Sensitivity analysis · Optimal control · Optimal design · Flexible multibody systems

✉ D. Lichtenecker
daniel.lichtenecker@tum.de

K. Nachbagauer
karin.nachbagauer@fh-wels.at

1 TUM School of Engineering and Design, Department of Mechanical Engineering, Chair of Applied Mechanics, Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich, Munich, Germany

2 Institute for Advanced Study, Technical University of Munich, Lichtenbergstraße 2a, 85748 Garching, Germany

3 Faculty of Engineering and Environmental Sciences, University of Applied Sciences Upper Austria, Stelzhamerstraße 23, 4600 Wels, Austria

## 1 Introduction

The direct differentiation and the adjoint variable approach represent two principal analytical methods employed for sensitivity analysis in the context of optimization. The direct differentiation method [13] facilitates implementation via straightforward differentiation of system equations, constraints, and cost functions with respect to optimization variables. Despite this, the direct differentiation yields a tremendous computational effort for large-scale optimization problems. Alternatively, the adjoint variable method [13] determines design sensitivities as the solution of adjoint variable equations deduced from variations of the system equations. This avoids the necessity for direct computation of state sensitivities and can dramatically reduce the computational effort required for large-scale optimization problems when adjoint variable equations and algorithms are properly formulated.

The computation of gradients in optimization problems includes the adjoint method with a long history in optimal control theory [24]. Adjoint gradients, e.g., applied for trajectory planning, have already been presented in 1975 by Bryson and Ho [4]. The adjoint gradients have become more and more important in various optimization problems in engineering, e.g., in design optimization [32] or multibody dynamics [7, 10], since the applications have become larger in dimension. Therefore, an efficient calculation of gradients has become essential. Sensitivity analysis has wide-ranging applications in science and engineering, including optimization, parameter identification, data assimilation, optimal control, uncertainty analysis, and experimental design. Current trends in neural networks benefit from solvers capable of building efficient gradient computation for training machine-learning embedded cost functionals in high dimensions [36]. Johnston and Patel stated in [17] that adjoint methods are used both in control theory and machine learning to efficiently compute gradients of functionals. Recent publications discuss the adjoint method in multibody dynamics for various applications, e.g., in a feedback–feedforward control to compute the input control signal and corresponding trajectory predicted by a model [27]. In the work by Schneider and Betsch [38], the choice of a mechanical Hamiltonian and the incorporation of constraints is discussed, and a new approach that preserves the variational structure of the problem is introduced. Moreover, an adjoint sensitivity analysis using a QR decomposition in [16] shows how the adjoint variable method can be applied to multibody systems whose system equations are initially set up in differential–algebraic form but solved in minimal coordinates.

For optimization in complex, large-scale optimization problems, a discrete version of the adjoint method with neat features in terms of stability and accuracy has been proposed recently by various authors [3, 6, 21]. In the discrete adjoint method, the adjoint differential equations are replaced by algebraic equations by introducing a finite-difference scheme for the adjoint system directly from the numerical time-integration method. The method provides exact gradients of the discretized cost function subjected to the discretized equations of motion. The equations of motion of the multibody system and adjoint equations may either be separately discretized from their representations as differential–algebraic equations, or the equations of motion of the multibody system may be discretized first, and the discrete adjoint equations are then derived directly from the discrete multibody equations, tracing back to [4]. It has been emphasized, e.g., by Callejo et al. [6], that the adjoint method is one of the most efficient methods to evaluate sensitivities for problems involving numerous design parameters and relatively few objective functions. The latter paper has presented a discrete version of the adjoint method, which can be applied to the dynamic simulation of flexible multibody systems not only by using an ad hoc backward integration solver but leads to a straightforward algebraic procedure that provides the desired design sensitivities of rigid and flexible multibody systems. Moreover, in [3], the discrete adjoint method

is discussed in different time-marching schemes, including backward difference formulas, Newmark and Adams–Bashforth–Moulton methods. In [25], a discrete adjoint sensitivity analysis considering a Newmark family integrator is presented. In Lauß et al. [21], the discrete adjoint equations for the computation of gradients of a cost function are derived using the Hilbert–Hughes–Taylor (HHT) solver to solve the system equations. The great advantage of this approach is that the cost function can also depend on the accelerations, thus allowing the use of measured data from acceleration sensors in the optimization procedure in a straightforward manner.

Flexible multibody formulations must be included in optimization problems when the cost function includes elasticity or deformation of mechanical systems. When dealing with flexible multibody systems with large deformations or large rotations, the absolute nodal coordinate formulation (ANCF) [40] is advantageous because the formulation does not use rotational degrees of freedom. Using these ANCF elements for flexible bodies, a gradient-based adjoint optimization approach has been presented by Held and Seifried [15]. There, a criterion accounts for the deformation energy of the flexible body. A recent work [44] presents an optimization approach that exploits the adjoint variable method in combination with the flexible natural coordinates formulation for obtaining the sensitivity information. A comprehensive literature review in [12] presents various gradient-based optimization methods, especially in the design optimization of flexible multibody systems. The latter-mentioned review paper discusses the main goals in the design optimization of flexible multibody dynamics and reviews concepts and applications in this field. Over 160 publications in the bibliography give a comprehensive overview of optimization algorithms, types and formulations, and sensitivity analysis. Optimal control and design optimization are discussed in various publications, but there is a gap in the literature on optimization or sensitivity analysis that combines optimal control and design optimization of multibody systems.

This paper significantly enhances optimal control and design optimization problems for flexible multibody systems. Promising results in a preliminary paper by Lichtenecker et al. [22] have shown an efficient optimal control strategy for highly flexible robotic systems based on the adjoint gradient computation method. Furthermore, flexible multibody systems, e.g., soft robots, allow new potential for performing various tasks. Soft robots have not yet fully demonstrated their capabilities, as nature is still clearly superior to them in some areas, particularly evolutionary improved motion and control. Future research should address critical challenges and focus on understanding the fundamental principles that govern the design, modeling, and control of soft robots, as stated in Hawkes et al. [14] and Della Santina et al. [9]. Improving the performance of mechanical systems, like soft robots, requires sophisticated optimization strategies to fulfill the high demands of current and future product requirements. In general, two problem formulations can be considered to describe various optimization applications: structural optimization of mechanical components and/or finding an optimal control for dynamical systems [43]. The focus of this paper is on the combination of both problems. A combined gradient computation for highly efficient optimal control while optimizing the structural components of a mechanical system within the same computation is presented.

To this end, a discrete adjoint gradient computation considering equality and inequality constraints is developed, which will be able to incorporate, e.g., final conditions and/or stress restrictions in design optimizations. With this novel approach, a sensitivity analysis of constraints with respect to optimization variables using discrete adjoints is possible, and a combined optimal control and optimal design of a mechanical system is realized. Three examples will show the application of the proposed discrete adjoint gradient approach: (1)

an academic example of a one-mass oscillator in a sensitivity analysis, (2) an energy optimal control problem of a nonlinear spring pendulum, and (3) a combined optimal control and optimal sizing problem of a flexible two-arm robot using the ANCF for describing large deformations.

## 2 Problem description

The increasing industrial relevance of high-end solutions that fulfill a wide range of requirements demands the consideration of novel approaches at an early stage of virtual product development. For instance, finding an optimal control of a flexible multibody system under consideration of final constraints is essential to perform a manipulation according to predefined tasks [22]. In addition, innovative lightweight design requires novel approaches in the field of structural optimization [28]. The performance of mechanical systems, either in optimal control problems and/or structural optimization problems, can be increased by optimization.

An optimization problem aims to find a set of optimization variables $\mathbf{z} = \mathbf{z}^* \in \mathbb{R}^z$ to minimize a defined cost function $J$ with respect to constraints. A standard nonlinear programming (NLP) problem can be formulated by

$$\min_{\mathbf{z}} J(\mathbf{z}) \tag{1}$$

s.t.

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} \tag{2}$$

$$\hat{\mathbf{g}} = \mathbf{0} \tag{3}$$

$$\hat{\mathbf{h}} \leq \mathbf{0}, \tag{4}$$

wherein (2) imposes lower and upper bounds of optimization variables. Equality and inequality constraints are denoted by $\hat{\mathbf{g}} \in \mathbb{R}^p$ and $\hat{\mathbf{h}} \in \mathbb{R}^q$, respectively. To address an optimal control problem with the above NLP formulation, it is necessary to transform the original infinite-dimensional optimization problem into a finite-dimensional one by using a direct transcription method. In general, transcription methods are categorized into shooting methods and collocation methods. A widely used method is the direct single-shooting method, where only the control is parameterized. The reader is referred to [2] for a detailed description of transcription methods. The NLP problem can be treated by well-known algorithms, e.g., the sequential quadratic programming (SQP) approach or the interior point (IP) method [33]. An optimal point $\mathbf{z}^*$ fulfills the Karush–Kuhn–Tucker (KKT) conditions [18, 19], which are necessary first-order optimality conditions for direct optimization methods.

In this paper, the constraints $\hat{\mathbf{g}} = \hat{\mathbf{g}}(\mathbf{x}, \boldsymbol{\xi})$ and $\hat{\mathbf{h}} = \hat{\mathbf{h}}(\mathbf{x}, \boldsymbol{\xi})$ depend on system parameters $\boldsymbol{\xi} \in \mathbb{R}^l$ and on state variables $\mathbf{x} \in \mathbb{R}^n$ due to a time-dependent control $\mathbf{u} \in \mathbb{R}^m$ of a mechanical system. Thus, evaluating the (in)equality constraints requires the evolution of the system response. The dynamics of mechanical systems can be described with ordinary differential equations (ODE)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}), \quad \mathbf{x}(0) = \mathbf{x}_0. \tag{5}$$

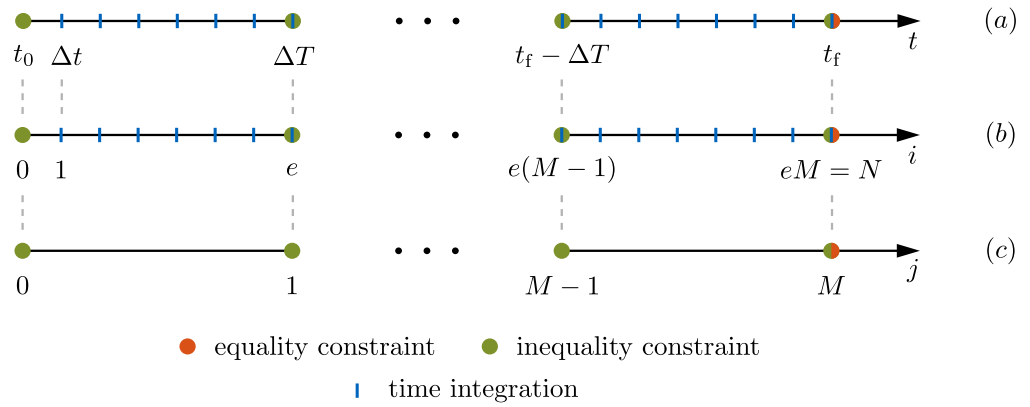



**Fig. 1** Time domain and index scale: $(a)$ is the time domain for time integration, $(b)$ is the $i$-index scale for time integration, and $(c)$ is the $j$-index scale for the evaluation of constraints

For the numerical computation of the state variables, the ODE can be formulated by a temporal discretization as

$$\mathbf{x}_{i+1} = \tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{u}_i, \mathbf{u}_{i+1}, \boldsymbol{\xi}), \quad i \in \{0, \ldots, N-1\}, \tag{6}$$

with the given initial state $\mathbf{x}_0$. The evolution of the state variables $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is influenced by the time integrator used, the control variables $\mathbf{u}_0, \ldots, \mathbf{u}_N$, and the set of parameters $\boldsymbol{\xi}$. The discrete ODE (6) represents a general form for an explicit or implicit one-step time integrator, such as a Runge–Kutta scheme [5]. The time domain $t \in [t_0, t_f]$ is discretized with $N$ uniform intervals leading to a constant time-integration step size $\Delta t$; see Fig. 1(a). Consequently, the micro (integration) time mesh is defined by $t_i = \Delta t\, i$, $i \in \{0, \ldots, N\}$, with $t_0 = 0$ and $t_N = t_f = \Delta t\, N$.

This paper considers equality constraints at the final time $t_f$, while inequality constraints must be satisfied on a macro (inequality) time mesh; see Fig. 1(c) for the according index scale of the macro time mesh. The macro time mesh $\hat{t}_j = \Delta T\, j$, $j \in \{0, \ldots, M\}$ is defined by $M$ time intervals between inequality constraints leading to the inequality step size $\Delta T = t_f/M$. The circumflex $\hat{(\cdot)}$ denotes the evaluation of a variable $(\cdot)$ regarding the macro time mesh. The number of chosen time-integration points $e \in \mathbb{N}$ divides the inequality step size $\Delta T$ into the integration step size $\Delta t$ by defining the time-integration step size $\Delta t = \Delta T/e$. For the case $M = N$, one defines inequality constraints at the micro time mesh in each time integration. Figure 1 illustrates the time domain, including the time-integration steps and the arrangement of (in)equality constraints. For example, an equality constraint can be a particular configuration of a mechanical system at the final time $t_f$. In contrast, an inequality constraint ensures that the acceleration is within a defined limit at the macro time mesh $\hat{t}$.

For the sake of convenience, the constraints $\hat{\mathbf{g}}$ and $\hat{\mathbf{h}}$ are concatenated into a general set of nonlinear constraints by $\mathbf{c}^\mathsf{T} = \left(\hat{\mathbf{g}}^\mathsf{T}, \hat{\mathbf{h}}^\mathsf{T}\right) \in \mathbb{R}^{p+q}$. The equality constraints $\hat{\mathbf{g}} = \mathbf{g}_N$ represent an evaluation of $p$ implicit time-dependent functions $\mathbf{g} = \mathbf{g}(\mathbf{x}(t), \boldsymbol{\xi}) \in \mathbb{R}^p$ at the final time $t_f$, while the inequality constraints $\hat{\mathbf{h}}$ are a concatenation of $r$ functions $\mathbf{h} = \mathbf{h}(\mathbf{x}(t), \boldsymbol{\xi}) \in \mathbb{R}^r$ evaluated at the macro time mesh. The size of the concatenated inequality constraints is $q = r(M+1)$; see the $j$-index scale for inequality constraints in Fig. 1. However, inequality constraints have to be defined in the $i$-index scale to be in accordance with a time integration of the ODE at the micro time mesh, i.e., inequality constraints are defined in the $i$-index scale

with $i = ej$. The general set of constraints is formulated in the $i$-index scale by

$$\mathbf{c} = \mathbf{B}_1 \mathbf{g}_N + \sum_{j=0}^{M} \mathbf{B}_{2,j} \mathbf{h}_{ej}. \tag{7}$$

Boolean matrices $\mathbf{B}$ map the (in)equality constraints into the combined set of constraints, i.e.,

$$\mathbf{c}^\mathsf{T} = (\underbrace{g_{1,N}, \ldots, g_{p,N}}_{\hat{\mathbf{g}}^\mathsf{T}}, \underbrace{h_{1,0}, \ldots, h_{r,0}, \ldots, h_{1,N}, \ldots, h_{r,N}}_{\hat{\mathbf{h}}^\mathsf{T}}), \tag{8}$$

in which the first subscript denotes the row and the second subscript denotes the corresponding time in the $i$-index scale of $\mathbf{g}$ and $\mathbf{h}$, respectively.

The constraint formulation in (7) allows a straightforward derivation of discrete adjoint gradients for sensitivity analysis and gradient-based optimizations. In gradient-based optimization algorithms, first-order gradients are crucial to compute a local minimum of an optimization problem. The accuracy of the gradient computation influences the convergence and robustness of optimization algorithms. In addition, the computational effort to solve an optimization problem depends on the runtime required to compute gradients. This paper addresses accurate and efficient first-order sensitivity analysis with particular emphasis on gradients of the constraint formulation in (7). One approach that meets both requirements is the adjoint method [26, 29]. This paper employs a discrete version of the adjoint method to derive first-order gradients of constraints. The proposed discrete adjoint approach replaces the finite-difference approach, usually the default of optimization toolboxes.

## 3 Sensitivity analysis

This section proposes a novel discrete adjoint approach to sensitivity analysis with emphasis on gradients of the constraint formulation in (7). The sensitivities are computed for a discrete set of optimization variables $\mathbf{z}$, i.e., the sensitivity analysis is defined by $\mathrm{d}\mathbf{c}/\mathrm{d}\mathbf{z}$. The adjoint method is an efficient method to compute gradients since the computational effort to compute the so-called adjoint variables does not depend on the number of optimization variables [6, 11, 31]. The discrete adjoint method constructs a finite-difference scheme for the adjoint variables directly from the time-integration method to solve the governing equations [21]. In this paper, the computation of the discrete adjoint gradients is formulated for governing equations in the form of (6). However, the discrete adjoint gradient computation can be easily adapted to deal with different one-step time integrators, as shown for an explicit and an implicit Euler method.

### 3.1 Discrete adjoint method for one-step time integrators

The goal of the adjoint gradient method is to avoid the expensive computation of state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$ by introducing adjoint variables. Following the fundamental work by Bryson and Ho [4], the constraint (7) is extended by the discrete state equations (6) leading to

$$\bar{\mathbf{c}} = \mathbf{B}_1 \mathbf{g}_N + \sum_{j=0}^{M} \mathbf{B}_{2,j} \mathbf{h}_{ej} + \sum_{j=0}^{M-1} \left[ \sum_{i=ej}^{e(j+1)-1} \mathbf{R}_{i+1} \left( \mathbf{x}_{i+1} - \tilde{\mathbf{f}}_{i,i+1} \right) \right], \tag{9}$$

for any choice of the adjoint variables $\mathbf{R}_1, \ldots, \mathbf{R}_N \in \mathbb{R}^{(p+q)\times n}$ in the case where the discrete state equations are satisfied. For the sake of easier reading, the temporal discretized right-hand side of the ODE is defined by $\tilde{\mathbf{f}}_{i,i+1} := \tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{u}_i, \mathbf{u}_{i+1}, \boldsymbol{\xi})$, $i \in \{0, \ldots, N-1\}$. The constraint (9) is extended by introducing a double sum to account for the system dynamics. The inner sum considers the time intervals between two inequality constraints $]\hat{t}_j, \hat{t}_{j+1}[$, $j \in \{0, \ldots, M-1\}$ and the outer sum considers exactly the time instances $\hat{t}_j$, $j \in \{0, \ldots, M\}$ at which inequality constraints must be satisfied; see scales in Fig. 1. However, the additional zero terms do not influence the constraints $\mathbf{c}$ in the case where the state equations are satisfied and, therefore, the gradients of $\mathbf{c}$ are equal to the gradients of $\bar{\mathbf{c}}$. The following derivation of discrete adjoint gradients is based on the extended constraints $\bar{\mathbf{c}}$.

Before deriving the gradients of $\bar{\mathbf{c}}$ by a discrete adjoint approach, the boundaries of the state variables $\mathbf{x}_N$ and (in)equality constraints $\mathbf{g}_N$ and $\mathbf{h}_N$, respectively, must be extracted from (9) to derive terminal conditions for the differential equations of the adjoint variables. Performing an index shift of the inner $i$-sum in (9) results in an extraction of the boundaries

$$\begin{aligned}\bar{\mathbf{c}} = \mathbf{B}_1 \mathbf{g}_N + \mathbf{B}_{2,M} \mathbf{h}_N + \sum_{j=0}^{M-1} \Bigg[ \sum_{i=ej+1}^{e(j+1)-1} \mathbf{R}_i \left( \mathbf{x}_i - \tilde{\mathbf{f}}_{i-1,i} \right) \\ + \mathbf{R}_{ej} \mathbf{x}_{ej} - \mathbf{R}_{e(j+1)} \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)} + \mathbf{B}_{2,j} \mathbf{h}_{ej} \Bigg] - \mathbf{R}_0 \mathbf{x}_0 + \mathbf{R}_N \mathbf{x}_N .\end{aligned} \tag{10}$$

Note that (9) and (10) are equal despite different formulations. Moreover, note that the subscripts $ej$ and $e(j+1)$ in (10) are a result of the performed index shift, and the related terms are evaluated in the $j$-sum at the macro time mesh; see Fig. 1.

The derivation of discrete adjoint gradients is based on the calculus of variations. The first-order variation of (10) in terms of $\delta\mathbf{x}_i$, $\delta\mathbf{x}_{i-1}$, $\delta\mathbf{u}_i$, $\delta\mathbf{u}_{i-1}$, and $\delta\boldsymbol{\xi}$ is given by

$$\begin{aligned}\delta\bar{\mathbf{c}} =& \mathbf{B}_1 \left( \frac{\partial \mathbf{g}_N}{\partial \mathbf{x}_N} \delta\mathbf{x}_N + \frac{\partial \mathbf{g}_N}{\partial \boldsymbol{\xi}} \delta\boldsymbol{\xi} \right) + \mathbf{B}_{2,M} \left( \frac{\partial \mathbf{h}_N}{\partial \mathbf{x}_N} \delta\mathbf{x}_N + \frac{\partial \mathbf{h}_N}{\partial \boldsymbol{\xi}} \delta\boldsymbol{\xi} \right) \\ &+ \sum_{j=0}^{M-1} \Bigg[ \sum_{i=ej+1}^{e(j+1)-1} \mathbf{R}_i \Bigg( \delta\mathbf{x}_i - \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{x}_{i-1}} \delta\mathbf{x}_{i-1} - \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_{i-1}} \delta\mathbf{u}_{i-1} - \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{x}_i} \delta\mathbf{x}_i \\ &- \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_i} \delta\mathbf{u}_i - \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \boldsymbol{\xi}} \delta\boldsymbol{\xi} \Bigg) + \mathbf{R}_{ej} \delta\mathbf{x}_{ej} + \mathbf{B}_{2,j} \left( \frac{\partial \mathbf{h}_{ej}}{\partial \mathbf{x}_{ej}} \delta\mathbf{x}_{ej} + \frac{\partial \mathbf{h}_{ej}}{\partial \boldsymbol{\xi}} \delta\boldsymbol{\xi} \right) \\ &- \mathbf{R}_{e(j+1)} \Bigg( \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{x}_{e(j+1)-1}} \delta\mathbf{x}_{e(j+1)-1} + \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{u}_{e(j+1)-1}} \delta\mathbf{u}_{e(j+1)-1} \\ &+ \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{x}_{e(j+1)}} \delta\mathbf{x}_{e(j+1)} + \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{u}_{e(j+1)}} \delta\mathbf{u}_{e(j+1)} \\ &+ \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \boldsymbol{\xi}} \delta\boldsymbol{\xi} \Bigg) \Bigg] - \mathbf{R}_0 \delta\mathbf{x}_0 + \mathbf{R}_N \delta\mathbf{x}_N .\end{aligned} \tag{11}$$

In this paper, the focus lies on a combined sensitivity analysis with respect to the system parameters $\boldsymbol{\xi}$ and a discrete set of control grid nodes $\bar{\mathbf{u}}$. Following Lichtenecker et al. [22],

the continuous control function is formulated with $\mathbf{u} = \mathbf{C}\bar{\mathbf{u}}$, where $\mathbf{C}$ is a time-dependent interpolation function. The variables of interest in the sensitivity analysis are combined in the vector $\mathbf{z}^\mathsf{T} = \left(\boldsymbol{\xi}^\mathsf{T}, \bar{\mathbf{u}}^\mathsf{T}\right)$ and can be used as optimization variables in gradient-based optimization problems. Note that the optimization variables $\mathbf{z}$ can consist of system parameters, e.g., the stiffness of a spring of a mechanical system and/or a parameterization of the control.

To derive a variation of the constraints with respect to the optimization variables $\mathbf{z}$, the variations $\delta\boldsymbol{\xi}$ and $\delta\mathbf{u}$ in (11) can be obtained in terms of $\delta\mathbf{z}$ with

$$\delta\boldsymbol{\xi} = \mathbf{B}_3\delta\mathbf{z}, \tag{12}$$

$$\delta\mathbf{u} = \mathbf{C}\delta\bar{\mathbf{u}} = \mathbf{C}\mathbf{B}_4\delta\mathbf{z}, \tag{13}$$

respectively. The Boolean matrices $\mathbf{B}$ map the combined set of optimization variables to system parameters and control grid nodes. Substituting (12) and (13) into (11) and reformulating leads to

$$\begin{aligned}
\delta\bar{\mathbf{c}} = \sum_{j=0}^{M-1} \Bigg[ \sum_{i=ej+1}^{e(j+1)-1} \Bigg[ & \left( \mathbf{R}_i - \mathbf{R}_i \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{x}_i} - \mathbf{R}_{i+1} \frac{\partial \tilde{\mathbf{f}}_{i,i+1}}{\partial \mathbf{x}_i} \right) \delta\mathbf{x}_i \\
& - \mathbf{R}_i \left( \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_{i-1}} \mathbf{C}_{i-1}\mathbf{B}_4 + \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_i} \mathbf{C}_i\mathbf{B}_4 + \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \boldsymbol{\xi}} \mathbf{B}_3 \right) \delta\mathbf{z} \Bigg] \\
& + \left( \mathbf{R}_{ej} + \mathbf{B}_{2,j} \frac{\partial \mathbf{h}_{ej}}{\partial \mathbf{x}_{ej}} - \mathbf{R}_{ej+1} \frac{\partial \tilde{\mathbf{f}}_{ej,ej+1}}{\partial \mathbf{x}_{ej}} \right) \delta\mathbf{x}_{ej} \\
& - \mathbf{R}_{e(j+1)} \left( \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{u}_{e(j+1)-1}} \mathbf{C}_{e(j+1)-1}\mathbf{B}_4 + \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{u}_{e(j+1)}} \mathbf{C}_{e(j+1)}\mathbf{B}_4 \right. \\
& \left. + \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \boldsymbol{\xi}} \mathbf{B}_3 \right) \delta\mathbf{z} + \mathbf{B}_{2,j} \frac{\partial \mathbf{h}_{ej}}{\partial \boldsymbol{\xi}} \mathbf{B}_3\delta\mathbf{z} \Bigg] \\
& + \left( \mathbf{B}_1 \frac{\partial \mathbf{g}_N}{\partial \boldsymbol{\xi}} + \mathbf{B}_{2,M} \frac{\partial \mathbf{h}_N}{\partial \boldsymbol{\xi}} \right) \mathbf{B}_3\delta\mathbf{z} + \left( \mathbf{B}_1 \frac{\partial \mathbf{g}_N}{\partial \mathbf{x}_N} + \mathbf{B}_{2,M} \frac{\partial \mathbf{h}_N}{\partial \mathbf{x}_N} + \mathbf{R}_N \right) \delta\mathbf{x}_N .
\end{aligned} \tag{14}$$

Equation (14) implies the relation between $\delta\mathbf{x}$ and $\delta\mathbf{z}$. The optimization variables $\mathbf{z}$ influence the state variables $\mathbf{x}$ and, therefore, the variation of state variables should be interpreted as [6, 23]

$$\delta\mathbf{x} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}}\delta\mathbf{z}. \tag{15}$$

The total derivatives of state variables with respect to optimization variables are obtained by solving the matrix differential equations

$$\frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}\mathbf{z}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{z}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{z}}. \tag{16}$$

This system is defined by taking the total derivative of (5) with respect to the optimization variables and, therefore, the dimension of the system depends on the number of optimization

variables. The solution of the matrix differential equations is obtained by applying a temporal discretization, where the computational effort to solve (16) can become expensive in the case of a large number of optimization variables. Using the state sensitivities (16) with (15) in (14), first-order gradients of the constraint formulation in (7) can be obtained without the need for the adjoint variables. This approach is called direct differentiation. However, the goal of the proposed discrete adjoint gradient approach is to avoid the direct computation of the state sensitivities $\mathrm{d}\mathbf{x}/\mathrm{d}\mathbf{z}$. To this end, the discrete adjoint variables in (14) are defined such that the brackets multiplied with $\delta\mathbf{x}$ are zero. Hence, the discrete adjoint variables are obtained by the matrix differential equations

$$\mathbf{R}_N = -\mathbf{B}_1 \frac{\partial \mathbf{g}_N}{\partial \mathbf{x}_N} - \mathbf{B}_{2,M} \frac{\partial \mathbf{h}_N}{\partial \mathbf{x}_N}, \tag{17}$$

$$\mathbf{R}_i = \mathbf{R}_{i+1} \frac{\partial \tilde{\mathbf{f}}_{i,i+1}}{\partial \mathbf{x}_i} + \mathbf{R}_i \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{x}_i}, \quad \forall i \in \{ej+1, \ldots, e\,(j+1)-1\}, \tag{18}$$

$$\mathbf{R}_{ej} = -\mathbf{B}_{2,j} \frac{\partial \mathbf{h}_{ej}}{\partial \mathbf{x}_{ej}} + \mathbf{R}_{ej+1} \frac{\partial \tilde{\mathbf{f}}_{ej,ej+1}}{\partial \mathbf{x}_{ej}}, \quad \forall j \in \{1, \ldots, M-1\}. \tag{19}$$

Equation (17) imposes the terminal condition of the adjoint variables at the final time $t_N$. The computation of the adjoint variables is performed in a backward manner starting from $\mathbf{R}_N$ and proceeding with the adjoint system (18). It has to be emphasized that (18) is defined within the macro time mesh $]\hat{t}_j, \hat{t}_{j+1}[$, $j \in \{0, \ldots, M-1\}$. The discrete adjoint variables at the macro time mesh $\hat{t}_j$, $j \in \{1, \ldots, M-1\}$ are determined by the intermediate condition in (19). The adjoint system and the intermediate condition are applied alternately in the backward integration to solve the discrete adjoint variables.

Once the discrete adjoint variables are computed by (17)–(19), the terms related to $\delta\mathbf{x}$ vanish in (14) and, therefore, the variation of the extended constraints simplifies to

$$\delta\bar{\mathbf{c}} = \underbrace{\{\ldots\}}_{\frac{\mathrm{d}\bar{\mathbf{c}}}{\mathrm{d}\mathbf{z}}} \delta\mathbf{z}, \tag{20}$$

where the variation $\delta\mathbf{z}$ is factored out. The simplified variation leads to first-order gradients of constraints with respect to optimization variables given by

$$\begin{aligned}
\frac{\mathrm{d}\bar{\mathbf{c}}}{\mathrm{d}\mathbf{z}} = \sum_{j=0}^{M-1} \Bigg[ \sum_{i=ej+1}^{e(j+1)-1} \mathbf{R}_i \left( -\frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_{i-1}} \mathbf{C}_{i-1} \mathbf{B}_4 - \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_i} \mathbf{C}_i \mathbf{B}_4 - \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \boldsymbol{\xi}} \mathbf{B}_3 \right) \\
+ \mathbf{B}_{2,j} \frac{\partial \mathbf{h}_{ej}}{\partial \boldsymbol{\xi}} \mathbf{B}_3 - \mathbf{R}_{e(j+1)} \left( \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{u}_{e(j+1)-1}} \mathbf{C}_{e(j+1)-1} \mathbf{B}_4 \right. \\
\left. + \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \mathbf{u}_{e(j+1)}} \mathbf{C}_{e(j+1)} \mathbf{B}_4 + \frac{\partial \tilde{\mathbf{f}}_{e(j+1)-1,e(j+1)}}{\partial \boldsymbol{\xi}} \mathbf{B}_3 \right) \Bigg] \\
+ \left( \mathbf{B}_1 \frac{\partial \mathbf{g}_N}{\partial \boldsymbol{\xi}} + \mathbf{B}_{2,M} \frac{\partial \mathbf{h}_N}{\partial \boldsymbol{\xi}} \right) \mathbf{B}_3.
\end{aligned} \tag{21}$$

The discrete adjoint gradient computation for the constraint formulation in (7) is obtained by (21). Note that the computation of discrete adjoint gradients is based on the solution of the adjoint system, whose size does not depend on the number of optimization variables.

The gradient computation using direct differentiation requires the solution of (16), which depends on the number of optimization variables. Therefore, the adjoint-based sensitivity analysis is computationally efficient, especially when dealing with optimization problems with a large number of optimization variables.

Employing the discrete adjoint gradient (21), e.g., within an optimization procedure, requires the specific formulation of the right-hand side vector $\tilde{\mathbf{f}}$. The chosen time integrator to solve the forward dynamics implies the backward integration of the discrete adjoint variables. Moreover, sensitivities of the forward time integrator are recognized in the computation of the discrete adjoint variables. To use the proposed discrete gradient approach, the sensitivities of the chosen forward time integrator need to be defined.

### 3.2 Application to the explicit Euler method

The explicit Euler method is an iterative solution scheme to approximate the state variables of the ODE in (5) by

$$\mathbf{x}_{i+1} = \underbrace{\mathbf{x}_i + \Delta t\, \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \boldsymbol{\xi})}_{\tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{u}_i, \boldsymbol{\xi})}. \tag{22}$$

Using the above time-integration scheme in the discrete adjoint sensitivity analysis, the explicit right-hand side vector $\tilde{\mathbf{f}}$ has to be defined in the general form $\tilde{\mathbf{f}}_{i,i+1} := \tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{u}_i, \boldsymbol{\xi})$. The computation of the discrete adjoint variables requires the derivatives of $\tilde{\mathbf{f}}$ with respect to the state variables $\mathbf{x}$ given by

$$\frac{\partial \tilde{\mathbf{f}}_{i,i+1}}{\partial \mathbf{x}_i} = \mathbf{I} + \Delta t\, \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} \quad \text{and} \quad \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{x}_i} = \mathbf{0}, \tag{23}$$

where $\mathbf{f}_i = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \boldsymbol{\xi})$ and $\mathbf{I}$ denotes the identity matrix. In addition, the discrete adjoint gradient computation requires the derivatives of $\tilde{\mathbf{f}}$ with respect to $\mathbf{u}$ and $\boldsymbol{\xi}$ given by

$$\frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_i} = \mathbf{0}, \quad \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_{i-1}} = \Delta t\, \frac{\partial \mathbf{f}_{i-1}}{\partial \mathbf{u}_{i-1}} \quad \text{and} \quad \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \boldsymbol{\xi}} = \Delta t\, \frac{\partial \mathbf{f}_{i-1}}{\partial \boldsymbol{\xi}}, \tag{24}$$

respectively.

### 3.3 Application to the implicit Euler method

Similar to the explicit Euler method, the implicit Euler method is an iterative solution scheme to approximate the state variables of the ODE in (5) by

$$\mathbf{x}_{i+1} = \underbrace{\mathbf{x}_i + \Delta t\, \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{u}_{i+1}, \boldsymbol{\xi})}_{\tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{u}_{i+1}, \boldsymbol{\xi})}. \tag{25}$$

Using the above time-integration scheme in the discrete adjoint sensitivity analysis, the implicit right-hand side vector $\tilde{\mathbf{f}}$ has to be defined in the general form $\tilde{\mathbf{f}}_{i,i+1} := \tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{u}_{i+1}, \boldsymbol{\xi})$. The computation of the discrete adjoint variables requires the derivatives of $\tilde{\mathbf{f}}$ with respect to the state variables $\mathbf{x}$ given by

$$\frac{\partial \tilde{\mathbf{f}}_{i,i+1}}{\partial \mathbf{x}_i} = \mathbf{I} \quad \text{and} \quad \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{x}_i} = \Delta t\, \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}. \tag{26}$$

In addition, the discrete adjoint gradient computation requires the derivatives of $\tilde{\mathbf{f}}$ with respect to $\mathbf{u}$ and $\boldsymbol{\xi}$ given by

$$\frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_i} = \Delta t \, \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_i}, \quad \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \mathbf{u}_{i-1}} = \mathbf{0} \quad \text{and} \quad \frac{\partial \tilde{\mathbf{f}}_{i-1,i}}{\partial \boldsymbol{\xi}} = \Delta t \, \frac{\partial \mathbf{f}_i}{\partial \boldsymbol{\xi}}, \tag{27}$$

respectively.

### 3.4 Procedure for the use of the discrete adjoint gradients

This section summarizes the sensitivity analysis using the proposed discrete adjoint gradient approach within an optimization problem. The focus is to compute first-order gradients of the constraint formulation in (7) with respect to optimization variables, i.e.,

$$\left. \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}\mathbf{z}} \right|_{\mathbf{z}=\mathbf{z}^{(k)}},$$

evaluated at the $(k)$th iteration in a gradient-based optimization. The use of the proposed discrete adjoint gradient approach can be summarized by the following steps:

1. Set up an optimization problem in the form of (1)–(4) and select an NLP software package to solve the optimization problem, e.g., IPOPT [45].
2. Compute the derivatives by symbolic differentiation for the discrete adjoint approach:
   a) Compute the derivatives of (in)equality constraints $\mathbf{g}$ and $\mathbf{h}$ with respect to $\mathbf{x}$ and $\boldsymbol{\xi}$, respectively, by symbolic differentiation.
   b) Select a numerical time-integration solver and compute the derivatives of $\mathbf{f}$ with respect to $\mathbf{x}$, $\mathbf{u}$, and $\boldsymbol{\xi}$ by symbolic differentiation. Additionally, define the derivatives of the solver-specific right-hand side vector $\tilde{\mathbf{f}}$, e.g., as shown in Sect. 3.2 and Sect. 3.3 for the explicit and the implicit Euler method, respectively.
3. Compute the first-order gradients of the constraints using the discrete adjoint approach:
   a) Compute the state variables $\mathbf{x}$ influenced by the set of optimization variables $\mathbf{z}^{(k)}$ with the chosen time-integration scheme.
   b) Compute the discrete adjoint variables by solving the matrix differential equations (17)–(19) backward in time.
   c) Compute the sensitivities of the constraint formulation in (7) using the discrete adjoint gradient approach (21).
4. Provide the cost function, constraints, and the respective first-order gradients via an interface to the chosen NLP software package. The Hessian of the cost function and the constraints are usually computed internally by the software package.
5. Repeat Steps 3 and 4 until the KKT conditions are satisfied and an optimal solution $\mathbf{z}^*$ is found.

As aforementioned in the second step, derivatives with respect to $\mathbf{x}$, $\mathbf{u}$, and $\boldsymbol{\xi}$ are computed by symbolic differentiation. Considering a complicated mechanical system with many state variables, the governing equations become extensive and difficult to solve. For such systems, the effort to derive system derivatives by symbolic differentiation is enormous or not feasible in a reasonable time. The following section discusses the symbolic differentiation when a mechanical system is formulated with flexible bodies.

## 4 Flexible multibody formulation

The governing equations of multibody systems with rigid and flexible bodies are described by the second-order differential equations

$$\mathbf{M}(\mathbf{q}, \boldsymbol{\xi})\ddot{\mathbf{q}} = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}, \boldsymbol{\xi}), \tag{28}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{q}$ denotes the generalized coordinates and $\mathbf{Q}$ is the generalized force vector. In this paper, the generalized force vector

$$\mathbf{Q} = \mathbf{Q}_\mathrm{u} + \mathbf{Q}_\mathrm{d} + \mathbf{Q}_\mathrm{g} - \mathbf{Q}_\mathrm{k} \tag{29}$$

consists of the term associated with the control $\mathbf{Q}_\mathrm{u}$, the viscous damping for joint friction $\mathbf{Q}_\mathrm{d}$, the gravity $\mathbf{Q}_\mathrm{g}$, and the elasticity $\mathbf{Q}_\mathrm{k}$. The second-order differential equations are transformed into

$$\dot{\mathbf{x}} = \mathbf{f} = \underbrace{\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1} \end{pmatrix}}_{\mathcal{M}^{-1}} \begin{pmatrix} \mathbf{v} \\ \mathbf{Q} \end{pmatrix}, \tag{30}$$

wherein the state variables are expressed by $\mathbf{x}^\mathsf{T} = \left(\mathbf{q}^\mathsf{T}, \mathbf{v}^\mathsf{T}\right) \in \mathbb{R}^n$ with the generalized velocities $\dot{\mathbf{q}} = \mathbf{v}$. In this paper, effects due to the elasticity of flexible bodies are considered by a nonlinear finite-element-based formulation using the ANCF as proposed by Omar and Shabana [34]. This standard ANCF element has been tested extensively in the literature and employed in structural-optimization problems, e.g., [15, 41]. The ANCF was developed to solve large-deformation problems in multibody dynamics [40]. Since the ANCF does not use rotational degrees of freedom, the formulation does not necessarily suffer from singularities arising from angular parameterizations. An essential advantage of the ANCF is that the mass matrix is constant with respect to the generalized coordinates, i.e., $\mathbf{M} = \mathbf{M}(\boldsymbol{\xi})$. For a detailed description of the flexible multibody formulation, the reader is referred to [34].

Employing the proposed discrete adjoint approach to study constraint sensitivities requires first-order derivatives of the governing equations with respect to the states, the control, and the set of parameters; see the procedure provided in Sect. 3.4. In this paper, the system derivatives are computed by using symbolic differentiation for efficient computation in the sensitivity analysis.

Modeling a mechanical system with a large number of structural elements results in an extensive system of governing equations. Therefore, the effort to derive the system derivatives by symbolic differentiation is enormous or not feasible in a reasonable time. Instead of directly computing the derivatives of the governing equations in (30), one can derive the global system derivatives based on the symbolic differentiation of a single structural ANCF element. The element-based derivatives are then assembled to compute the global system derivatives. Therefore, the computation of symbolic differentiations is very efficient and independent of the number of structural elements. To this end, the global mass matrix is defined by the local mass matrix of an element with superscript $(e)$

$$\mathbf{M} = \sum_{(e)} \mathbf{T}_1^{(e)^\mathsf{T}} \mathbf{M}^{(e)} \mathbf{T}_1^{(e)}, \tag{31}$$

where the element specific Boolean transformation matrix $\mathbf{T}_1^{(e)}$ maps a global variable to its local representation. Similarly, the global generalized force vector reads

$$\mathbf{Q} = \sum_{(e)} \mathbf{T}_1^{(e)^\mathsf{T}} \mathbf{Q}^{(e)}, \tag{32}$$

and the local state variables are given by

$$\mathbf{x}^{(e)} = \begin{pmatrix} \mathbf{q}^{(e)} \\ \mathbf{v}^{(e)} \end{pmatrix} = \begin{pmatrix} \mathbf{T}_1^{(e)} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_1^{(e)} \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{v} \end{pmatrix} = \tilde{\mathbf{T}}_1^{(e)} \mathbf{x}. \tag{33}$$

In addition, the local control and the local set of parameters are defined by

$$\mathbf{u}^{(e)} = \mathbf{T}_2^{(e)} \mathbf{u} \quad \text{and} \quad \boldsymbol{\xi}^{(e)} = \mathbf{T}_3^{(e)} \boldsymbol{\xi}, \tag{34}$$

respectively, with Boolean transformation matrices. The element-based formulations in (31)–(34) are used in the proceeding section to derive global first-order derivatives based on local representations.

### 4.1 Element-based derivatives for an efficient implementation of the proposed approach

The computation of the discrete adjoint equations in (17)–(19) and the discrete adjoint gradient in (21) requires the system derivatives with respect to the state variables $\partial \mathbf{f}/\partial \mathbf{x}$, the control $\partial \mathbf{f}/\partial \mathbf{u}$, and the set of parameters $\partial \mathbf{f}/\partial \boldsymbol{\xi}$. As aforementioned, an essential advantage of the ANCF is that the mass matrix does not depend on the generalized coordinates, i.e., the derivatives with respect to the state variables vanish. The constant mass matrix leads to a simplification of the derivatives of the first-order state equations (30) with respect to the state variables by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \boldsymbol{\mathcal{M}}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \frac{\partial \mathbf{Q}}{\partial \mathbf{q}} & \frac{\partial \mathbf{Q}}{\partial \mathbf{v}} \end{pmatrix}. \tag{35}$$

To derive element-based derivatives, the derivatives of the global generalized force vector can be formulated with (32) as follows:

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{x}} = \sum_{(e)} \mathbf{T}_1^{(e)^\mathsf{T}} \frac{\partial \mathbf{Q}^{(e)}}{\partial \mathbf{x}^{(e)}} \underbrace{\frac{\partial \mathbf{x}^{(e)}}{\partial \mathbf{x}}}_{\tilde{\mathbf{T}}_1^{(e)}}. \tag{36}$$

The symbolic differentiation of the global generalized force vector $\mathbf{Q}$ for all elements simplifies to the symbolic differentiation of the local generalized force vector $\mathbf{Q}^{(e)}$ for one element. The derivatives of the global generalized force vector are assembled by using the element-specific Boolean transformation matrices $\mathbf{T}_1^{(e)}$ and $\tilde{\mathbf{T}}_1^{(e)}$. Thus, the computational effort for symbolic differentiations is tremendously reduced and independent of the number of elements.

In addition, the derivatives of the first-order state equations (30) with respect to the control are given by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \boldsymbol{\mathcal{M}}^{-1} \begin{pmatrix} \mathbf{0} \\ \frac{\partial \mathbf{Q}}{\partial \mathbf{u}} \end{pmatrix}, \tag{37}$$

where the derivatives of the generalized force vector with respect to the control are formulated with

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{u}} = \sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \frac{\partial \mathbf{Q}^{(e)}}{\partial \mathbf{u}^{(e)}} \underbrace{\frac{\partial \mathbf{u}^{(e)}}{\partial \mathbf{u}}}_{\mathbf{T}_2^{(e)}}. \tag{38}$$

Finally, the derivatives of the first-order state equations (30) with respect to the set of parameters read

$$\frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \mathbf{0} \\ \frac{\partial \left(\mathbf{M}^{-1}\mathbf{Q}\right)}{\partial \boldsymbol{\xi}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \frac{\partial \mathbf{a}}{\partial \boldsymbol{\xi}} \end{pmatrix}, \tag{39}$$

with the generalized accelerations $\mathbf{a} = \mathbf{M}^{-1}\mathbf{Q}$. The derivatives of the generalized accelerations with respect to the set of parameters are difficult to compute because the mass matrix and the generalized force vector are functions of the parameters. The element-based formulation of the generalized accelerations is given by

$$\mathbf{M}\mathbf{a} = \mathbf{Q} \Rightarrow \sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \mathbf{M}^{(e)} \underbrace{\mathbf{T}_1^{(e)} \mathbf{a}}_{\mathbf{a}^{(e)}} = \sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \mathbf{Q}^{(e)}. \tag{40}$$

The matrix–vector product $\mathbf{M}^{(e)}\mathbf{a}^{(e)}$ is reformulated to avoid the direct derivative of the mass matrix with respect to the parameters as a sum of vector-scalar products, leading to

$$\sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \sum_{c} \mathbf{M}_c^{(e)} a_c^{(e)} = \sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \mathbf{Q}^{(e)}, \tag{41}$$

where the subscript $c$ denotes the column of the mass matrix and the row of generalized accelerations, respectively. The derivatives of the element-based formulation with respect to $\boldsymbol{\xi}$ read

$$\sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \sum_{c} \left( \frac{\partial \mathbf{M}_c^{(e)}}{\partial \boldsymbol{\xi}^{(e)}} \underbrace{\frac{\partial \boldsymbol{\xi}^{(e)}}{\partial \boldsymbol{\xi}}}_{\mathbf{T}_3^{(e)}} a_c^{(e)} + \mathbf{M}_c^{(e)} \frac{\partial a_c^{(e)}}{\partial \boldsymbol{\xi}} \right) = \sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \frac{\partial \mathbf{Q}^{(e)}}{\partial \boldsymbol{\xi}^{(e)}} \mathbf{T}_3^{(e)}. \tag{42}$$

By reformulation of the dyadic product of an element

$$\sum_{c} \mathbf{M}_c^{(e)} \frac{\partial a_c^{(e)}}{\partial \boldsymbol{\xi}} = \mathbf{M}^{(e)} \frac{\partial \mathbf{a}^{(e)}}{\partial \boldsymbol{\xi}} = \mathbf{M}^{(e)} \mathbf{T}_1^{(e)} \frac{\partial \mathbf{a}}{\partial \boldsymbol{\xi}}, \tag{43}$$

the derivatives of the global acceleration are computed by substituting (43) into (42) and using (31)

$$\frac{\partial \mathbf{a}}{\partial \boldsymbol{\xi}} = \mathbf{M}^{-1} \sum_{(e)} \mathbf{T}_1^{(e)\mathsf{T}} \left( \frac{\partial \mathbf{Q}^{(e)}}{\partial \boldsymbol{\xi}^{(e)}} \mathbf{T}_3^{(e)} - \sum_{c} \frac{\partial \mathbf{M}_c^{(e)}}{\partial \boldsymbol{\xi}^{(e)}} \mathbf{T}_3^{(e)} a_c^{(e)} \right). \tag{44}$$

Note that the derivatives are time-dependent functions, but the inverse of the global mass matrix only needs to be computed once for the numerical evaluation since the mass matrix is constant for the ANCF.

The element-based derivatives in (36), (38), and (44) are computed by symbolic differentiation and used to assemble the global derivatives required for the proposed discrete adjoint gradient computation. In this paper, the formulation of the flexible multibody and all required symbolic differentiations are written in the computer algebra system toolbox SymPy [30], which is written in pure Python. In addition, the analytical expressions are compiled ahead of (simulation) time with Numba [20] to efficiently evaluate the derived terms. Both packages are available under an open-source license.

## 5 Numerical examples

This section discusses three examples to demonstrate the use and advantages of the proposed discrete adjoint gradient approach. In the first example, the sensitivity analysis of an academic one-mass oscillator is analyzed. This example provides a deep insight into the proposed approach and discusses the role of the discrete adjoint variables. The second example studies the energy optimal control problem of a nonlinear spring pendulum to discuss the efficiency of the proposed approach. The third example analyzes a combined optimal control and design problem of a Selective Compliance Assembly Robot Arm (SCARA) in a rest-to-rest motion. The bodies of the robot are modeled with flexible components, where the ANCF discussed in Sect. 4 is used to examine the effects due to elasticity. Explicit and implicit integration schemes are applied to demonstrate the versatility of the proposed approach to different time integrators, with a particular focus on the computation of adjoint variables. In addition, all examples are used to verify the proposed discrete adjoint gradients using numerical computed gradients via the finite-difference method.

### 5.1 Sensitivity analysis of a one-mass oscillator

As a first example, the proposed discrete adjoint gradient approach in Sect. 3 is applied to the sensitivity analysis of an academic one-mass oscillator to demonstrate the proposed procedure in Sect. 3.4 and to discuss the role of the discrete adjoint variables. The mechanical system consists of a mass $m$, a linear damping parameter $d$, and a linear spring parameter $c$. The mass is driven by a time-dependent control $u$. The state equations are given by a linear first-order differential system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \begin{pmatrix} v \\ \frac{1}{m}(u - dv - cx) \end{pmatrix}, \tag{45}$$

wherein the state variables are expressed by $\mathbf{x} = (x,\, v)^{\mathsf{T}}$ with the position $x$ and velocity $v$ of the mass. The control variable is formulated as proposed in [22] with $u = \mathbf{C}\hat{\mathbf{u}}$, where $\mathbf{C}$ is a time-dependent cubic spline interpolation function and $\hat{\mathbf{u}} = \left(\hat{u}_0, \ldots, \hat{u}_M\right)^{\mathsf{T}}$ is a set of control grid nodes. The grid nodes are defined at the uniformly distributed macro time mesh $\hat{t}_j$, $j \in \{0, \ldots, M\}$ within the time interval $[t_0,\, t_{\mathrm{f}}]$.

This example examines sensitivities of the inequality constraint $h = f - f_{\max} \leq 0$ regarding the reaction force $f = c\, x + d\, v$ ($r = 1$ time-dependent function) with respect to the grid nodes $\hat{\mathbf{u}}$. Hence, the variables of interest in the sensitivity analysis are defined by $\mathbf{z} = \hat{\mathbf{u}}$. The reaction force is evaluated at the macro time mesh $\hat{t}_j$, resulting in the concatenated vector $\hat{\mathbf{h}} = (\hat{h}_0, \ldots, \hat{h}_M)^{\mathsf{T}}$. The reaction force depends on the state variables, which are time integrated under the influence of the control. Therefore, a change in the control grid nodes leads to a change in the reaction force. A graphical illustration of the dependencies is

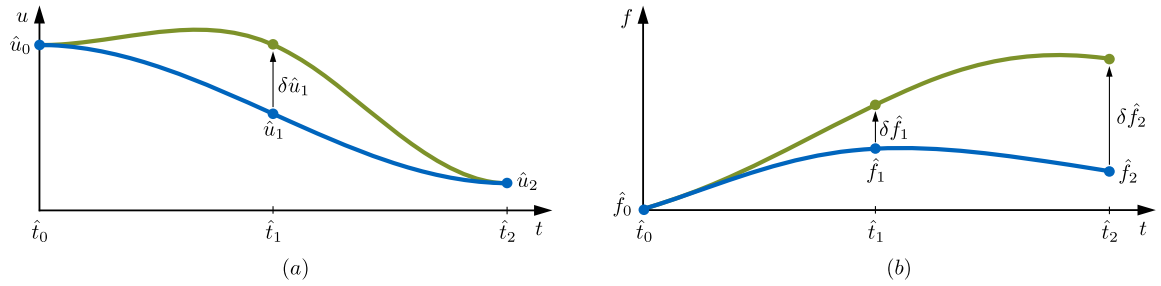



**Fig. 2** Influence of changing a control grid node on the reaction force: (a) continuous control function with grid nodes, (b) reaction force due to the control in (a)

shown in Fig. (2), where the change of the control grid node $\delta\hat{u}_1$ leads to a change of $\delta\hat{f}_1$ and $\delta\hat{f}_2$.

The sensitivity analysis uses the following set of parameters: the mass $m = 1$ kg, the damping coefficient $d = 0.5$ Ns/m, the stiffness $c = 1$ N/m, the constant time-integration step size $\Delta t = 0.001$ s, and the final time $t_\mathrm{f} = 2$ s. In addition, the control $u$ is discretized by three grid nodes ($M = 2$ uniform distributed intervals in the time interval $[t_0,\, t_\mathrm{f}]$) set to $\hat{\mathbf{u}} = (10,\, 6,\, 2)^\mathsf{T}$. The initial conditions of the state variables are defined by $\mathbf{x}_0 = \mathbf{0}$, i.e., the initial reaction force is zero.

### 5.1.1 Discrete adjoint gradient computation

The sensitivities of $\mathrm{d}\hat{\mathbf{h}}/\mathrm{d}\mathbf{z}$ are obtained by using the proposed discrete adjoint gradient in (21), which includes the discrete adjoint variables $\mathbf{R}$ defined in (17)–(19). Referring to the procedure in Sect. 3.4, the first step of the adjoint gradient computation is the symbolic differentiation of constraints. In this example, the reaction force is interpreted as an inequality constraint, and the derivative with respect to the state variables is given by $\frac{\partial h}{\partial \mathbf{x}} = (c,\, d)$. Derivatives with respect to $\boldsymbol{\xi}$ are not defined since the state equations are not parameterized. The second step of the adjoint gradient computation is the symbolic differentiation of the state equations. The derivatives with respect to the state variables and the control are represented by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{d}{m} \end{pmatrix} := \mathbf{A} \quad \text{and} \quad \frac{\partial \mathbf{f}}{\partial u} = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}, \tag{46}$$

respectively. Note that the simple structure of the linear state equations leads to constant system derivatives and, therefore, no forward integration is required to evaluate the derivatives.

The computation of the discrete adjoint variables is performed in a backward manner in which the final value defined in (17) is given in this example by

$$\mathbf{R}_N = -\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ c & d \end{pmatrix}, \tag{47}$$

where the derivative of the reaction force with respect to state variables is recognized. The explicit Euler method is used to solve the forward dynamics, and, therefore, the specific sensitivity of the right-hand side $\tilde{\mathbf{f}}$ is required, as shown in Sect. 3.2. Substituting (46) and (23) into (18) leads to the algebraic equations of the discrete adjoint variables. Starting from $\mathbf{R}_N$ and proceeding with

$$\mathbf{R}_i = \mathbf{R}_{i+1}\,(\mathbf{I} + \Delta t \mathbf{A})\,, \tag{48}$$

the discrete adjoint variables are computed in a backward manner. Note that (48) is defined within the macro time mesh $]\hat{t}_j, \hat{t}_{j+1}[$, $j \in \{0, \ldots, M-1\}$. The discrete adjoint variables at the macro time mesh $\hat{t}_j$, $j \in \{1, \ldots, M\}$ are determined by the intermediate condition in (19) given by

$$\mathbf{R}_{ej} = \mathbf{R}_{ej+1}\left(\mathbf{I} + \Delta t \mathbf{A}\right) - \mathbf{B}_{2,j}\left(c,\, d\right). \tag{49}$$

After computing the adjoint variables backward in time, the sensitivity of the inequality constraints $\hat{\mathbf{h}}$ with respect to the grid nodes $\hat{\mathbf{u}}$ can be evaluated by the proposed formulation in (21).

### 5.1.2 Interpretation

Figure 3 represents the evolution of each component of the adjoint variables $\mathbf{R}$. The adjoint variables $\mathbf{R} \in \mathbb{R}^{(p+q)\times n}$ consists of six components with $p = 0$, $q = r(M+1) = 3$, and $n = 2$. It can be observed that the adjoint variables are not necessarily smooth functions due to the intermediate condition in (49). The intermediate condition computes the adjoint variables in the same manner as in (48) and initializes the adjoint variables for the time interval $]\hat{t}_0, \hat{t}_j]$ in addition. The gradient computation of $\mathrm{d}\hat{h}_j/\mathrm{d}\hat{\mathbf{u}}$ evaluated at the time $\hat{t}_j$ is not influenced by the system dynamics in the time interval $]\hat{t}_j, \hat{t}_M]$ and, therefore, $\mathbf{R}_{j,\bullet} = \mathbf{0}$ in this interval for all $\bullet$ columns of the adjoint variables. To be more precise, the system dynamics of the gradient computation for $\hat{h}_j$ is only taken into account for the time interval $]\hat{t}_0, \hat{t}_j]$, i.e., the

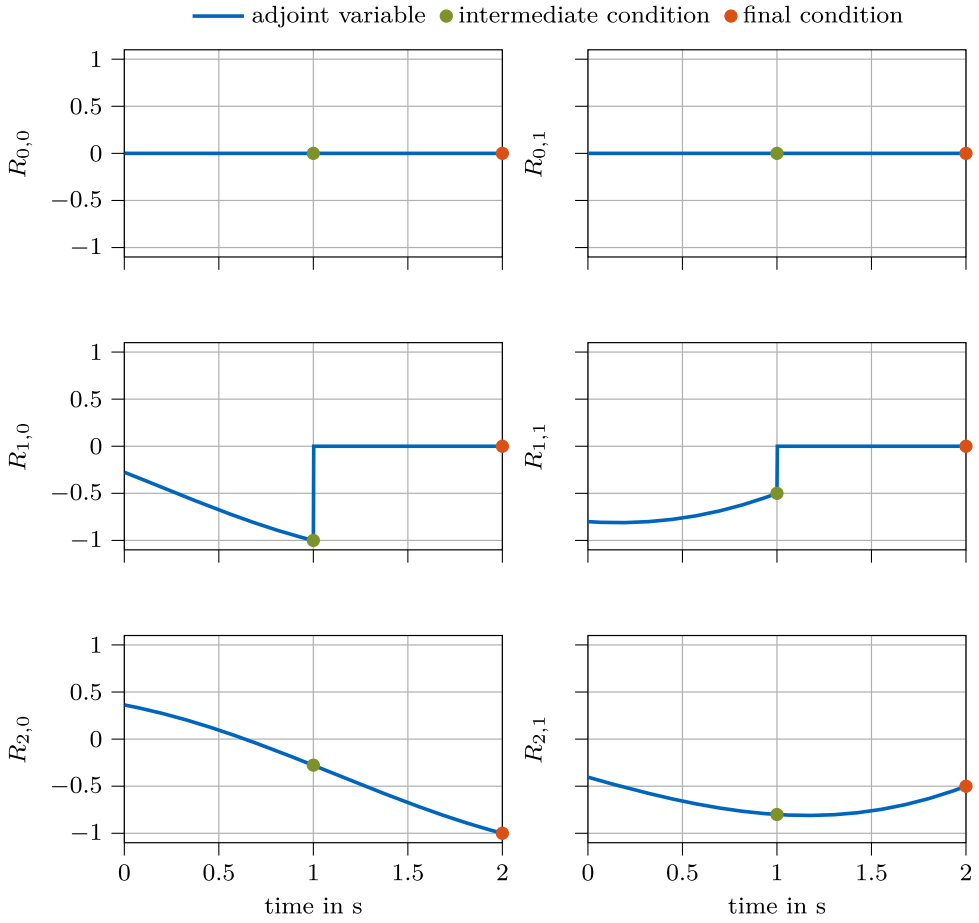



**Fig. 3** Time evolution of the discrete adjoint variables

adjoint variables for the time interval $]\hat{t}_j, \hat{t}_M]$ are zero and do not influence the gradient. This effect is seen in the second row of the adjoint variables visualized in Fig. 3. Here, the gradient computation of $\mathrm{d}\hat{h}_1/\mathrm{d}\hat{\mathbf{u}}$ evaluated at the time $\hat{t}_1 = 1$ s is not influenced by the system dynamics in the time interval $]\hat{t}_1, \hat{t}_2]$ and, therefore, the adjoint variables $R_{1,0}$ and $R_{1,1}$ are zero in this time interval. However, the discrete adjoint sensitivities are compared with the sensitivities computed by the finite-difference method to verify the proposed approach and its implementation. The sensitivity analysis results obtained by both approaches are in good agreement.

### 5.2 Energy optimal control of a nonlinear spring pendulum

The second example is focused on the energy optimal control of a nonlinear spring pendulum, as depicted in Fig. 4, inspired by the example studied in [1]. The aim is to compute a control to manipulate the mechanical system from an initial to a final state. The example studied in [1] is adapted with an additional inequality constraint during the maneuver to test the proposed discrete adjoint gradient approach. The mechanical system consists of three degrees of freedom $\mathbf{q} = (r_\mathrm{x}, r_\mathrm{y}, r_\mathrm{z})^\mathsf{T}$ describing the position vector of a mass $m$ formulated in the inertial Cartesian coordinate system. The mass is connected to the ground by a spring $c$, and the strain is formulated by using a Green–Lagrangian-type strain measure

$$\varepsilon = \frac{1}{2l_0^2}\left(\mathbf{q}^\mathsf{T}\mathbf{q} - l_0^2\right), \tag{50}$$

wherein $l_0$ denotes the strain-free spring length. The mass matrix $\mathbf{M}$ of the mechanical system is defined by using the kinetic energy

$$T = \frac{1}{2}\dot{\mathbf{q}}^\mathsf{T}\underbrace{m\mathbf{I}}_{\mathbf{M}}\dot{\mathbf{q}}, \tag{51}$$

where $\mathbf{I}$ is the identity matrix. Forces introduced due to the gravitational acceleration $g$ and the deformation of the spring are defined by

$$\mathbf{Q}_\mathrm{nl} = \left(\frac{\partial V}{\partial \mathbf{q}}\right)^\mathsf{T}. \tag{52}$$

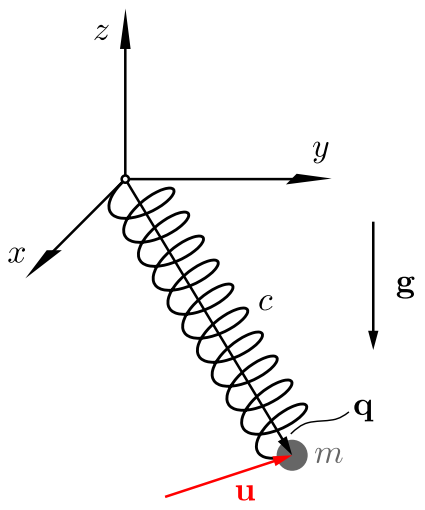



**Fig. 4** Nonlinear spring pendulum in a general configuration

The potential energy of the mechanical system reads

$$V = mgr_z + \frac{1}{2}cl_0^2\varepsilon^2, \tag{53}$$

where the first term corresponds to gravity and the second to the deformation of the spring. Note that the force term $\mathbf{Q}_{\text{nl}}$ is nonlinear due to the nonlinear Green–Lagrange strain measure in (50). In addition, the mass is driven by a time-dependent control $\mathbf{u} = (u_x,\, u_y,\, u_z)^\mathsf{T}$ and the corresponding force in the inertial Cartesian coordinate system is defined by the principle of virtual work as $\mathbf{Q}_u = \mathbf{u}$. The state equations are given by a nonlinear first-order differential system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} \mathbf{v} \\ \mathbf{M}^{-1}\,(\mathbf{Q}_u - \mathbf{Q}_{\text{nl}}) \end{pmatrix}, \tag{54}$$

wherein the state variables are expressed by introducing the velocity $\dot{\mathbf{q}} = \mathbf{v}$ of the mass by $\mathbf{x} = (\mathbf{q},\, \mathbf{v})^\mathsf{T}$. In this example, the control is formulated as proposed by Lichtenecker et al. [22] with $\mathbf{u} = \mathbf{C}\bar{\mathbf{u}}$, where $\mathbf{C} \in \mathbb{R}^{3\times 3k}$ is a time-dependent cubic spline interpolation matrix and $\bar{\mathbf{u}}^\mathsf{T} = \left(\hat{\mathbf{u}}_x^\mathsf{T},\, \hat{\mathbf{u}}_y^\mathsf{T},\, \hat{\mathbf{u}}_z^\mathsf{T}\right) \in \mathbb{R}^{3k}$ is a set of concatenated control grid nodes regarding the control functions. Each control is discretized with $k$ grid nodes defined at a uniformly distributed time mesh within the time interval $[t_0,\, t_f]$.

According to the example studied in [1], the following set of parameters is used: the mass $m = 1$ kg, the stiffness $c = 0.6$ N/m, the strain-free spring length $l_0 = 5$ m, and the gravitational acceleration $g = 9.81$ m/s$^2$. The state variables $\mathbf{x}$ are time-integrated using the explicit Euler scheme on the micro time mesh in the interval $[t_0,\, t_f]$, where the final time is $t_f = 5$ s with a constant time-integration step size $\Delta t = 0.001$ s. The initial state variables are defined by $\mathbf{x}_0 = (-2,\, -5,\, -5,\, -3,\, 0,\, 0)^\mathsf{T}$.

### 5.2.1 Optimization problem

The energy optimal control problem of mechanical systems has been studied by various authors with different formulations of the cost function. In this example, the energy optimal control problem is formulated by minimizing the signal energy required to manipulate the mechanical system. The signal energy is defined as the integrated quadratic control in the time interval $[t_0,\, t_f]$. To address the energy optimal control problem, the concatenated set of grid nodes is used to define optimization variables, i.e., $\mathbf{z} = \bar{\mathbf{u}} \in \mathbb{R}^{3k}$.

The energy optimal control problem yields the NLP problem formulated as a direct single shooting by

$$\min_{\mathbf{z}} J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}^\mathsf{T}\mathbf{u}\,\mathrm{d}t = \frac{1}{2}\mathbf{z}^\mathsf{T}\underbrace{\int_{t_0}^{t_f} \mathbf{C}^\mathsf{T}\mathbf{C}\,\mathrm{d}t}_{\mathcal{A}}\,\mathbf{z} \tag{55}$$

s.t.

$$\mathbf{z}_{\min} \le \mathbf{z} \le \mathbf{z}_{\max} \tag{56}$$

$$\mathbf{q}_N = \mathbf{q}_f \tag{57}$$

$$\dot{\mathbf{q}}_N = \mathbf{0} \tag{58}$$

$$\hat{\mathbf{l}}/\,l_{\max} \le 1 \tag{59}$$

$$\mathbf{x}_{i+1} = \tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{u}_i), \tag{60}$$

where the control in (55) is formulated with $\mathbf{u} = \mathbf{C}\mathbf{z}$. The position $\mathbf{q}_N$ and velocity $\dot{\mathbf{q}}_N$ are prescribed at the fixed final time $t_\mathrm{f}$. The final position of the mass is defined according to [1] by $\mathbf{q}_\mathrm{f} = (2, -10, -4)^\mathsf{T}$. In addition, the optimization problem concerns normalized inequality constraints regarding the spring length $l = \sqrt{\mathbf{q}^\mathsf{T}\mathbf{q}}$. The inequality constraints are considered at the uniformly distributed macro time mesh $\hat{t}_j$, $j \in \{0, \ldots, M\}$ within the time interval $[t_0, t_\mathrm{f}]$. Evaluating the spring length $(M+1)$ times leads to the concatenated vector $\hat{\mathbf{l}} \in \mathbb{R}^{M+1}$ with $M = 500$. The defined inequality constraints forces the mass to a position within a sphere with radius $r = l_\mathrm{max}$ centered on the inertial Cartesian coordinate system with $l_\mathrm{max} = 12$ m. Lower and upper bounds of the optimization variables are taken into account with $-5.5\ \mathrm{N} \leq \hat{u} \leq 5.5\ \mathrm{N}$. As an initial guess, the optimization variables are set to $\mathbf{z} = \mathbf{0}$.

For efficient numerical computation, the NLP is solved with IPOPT 3.14.12 [45] (HSL MA97 to solve linear subproblems). The cost function, constraints, and the respective first-order gradients are provided to IPOPT via an interface. The first-order gradients of the cost function (55) can be easily computed by symbolic differentiation

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{z}} = \mathbf{z}^\mathsf{T}\boldsymbol{\mathcal{A}}, \tag{61}$$

where the symmetric property of $\boldsymbol{\mathcal{A}}$ due to the block diagonal matrix $\mathbf{C}$ is utilized. The first-order gradients of the constraints (57)–(59) are computed following the proposed procedure in Sect. 3.4, implemented in Python.

### 5.2.2 Optimization results

Each control is discretized with $k = 10$ grid nodes leading to $z = 30$ optimization variables. Figure 5 shows the time evolution of the state variables obtained for the energy optimal control. The results are in accordance with the defined equality constraints in (57) and (58) at the final time. Figure 6 visualizes the obtained energy optimal control history and the inequality constraint regarding the spring length. It can be observed that the control variables are within the lower and upper bounds, while the inequality constraint in (59) is active. Similar to the previous example, the discrete adjoint gradients of the constraints are compared with the gradients computed by the finite-difference method to verify the proposed approach and its implementation. The sensitivity analysis results obtained by both approaches are in good agreement.

To demonstrate the efficiency of the proposed discrete adjoint gradient approach, the NLP problem is solved by providing gradients of the constraints to IPOPT once using the

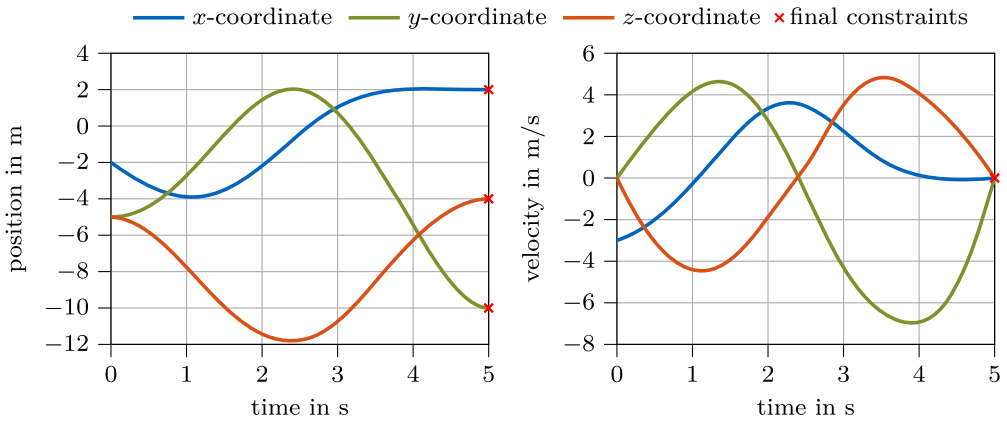



**Fig. 5** Time evolution of the state variables obtained for the energy optimal control problem

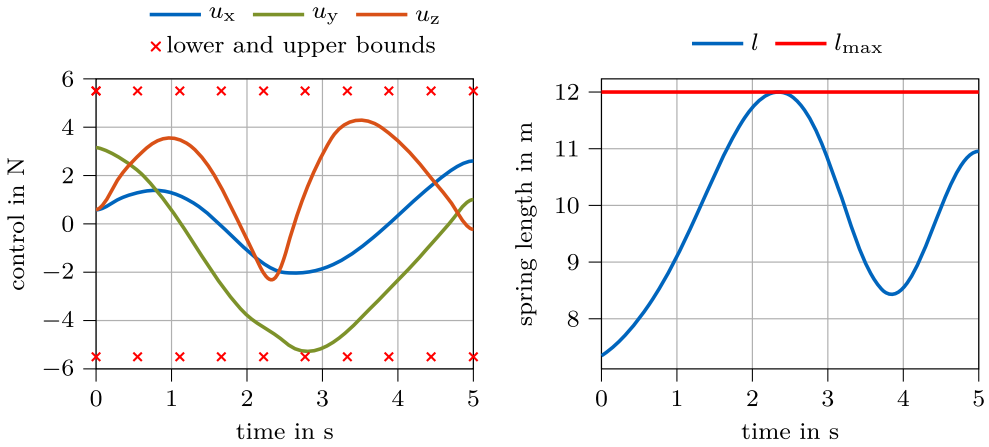



**Fig. 6** Optimal control history for the energy optimal control problem and the resulting spring length

**Table 1** Runtime comparison of two different approaches to provide first-order gradients of constraints to IPOPT for converged solutions

| Grid nodes per control | Type of gradient computation | Runtime |
|---|---|---|
| $k = 10$ | forward finite-difference | 22.3 s |
| | discrete adjoint method | 13.2 s |
| $k = 20$ | forward finite-difference | 37.8 s |
| | discrete adjoint method | 14.7 s |

finite-difference method and once using the discrete adjoint gradient approach. The runtimes required for converged solutions are given in Table 1. The comparison is performed with $k = 10$ and $k = 20$ grid nodes for each control. The proposed discrete adjoint approach outperforms the finite-difference method in terms of runtimes. Employing the discrete adjoint gradient approach, the change in the runtime required for $k = 10$ and $k = 20$ is small. In contrast, the runtime required for the finite-difference method depends strongly on the number of grid nodes. Note that the number of grid nodes $k$ influences the number of optimization variables $z = 3k$ and, therefore, the runtime for the finite-difference method. The number of iterations to fulfill the KKT conditions is equal for both approaches, which demonstrates the correct implementation of the discrete adjoint gradient approach.

### 5.3 Optimal control and design of a flexible SCARA

The third example focuses on the combined optimal control and structural optimization problem for flexible multibody systems. The idea of coupling both optimization tasks is promising to obtain the best possible mechanical structure concerning an optimal control problem. Engineers usually do not address the combined structural optimization and optimal control problem; the two challenges are typically considered independently. In addition, multibody systems with flexible components are usually underactuated, and the optimal control problem becomes more complicated than fully actuated systems [39]. Lichtenecker et al. studied in [22] the time-optimal control problem of a SCARA with flexible components. A similar system configuration is used in this example to employ the proposed discrete adjoint approach for the sensitivity analysis of a combined optimal control and structural optimization problem. Thus, a combined set of discrete adjoint gradients is used to efficiently and accurately compute first-order gradients to speed up the runtime in a direct optimization method.

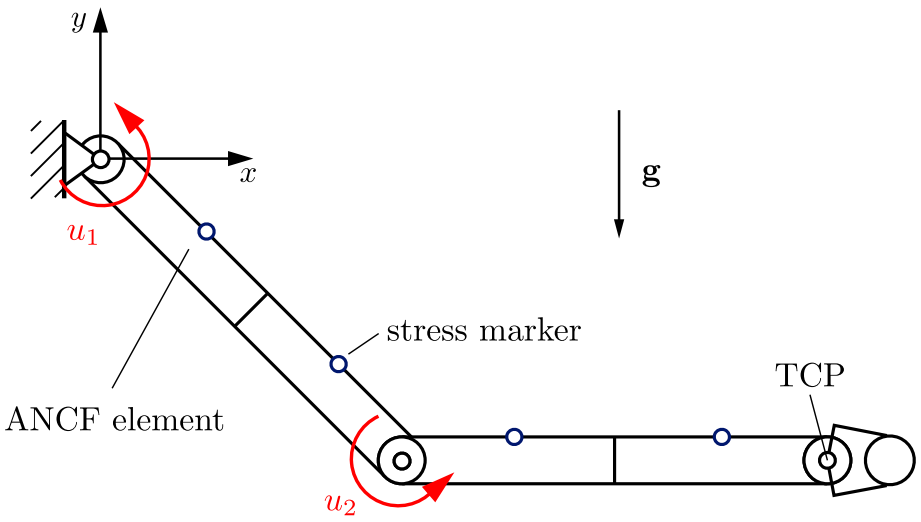



**Fig. 7** SCARA in a general undeformed configuration

Inspired by the example studied in [35], the two-arm robot depicted in Fig. 7 is analyzed in the combined optimization problem. The SCARA is driven by two controls, $u_1$ and $u_2$, in which the control variables are formulated by cubic splines similar to the previous example. In application to industrial robots, smooth trajectory planning is essential and has been presented using cubic splines, e.g., in [8, 37].

Each arm is divided into two ANCF elements, and an additional mass is attached to the tool-center point (TCP). Moreover, a stress marker is attached to each structural element to determine the equivalent stress $\sigma_{\mathrm{V}}$ while the robot performs a task. The material properties of the structural elements are set to $E = 3\mathrm{e}9\ \mathrm{N/m^2}$ and $\rho = 1300\ \mathrm{kg/m^3}$ for the Young's modulus and the density, respectively. The length of both arms is $l_1 = l_2 = 1$ m and the viscous damping coefficient in the revolute joints is set to $d_1 = d_2 = 0.2$ Nm s/rad. The width of the structural elements is set to $w^{(e)} = 0.002$ m. In addition, the system is affected by the gravity field, and an additional mass $m_{\mathrm{E}} = 1$ kg is attached to the TCP. The state variables $\mathbf{x}$ are integrated using the implicit Euler scheme on the micro time mesh in the interval $[t_0,\ t_{\mathrm{f}}]$, where the final time is $t_{\mathrm{f}} = 2$ s with a constant time-integration step size $\Delta t = 0.001$ s.

### 5.3.1 Optimization problem

Structural-optimization problems can be treated with so-called weakly and fully coupled methods [42]. Weakly coupled methods are based on equivalent static loads, while fully coupled methods incorporate the system dynamics into the optimization process. This example focuses on extending fully coupled methods to embed the optimal control of flexible multibody systems. Considering an optimal control problem and a structural optimization problem leads to a combined set of optimization variables, including the control parameterization and design parameter of the multibody system. In this example, the structural elements are parameterized by the height $h^{(e)}$, while the width $w^{(e)}$ and the length $l^{(e)}$ are set to constant values. The set of design parameters regarding the multibody system depicted in Fig. 7 is defined by $\boldsymbol{\xi}^{\mathsf{T}} = \left(h^{(1)},\ h^{(2)},\ h^{(3)},\ h^{(4)}\right)$. The optimal control problem of flexible multibody systems requires smooth control functions to reduce vibrations. In this example, a continuity requirement up to $\mathcal{C}^2$ of the control is enforced, similar to the previous example in Sect. 5.2, by a cubic spline interpolation. Both continuous control functions $u_1$ and $u_2$ are discretized with $k = 21$ grid nodes at a uniformly distributed time mesh within the time

interval $[t_0, t_f]$. The discretization of the control leads to the set of grid nodes $\bar{\mathbf{u}}^\mathsf{T} = \left(\hat{\mathbf{u}}_1^\mathsf{T}, \hat{\mathbf{u}}_2^\mathsf{T}\right)$. Concatenating the parameterization of the multibody system and the control leads to the combined set of optimization variables $\mathbf{z}^\mathsf{T} = \left(\boldsymbol{\xi}^\mathsf{T}, \bar{\mathbf{u}}^\mathsf{T}\right)$.

Minimizing the mass of mechanical systems is a common approach in structural optimization to enable innovative lightweight designs. In this paper, the mass minimization yields the NLP problem formulated as direct single shooting by

$$\min_{\mathbf{z}} m = \rho \sum_{(e)} w^{(e)}\, h^{(e)}\, l^{(e)} \tag{62}$$

s.t.

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} \tag{63}$$

$$\mathbf{r}_{\mathrm{TCP},N} = \mathbf{r}_\mathrm{f} \tag{64}$$

$$\dot{\mathbf{r}}_{\mathrm{TCP},N} = \mathbf{0} \tag{65}$$

$$\hat{\boldsymbol{\sigma}}_\mathrm{V} / \sigma_{\mathrm{Vmax}} \leq 1 \tag{66}$$

$$\mathbf{x}_{i+1} = \tilde{\mathbf{f}}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{u}_i, \mathbf{u}_{i+1}, \boldsymbol{\xi}), \tag{67}$$

with special attention to position and velocity constraints of the TCP, $\mathbf{r}_{\mathrm{TCP},N}$, and $\dot{\mathbf{r}}_{\mathrm{TCP},N}$, respectively, at the fixed final time $t_f$. The final position of the TCP is defined by $\mathbf{r}_\mathrm{f} = (1,\, 1)^\mathsf{T}$. In addition, the optimization problem concerns normalized inequality constraints regarding the equivalent stress $\sigma_V$ of the four markers attached to the structure. All four markers are evaluated at the macro time mesh $\hat{t}_j,\ j \in \{0, \ldots, M\}$ with $M = 100$ leading to the concatenated equivalent stress vector $\hat{\boldsymbol{\sigma}}_\mathrm{V}$. The upper limit of the equivalent stress is $\sigma_{\mathrm{Vmax}} = 1.1\mathrm{e}7\ \mathrm{N/m^2}$. Lower and upper bounds of the optimization variables are taken into account with $0.002\ \mathrm{m} \leq h^{(e)} \leq 0.02\ \mathrm{m}$ and $-5\ \mathrm{Nm} \leq \hat{u} \leq 5\ \mathrm{Nm}$, respectively. Regarding the initial conditions of the state variables, the robot is defined in the undeformed configuration, where both arms are hanging vertically downward with generalized velocities equal to zero.

In terms of initializing the NLP problem, a two-stage procedure is utilized with decoupling of the optimization variables. The first stage solves an optimal control problem with a constant height of the structural elements, i.e., the optimization variables $\mathbf{z} = \bar{\mathbf{u}}$ consists of the grid nodes. The aim is to identify a control with a defined set of parameters $\boldsymbol{\xi}$ that manipulates the robot from the initial configuration so that the TCP satisfies the constraints (64) and (65) at the final time. As an initial guess for this first-stage optimization, the control grid nodes are set to $\hat{u} = 0\ \mathrm{Nm}$ and the constant height of the elements is defined by $h^{(e)} = 0.02\ \mathrm{m}$. The preoptimized set of control grid nodes $\bar{\mathbf{u}}^*$ and heights of the elements $h^{(e)}$ are employed as an initial guess to the combined optimization problem.

For efficient numerical computation, the NLP is solved with IPOPT 3.14.12 [45] (HSL MA97 to solve linear subproblems). The cost function, constraints, and the respective first-order gradients are provided to IPOPT via an interface. The first-order gradients of the cost function (62) can be easily computed by symbolic differentiation, while the first-order gradients of the constraints (64)–(66) are computed following the proposed procedure in Sect. 3.4, implemented in Python.

### 5.3.2 Optimization results

Figure 8 shows the obtained control history for both controls. It can be observed that the control is within the lower and upper bounds, while the stress constraint in (66) is active.

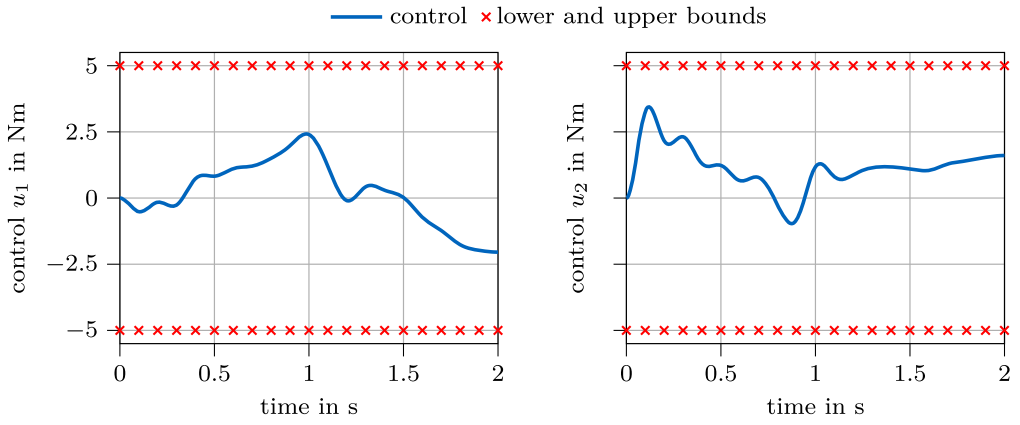



**Fig. 8** Optimal control history of a flexible two-arm robot for a rest-to-rest maneuver

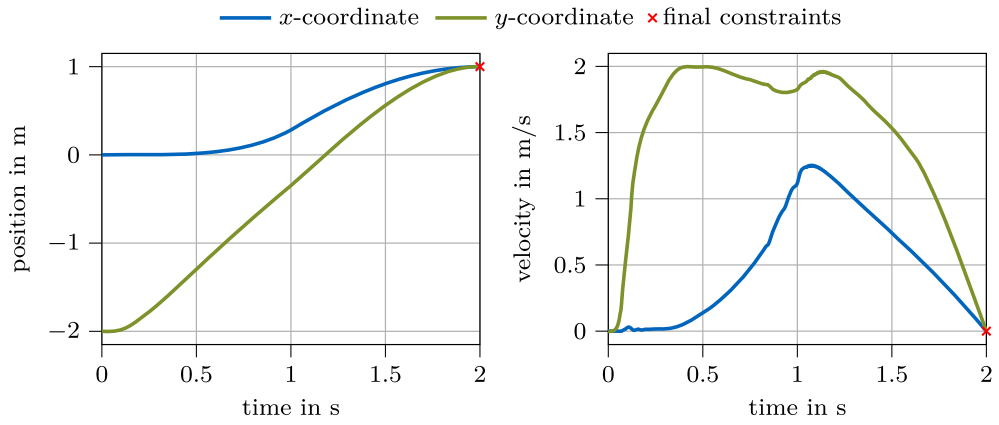



**Fig. 9** Time evolution of the position and velocity of the TCP obtained for the optimal control and design problem

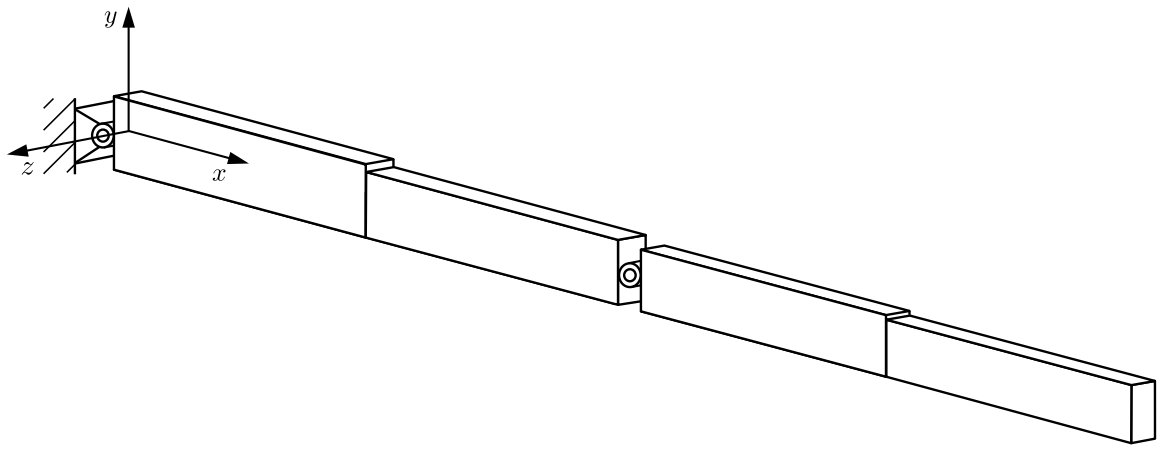



**Fig. 10** Final sizing of the flexible two-arm robot

Applying the optimal control to the SCARA leads to the position and velocity of the TCP as shown in Fig. 9. Note there are no acceleration constraints defined at the final time in this example. It can be observed that the equality constraints in (64) and (65) are fulfilled at the final time. Figure 10 visualizes the final sizing of the structural elements. The final sizing provides a structure in which the height of the structural elements becomes smaller towards

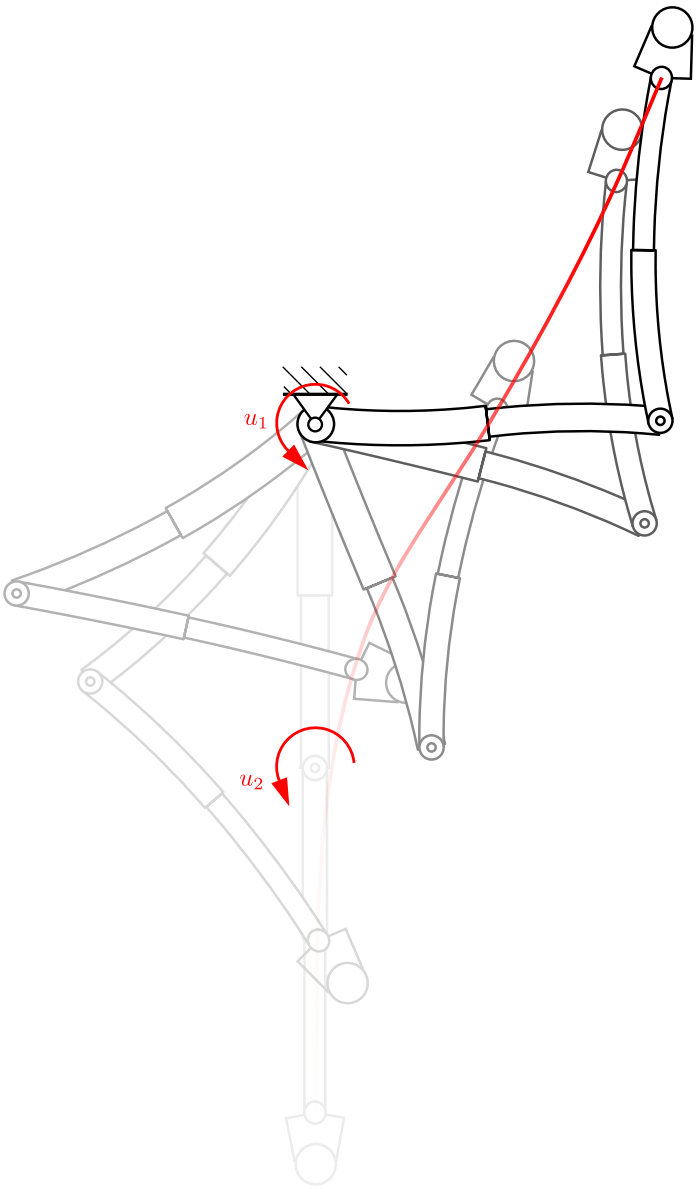



**Fig. 11** Snapshots of the robot's motion history

the TCP; thus, the bending stiffness also becomes smaller towards the TCP. The initial mass of the robot $m_{\text{init}} = 0.1040\,\text{kg}$ is reduced by the final design to $m^* = 0.0450\,\text{kg}$, which corresponds to a reduction of 56.7% regarding the initial mass. With the final design and the corresponding control, the SCARA undergoes a large deformation during manipulation. However, the results are in accordance with the constraints posed in (63)–(67) and provide a local minimum of the robot's mass. Snapshots of the robot's motion are illustrated in Fig. 11.

It has to be mentioned that normalizing inequality constraints is essential since the numerical value of the equality constraints in (64) and (65) are relatively small compared to those of the inequality constraint in (66). Without normalizing, the optimization would not converge to a local minimum. Similar to the previous examples, the discrete adjoint gradients are verified by applying the finite-difference method. The sensitivity analysis results obtained by both approaches are in good agreement. The proposed discrete adjoint gradient approach is an efficient technique to incorporate a large number of optimization variables, and the computational effort is less than using the finite-difference method. A comparison of the runtimes required to compute first-order gradients of the constraints is performed to demonstrate the efficiency of the proposed discrete adjoint gradient approach. The gradients

are computed with the optimal set of optimization variables $\mathbf{z}^*$ ten times in a row. The average runtime required for the finite-difference method is $t_{\mathrm{FDM}} = 592.4$ s, while the average runtime required for the discrete adjoint method is $t_{\mathrm{DAM}} = 52.2$ s. The proposed approach reduces the computational time by 91.2% regarding the finite-difference method. The high computational time in the finite-difference method is because the number of state equations to be solved depends on the number of optimization variables. It has to be emphasized that the computation of gradients is required at each iteration of the NLP solver package, which encourages the use of the proposed discrete adjoint gradient method to improve computational efficiency.

## 6 Conclusion

This paper discusses adjoint-based sensitivity analysis for dynamic systems in gradient-based optimization problems. Deriving adjoint gradients is mathematically more laborious than simply computing finite differences for numerical gradients. However, the significant time advantage when using adjoint gradients for the sensitivity analysis justifies the considerable preprocessing effort. This paper presents a novel discrete adjoint gradient approach to incorporate (in)equality constraints. Moreover, the paper shows the application of different time-integration schemes, highlighting their efficiency and applicability to large-scale problems. Three numerical examples are investigated to show the application of the proposed discrete adjoint gradient approach. The sensitivity analysis of an academic example discusses the role of the discrete adjoint variables. The energy optimal control problem of a nonlinear spring pendulum studies the efficiency of the proposed approach. In addition, the proposed discrete adjoint gradients are utilized in a coupled optimal control and optimal design problem in flexible multibody dynamics.

**Acknowledgements** Daniel Lichtenecker and Karin Nachbagauer acknowledge support from the Technical University of Munich – Institute for Advanced Study.

**Author contributions** D. Lichtenecker performed simulation and analysis, and wrote the main manuscript text. K. Nachbagauer supervised the project. Both authors jointly conceived the presented ideas. Both authors reviewed the manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

### Declarations

**Competing interests** The authors declare no competing interests.



## References

1. Betsch, P., Becker, C.: Conservation of generalized momentum maps in mechanical optimal control problems with symmetry. Int. J. Numer. Methods Eng. **111**(2), 144–175 (2017). https://doi.org/10.1002/nme.5459

2. Betts, J.T.: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2nd edn. SIAM, Philadelphia (2010). https://doi.org/10.1137/1.9780898718577
3. Boopathy, K., Kennedy, G.: Adjoint-based derivative evaluation methods for flexible multibody systems with rotorcraft applications. In: 55th AIAA Aerospace Sciences Meeting (2017). https://doi.org/10.2514/6.2017-1671
4. Bryson, A.E., Ho, Y.C.: Applied Optimal Control: Optimization, Estimation, and Control. Taylor & Francis, New York (1975). https://doi.org/10.1201/9781315137667
5. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. Wiley, New York (2016). https://doi.org/10.1002/9781119121534
6. Callejo, A., Sonneville, V., Bauchau, O.A.: Discrete adjoint method for the sensitivity analysis of flexible multibody systems. J. Comput. Nonlinear Dyn. **14**(2), 021001 (2019). https://doi.org/10.1115/1.4041237
7. Cao, Y., Li, S., Petzold, L., Serban, R.: Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution. SIAM J. Sci. Comput. **24**(3), 1076–1089 (2003). https://doi.org/10.1137/S1064827501380630
8. Constantinescu, D., Croft, E.A.: Smooth and time–optimal trajectory planning for industrial manipulators along specified paths. J. Robot. Syst. **17**(5), 233–249 (2000). https://doi.org/10.1002/(SICI)1097-4563(200005)17:5<233::AID-ROB1>3.0.CO;2-Y
9. Della Santina, C., Duriez, C., Rus, D.: Model-based control of soft robots: a survey of the state of the art and open challenges. IEEE Control Syst. Mag. **43**(3), 30–65 (2023). https://doi.org/10.1109/MCS.2023.3253419
10. Eberhard, P.: Adjoint variable method for sensitivity analysis of multibody systems interpreted as a continuous, hybrid form of automatic differentiation. In: Proceedings of the Second International Workshop of Computational Differentiation, pp. 12–14 (1996)
11. Ebrahimi, M., Butscher, A., Cheong, H., Iorio, F.: Design optimization of dynamic flexible multibody systems using the discrete adjoint variable method. Comput. Struct. **213**, 82–99 (2019). https://doi.org/10.1016/j.compstruc.2018.12.007
12. Gufler, V., Wehrle, E., Zwölfer, A.: A review of flexible multibody dynamics for gradient-based design optimization. Multibody Syst. Dyn. **53**(4), 379–409 (2021). https://doi.org/10.1007/s11044-021-09802-z
13. Haug, E.J., Wehage, R.A., Mani, N.K.: Design sensitivity analysis of large-scale constrained dynamic mechanical systems. J. Mech. Transm. Autom. Des. **106**(2), 156–162 (1984). https://doi.org/10.1115/1.3258573
14. Hawkes, E.W., Majidi, C., Tolley, M.T.: Hard questions for soft robotics. Sci. Robot. **6**(53), eabg6049 (2021). https://doi.org/10.1126/scirobotics.abg6049
15. Held, A., Seifried, R.: Gradient-based optimization of flexible multibody systems using the absolute nodal coordinate formulation. In: Proceedings of the 6th ECCOMAS Thematic Conference on Multibody Dynamics (2013)
16. Held, A., Seifried, R.: Adjoint sensitivity analysis of multibody system equations in state-space representation obtained by QR decomposition. In: Proceedings of the 11th ECCOMAS Thematic Conference on Multibody Dynamics (2023)
17. Johnston, L., Patel, V.: Second-order sensitivity methods for robustly training recurrent neural network models. IEEE Control Syst. Lett. **5**(2), 529–534 (2021). https://doi.org/10.1109/LCSYS.2020.3001498
18. Karush, W.: Minima of functions of several variables with inequalities as side constraints. Master's thesis, Department of Mathematics, University of Chicago (1939)
19. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492 (1951)
20. Lam, S.K., Pitrou, A., Seibert, S.: Numba: a LLVM-based python JIT compiler. In: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. Assoc. Comput. Mach., New York (2015). https://doi.org/10.1145/2833157.2833162
21. Lauß, T., Oberpeilsteiner, S., Steiner, W., Nachbagauer, K.: The discrete adjoint method for parameter identification in multibody system dynamics. Multibody Syst. Dyn. **42**(4), 397–410 (2018). https://doi.org/10.1007/s11044-017-9600-9
22. Lichtenecker, D., Rixen, D., Eichmeir, P., Nachbagauer, K.: On the use of adjoint gradients for time-optimal control problems regarding a discrete control parameterization. Multibody Syst. Dyn. **59**(3), 313–334 (2023). https://doi.org/10.1007/s11044-023-09898-5
23. Lichtenecker, D., Eichmeir, P., Nachbagauer, K.: On the usage of analytically computed adjoint gradients in a direct optimization for time-optimal control problems. In: Optimal Design and Control of Multibody Systems. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-50000-8_14
24. Lions, J.L.: Optimal Control of Systems Governed by Partial Differential Equations. Springer, New York (1971)

25. López Varela, Á., Sandu, C., Sandu, A., Dopico Dopico, D.: Discrete adjoint variable method for the sensitivity analysis of ALI3-P formulations. Multibody Syst. Dyn. (2023). https://doi.org/10.1007/s11044-023-09911-x
26. Maciąg, P., Malczyk, P., Frączek, J.: Hamiltonian direct differentiation and adjoint approaches for multibody system sensitivity analysis. Int. J. Numer. Methods Eng. **121**(22), 5082–5100 (2020). https://doi.org/10.1002/nme.6512
27. Maciąg, P., Malczyk, P., Frączek, J.: Adjoint-based feedforward control of two-degree-of-freedom planar robot. In: Proceedings of the 11th ECCOMAS Thematic Conference on Multibody Dynamics (2023)
28. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidiscip. Optim. **26**(6), 369–395 (2004). https://doi.org/10.1007/s00158-003-0368-6
29. Martins, J., Hwang, J.T.: Review and unification of methods for computing derivatives of multidisciplinary computational models. AIAA J. **51**(11), 2582–2599 (2013). https://doi.org/10.2514/1.J052184
30. Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J.K., Singh, S., Rathnayake, T., Vig, S., Granger, B.E., Muller, R.P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M.J., Terrel, A.R., Roučka, Š., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., Scopatz, A.: SymPy: symbolic computing in Python. PeerJ Comput. Sci. **3**, e103 (2017). https://doi.org/10.7717/peerj-cs.103
31. Nachbagauer, K., Oberpeilsteiner, S., Sherif, K., Steiner, W.: The use of the adjoint method for solving typical optimization problems in multibody dynamics. J. Comput. Nonlinear Dyn. **10**(6), 061011 (2015). https://doi.org/10.1115/1.4028417
32. Nadarajah, S., Jameson, A.: A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In: 38th Aerospace Sciences Meeting and Exhibit (2000). https://doi.org/10.2514/6.2000-667
33. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5
34. Omar, M.A., Shabana, A.A.: A two-dimensional shear deformable beam for large rotation and deformation problems. J. Sound Vib. **243**(3), 565–576 (2001). https://doi.org/10.1006/jsvi.2000.3416
35. Oral, S., Kemal Ider, S.: Optimum design of high-speed flexible robotic arms with dynamic behavior constraints. Comput. Struct. **65**(2), 255–259 (1997). https://doi.org/10.1016/S0045-7949(96)00269-6
36. Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., Edelman, A.: Universal Differential Equations for Scientific Machine Learning (2020) https://doi.org/10.48550/arXiv.2001.04385. arXiv:2001.04385. ArXiv preprint
37. Reiter, A., Müller, A., Gattringer, H.: On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators. IEEE Trans. Ind. Inform. **14**(4), 1681–1690 (2018). https://doi.org/10.1109/TII.2018.2792002
38. Schneider, S., Betsch, P.: On optimal control problems in redundant coordinates. In: Proceedings of the 11th ECCOMAS Thematic Conference on Multibody Dynamics (2023)
39. Seifried, R.: Dynamics of underactuated multibody systems: modeling, control and optimal design. In: Solid Mechanics and Its Applications. Springer, Berlin (2014). https://doi.org/10.1007/978-3-319-01228-5
40. Shabana, A.A.: Definition of the slopes and the finite element absolute nodal coordinate formulation. Multibody Syst. Dyn. **1**(3), 339–348 (1997). https://doi.org/10.1023/A:1009740800463
41. Sun, J., Tian, Q., Hu, H.: Structural optimization of flexible components in a flexible multibody system modeled via ANCF. Mech. Mach. Theory **104**, 59–80 (2016). https://doi.org/10.1016/j.mechmachtheory.2016.05.008
42. Tromme, E., Brüls, O., Duysinx, P.: Weakly and fully coupled methods for structural optimization of flexible mechanisms. Multibody Syst. Dyn. **38**(4), 391–417 (2016). https://doi.org/10.1007/s11044-015-9493-4
43. Tromme, E., Held, A., Duysinx, P., Brüls, O.: System-based approaches for structural optimization of flexible mechanisms. Arch. Comput. Methods Eng. **25**(3), 817–844 (2018). https://doi.org/10.1007/s11831-017-9215-6
44. Vanpaemel, S., Vermaut, M., Desmet, W., Naets, F.: Input optimization for flexible multibody systems using the adjoint variable method and the flexible natural coordinates formulation. Multibody Syst. Dyn. **57**(3), 259–277 (2023). https://doi.org/10.1007/s11044-023-09874-z
45. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 15–57 (2006). https://doi.org/10.1007/s10107-004-0559-y

# Bibliography


[1] Allen, L., O'Connell, A., and Kiermer, V. "How can we ensure visibility and diversity in research contributions? How the Contributor Role Taxonomy (CRediT) is helping the shift from authorship to contributorship". In: *Learned Publishing* 32.1 (2019), pp. 71–74. DOI: 10.1002/leap.1210.

[2] Armanini, C., Boyer, F., Mathew, A. T., Duriez, C., and Renda, F. "Soft robots modeling: A structured overview". In: *IEEE Transactions on Robotics* 39.3 (2023), pp. 1728–1748. DOI: 10.1109/TRO.2022.3231360.

[3] Arora, J. S. and Haug, E. J. "Efficient optimal design of structures by generalized steepest descent programming". In: *International Journal for Numerical Methods in Engineering* 10.4 (1976), pp. 747–766. DOI: 10.1002/nme.1620100404.

[4] Arora, J. S. and Haug, E. J. "Methods of design sensitivity analysis in structural optimization". In: *AIAA Journal* 17.9 (1979), pp. 970–974. DOI: 10.2514/3.61260.

[5] Ascher, U., Christiansen, J., and Russell, R. D. "A collocation solver for mixed order systems of boundary value problems". In: *Mathematics of Computation* 33.146 (1979), pp. 659–679. DOI: 10.2307/2006301.

[6] Ascher, U., Mattheij, R. M. M., and Russell, R. D. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Society for Industrial and Applied Mathematics, 1995. DOI: 10.1137/1.9781611971231.

[7] Bauchau, O. A. *Flexible Multibody Dynamics*. Vol. 176. Solid Mechanics and Its Applications. Springer, Dordrecht, 2011. DOI: 10.1007/978-94-007-0335-3.

[8] Bellman, R. E. *Dynamic Programming*. Princeton University Press, 1957.

[9] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V. "Direct trajectory optimization and costate estimation via an orthogonal collocation method". In: *Journal of Guidance, Control, and Dynamics* 29.6 (2006), pp. 1435–1440. DOI: 10.2514/1.20478.

[10] Bertsekas, D. P. *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific, 1995.

[11] Berzeri, M. and Shabana, A. A. "Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation". In: *Journal of Sound and Vibration* 235.4 (2000), pp. 539–565. DOI: 10.1006/jsvi.1999.2935.

[12] Bestle, D. and Eberhard, P. "Analyzing and optimizing multibody systems". In: *Mechanics of Structures and Machines* 20.1 (1992), pp. 67–92. DOI: 10.1080/08905459208905161.

[13] Bestle, D. and Seybold, J. "Sensitivity analysis of constrained multibody systems". In: *Archive of Applied Mechanics* 62.3 (1992), pp. 181–190. DOI: 10.1007/BF00787958.

[14] Betts, J. T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2nd Edition. Society for Industrial and Applied Mathematics, 2010. DOI: 10.1137/1.9780898718577.

[15] Betts, J. T. and Huffman, W. P. “Trajectory optimization on a parallel processor”. In: *Journal of Guidance, Control, and Dynamics* 14.2 (1991), pp. 431–439. DOI: 10.2514/3.20656.

[16] Bhalerao, K. D., Poursina, M., and Anderson, K. S. “An efficient direct differentiation approach for sensitivity analysis of flexible multibody systems”. In: *Multibody System Dynamics* 23.2 (2010), pp. 121–140. DOI: 10.1007/s11044-009-9176-0.

[17] Biegler, L. T. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Society for Industrial and Applied Mathematics, 2010. DOI: 10.1137/1.9780898719383.

[18] Biggs, M. C. “Constrained minimization using recursive equality quadratic programming”. In: *Numerical Methods for Nonlinear Optimization* (1972), pp. 411–428.

[19] Bittner, M. “Utilization of Problem and Dynamic Characteristics for Solving Large Scale Optimal Control Problems”. PhD thesis. Technical University of Munich, 2017.

[20] Blonigan, P. J. and Wang, Q. “Multiple shooting shadowing for sensitivity analysis of chaotic dynamical systems”. In: *Journal of Computational Physics* 354 (2018), pp. 447–475. DOI: 10.1016/j.jcp.2017.10.032.

[21] Bock, H. G. “Numerical solution of nonlinear multipoint boundary value problems with applications to optimal control”. In: *Zeitschrift für Angewandte Mathematik und Mechanik* 58.7 (1978), pp. 407–409.

[22] Bock, H. G. and Plitt, K. J. “A multiple shooting algorithm for direct solution of optimal control problems”. In: *Proceedings of the IFAC World Congress*. 1984, pp. 242–247.

[23] Bolza, O. *Lectures on the Calculus of Variations*. University of Chicago Press, 1904.

[24] Bordalba, R., Schoels, T., Ros, L., Porta, J. M., and Diehl, M. “Direct collocation methods for trajectory optimization in constrained robotic systems”. In: *IEEE Transactions on Robotics* 39.1 (2023), pp. 183–202. DOI: 10.1109/TRO.2022.3193776.

[25] Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

[26] Broyden, C. G. “The convergence of a class of double-rank minimization algorithms 1. General considerations”. In: *IMA Journal of Applied Mathematics* 6.1 (1970), pp. 76–90. DOI: 10.1093/imamat/6.1.76.

[27] Brüls, O., Lemaire, E., Duysinx, P., and Eberhard, P. “Optimization of multibody systems and their structural components”. In: *Multibody Dynamics: Computational Methods and Applications*. Ed. by Arczewski, K., Blajer, W., Frączek, J., and Wojtyra, M. Vol. 23. Computational Methods in Applied Sciences. Springer, Dordrecht, 2011, pp. 49–68. DOI: 10.1007/978-90-481-9971-6_3.

[28] Bryson, A. E. and Denham, W. F. “Optimal programming problems with inequality constraints II: Solution by steepest-ascent”. In: *AIAA Journal* 2.1 (1964), pp. 23–34. DOI: 10.2514/3.2209.

[29] Bryson, A. E., Denham, W. F., Carroll, F. J., and Mikami, K. “Determination of lift or drag programs to minimize re-entry heating”. In: *Journal of the Aerospace Sciences* 29.4 (1962), pp. 420–430. DOI: 10.2514/8.9476.

[30] Bryson, A. E. and Ho, Y. C. *Applied Optimal Control: Optimization, Estimation, and Control*. Taylor & Francis, 1975. DOI: 10.1201/9781315137667.

[31] Bulirsch, R. “Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung”. In: *Report of the Carl-Cranz-Gesellschaft* 251 (1971).

[32] Butcher, J. C. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016. DOI: 10.1002/9781119121534.

[33] Byrd, R. H., Gilbert, J. C., and Nocedal, J. “A trust region method based on interior point techniques for nonlinear programming”. In: *Mathematical Programming* 89.1 (2000), pp. 149–185. DOI: 10.1007/PL00011391.

[34] Byrd, R. H., Hribar, M. E., and Nocedal, J. “An interior point algorithm for large-scale nonlinear programming”. In: *SIAM Journal on Optimization* 9.4 (1999), pp. 877–900. DOI: 10.1137/S1052623497325107.

[35] Byrd, R. H., Liu, G., and Nocedal, J. “On the local behavior of an interior point method for nonlinear programming”. In: *Numerical Analysis* (1997), pp. 37–56.

[36] Byrd, R. H., Nocedal, J., and Waltz, R. A. “KNITRO: An integrated package for nonlinear optimization”. In: *Large-Scale Nonlinear Optimization*. Ed. by Di Pillo, G. and Roma, M. Vol. 83. Nonconvex Optimization and Its Applications. Springer, Boston, 2006, pp. 35–59. DOI: 10.1007/0-387-30065-1_4.

[37] Callejo, A., Sonneville, V., and Bauchau, O. A. “Discrete adjoint method for the sensitivity analysis of flexible multibody systems”. In: *Journal of Computational and Nonlinear Dynamics* 14.2 (2019), p. 021001. DOI: 10.1115/1.4041237.

[38] Cao, Y., Li, S., Petzold, L., and Serban, R. “Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution”. In: *SIAM Journal on Scientific Computing* 24.3 (2003), pp. 1076–1089. DOI: 10.1137/S1064827501380630.

[39] Chen, Y., Scarabottolo, N., Bruschetta, M., and Beghi, A. “Efficient move blocking strategy for multiple shooting–based non–linear model predictive control”. In: *IET Control Theory and Applications* 14.2 (2020), pp. 343–351. DOI: 10.1049/iet-cta.2019.0168.

[40] Corner, S., Sandu, A., and Sandu, C. “Adjoint sensitivity analysis of hybrid multibody dynamical systems”. In: *Multibody System Dynamics* 49.4 (2020), pp. 395–420. DOI: 10.1007/s11044-020-09726-0.

[41] Davis, P. J. and Rabinowitz, P. *Methods of Numerical Integration*. Academic Press, 1984. DOI: 10.1016/C2013-0-10566-1.

[42] Della Santina, C., Duriez, C., and Rus, D. “Model-based control of soft robots: A survey of the state of the art and open challenges”. In: *IEEE Control Systems Magazine* 43.3 (2023), pp. 30–65. DOI: 10.1109/MCS.2023.3253419.

[43] Dias, J. M. P. and Pereira, M. S. “Sensitivity analysis of rigid-flexible multibody systems”. In: *Multibody System Dynamics* 1.3 (1997), pp. 303–322. DOI: 10.1023/A:1009790202712.

[44] Dibold, M., Gerstmayr, J., and Irschik, H. “A detailed comparison of the absolute nodal coordinate and the floating frame of reference formulation in deformable multibody systems”. In: *Journal of Computational and Nonlinear Dynamics* 4.2 (2009), p. 021006. DOI: 10.1115/1.3079825.

[45] Dickmanns, E. D. and Well, K. H. “Approximate solution of optimal control problems using third order Hermite polynomial functions”. In: *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*. Ed. by Marchuk, G. I. Vol. 27. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 1975, pp. 158–166. DOI: 10.1007/3-540-07165-2_21.

[46] Diehl, M., Bock, H. G., Diedam, H., and Wieber, P. B. “Fast direct multiple shooting algorithms for optimal robot control”. In: *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*. Ed. by Diehl, M. and Mombaur, K. Vol. 340. Lecture Notes in Control and Information Sciences. Springer, Berlin, Heidelberg, 2006, pp. 65–93. DOI: 10.1007/978-3-540-36119-0_4.

[47] Diehl, M., Bock, H. G., and Schlöder, J. P. “A real-time iteration scheme for nonlinear optimization in optimal feedback control”. In: *SIAM Journal on Control and Optimization* 5.43 (2005), pp. 1714–1736. DOI: 10.1137/S0363012902400713.

[48] Djeddi, R. and Ekici, K. “FDOT: A fast, memory-efficient and automated approach for discrete adjoint sensitivity analysis using the operator overloading technique”. In: *Aerospace Science and Technology* 91 (2019), pp. 159–174. DOI: 10.1016/j.ast.2019.05.004.

[49] Dopico, D., Zhu, Y., Sandu, A., and Sandu, C. “Direct and adjoint sensitivity analysis of ordinary differential equation multibody formulations”. In: *Journal of Computational and Nonlinear Dynamics* 10.1 (2014), p. 011012. DOI: 10.1115/1.4026492.

[50] Dwivedy, S. K. and Eberhard, P. “Dynamic analysis of flexible manipulators, a literature review”. In: *Mechanism and Machine Theory* 41.7 (2006), pp. 749–777. DOI: 10.1016/j.mechmachtheory.2006.01.014.

[51] Eberhard, P. “Adjoint variable method for sensitivity analysis of multibody systems interpreted as a continuous, hybrid form of automatic differentiation”. In: *Proceedings of the Second International Workshop of Computational Differentiation*. 1996, pp. 12–14.

[52] Eberhard, P. and Schiehlen, W. “Computational dynamics of multibody systems: History, formalisms, and applications”. In: *Journal of Computational and Nonlinear Dynamics* 1.1 (2005), pp. 3–12. DOI: 10.1115/1.1961875.

[53] Eichmeir, P., Nachbagauer, K., Lauß, T., Sherif, K., and Steiner, W. “Time-optimal control of dynamic systems regarding final constraints”. In: *Journal of Computational and Nonlinear Dynamics* 16.3 (2021), p. 031003. DOI: 10.1115/1.4049334.

[54] Enright, P. J. and Conway, B. A. “Discrete approximations to optimal trajectories using direct transcription and nonlinear programming”. In: *Journal of Guidance, Control, and Dynamics* 15.4 (1992), pp. 994–1002. DOI: 10.2514/3.20934.

[55] Escalona, J. L., Hussien, H. A., and Shabana, A. A. “Application of the absolute nodal co-ordinate formulation to multibody system dynamics”. In: *Journal of Sound and Vibration* 214.5 (1998), pp. 833–851. DOI: 10.1006/jsvi.1998.1563.

[56] Fiacco, A. V. and McCormick, G. P. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990. DOI: 10.1137/1.9781611971316.

[57] Fletcher, R. “A new approach to variable metric algorithms”. In: *The Computer Journal* 13.3 (1970), pp. 317–322. DOI: 10.1093/comjnl/13.3.317.

[58] Fletcher, R. and Leyffer, S. “Nonlinear programming without a penalty function”. In: *Mathematical Programming* 91.2 (2002), pp. 239–269. DOI: 10.1007/s101070100244.

[59] Frisch, K. R. “The logarithmic potential method of convex programming”. In: *Memorandum, University Institute of Economics, Oslo* 5.6 (1955).

[60] Gear, C. W., Gupta, G. K., and Leimkuhler, B. “Automatic integration of Euler-Lagrange equations with constraints”. In: *Journal of Computational and Applied Mathematics* 12/13 (1985), pp. 77–90. DOI: 10.1016/0377-0427(85)90008-1.

[61] Géradin, M. and Cardano, A. *Flexible Multibody Dynamics: A Finite Element Approach*. Wiley, 2001.

[62] Gerstmayr, J., Matikainen, M. K., and Mikkola, A. M. “A geometrically exact beam element based on the absolute nodal coordinate formulation”. In: *Multibody System Dynamics* 20.4 (2008), pp. 359–384. DOI: 10.1007/s11044-008-9125-3.

[63] Gerstmayr, J., Sugiyama, H., and Mikkola, A. M. “Review on the absolute nodal coordinate formulation for large deformation analysis of multibody systems”. In: *Journal of Computational and Nonlinear Dynamics* 8.3 (2013), p. 031016. DOI: 10.1115/1.4023487.

[64] Gholami, A., Keutzer, K., and Biros, G. “ANODE: Unconditionally accurate memory-efficient gradients for neural ODEs”. In: *arXiv preprint arXiv:1902.10298* (2019). DOI: 10.48550/arXiv.1902.10298.

[65] Giles, M. B. and Pierce, N. A. “An introduction to the adjoint approach to design”. In: *Flow, Turbulence and Combustion* 65.3 (2000), pp. 393–415. DOI: 10.1023/A:1011430410075.

[66] Gill, P. E., Murray, W., and Saunders, M. A. “SNOPT: An SQP algorithm for large-scale constrained optimization”. In: *SIAM Review* 47.1 (2005), pp. 99–131. DOI: 10.1137/S0036144504446096.

[67] Gill, P. E., Murray, W., Saunders, M. A., and Wong, E. “User’s guide for SNOPT version 7.7: Software for large-scale nonlinear programming”. Department of Mathematics, University of California, San Diego, La Jolla, CA, 2019.

[68] Gill, P. E., Murray, W., Saunders, M. A., and Wong, E. “User’s guide for SQOPT version 7.7: Software for large-scale linear and quadratic programming”. Department of Mathematics, University of California, San Diego, La Jolla, CA, 2021.

[69] Gill, P. E., Murray, W., and Wright, M. H. *Practical Optimization*. Society for Industrial and Applied Mathematics, 2019. DOI: 10.1137/1.9781611975604.

[70] Goldfarb, D. “A family of variable-metric methods derived by variational means”. In: *Mathematics of Computation* 24.109 (1970), pp. 23–26. DOI: 10.1090/S0025-5718-1970-0258249-6.

[71] Gould, N. I. M., Orban, D., Sartenaer, A., and Toint, P. L. “Superlinear convergence of primal-dual interior point algorithms for nonlinear programming”. In: *SIAM Journal on Optimization* 11.4 (2001), pp. 974–1002. DOI: 10.1137/S1052623400370515.

[72] Griewank, A. “Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation”. In: *Optimization Methods and Software* 1.1 (1992), pp. 35–54. DOI: 10.1080/10556789208805505.

[73] Griewank, A. and Walther, A. “Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation”. In: *ACM Transactions on Mathematical Software* 26.1 (2000), pp. 19–45. DOI: 10.1145/347837.347846.

[74] Gufler, V., Wehrle, E., and Zwölfer, A. “A review of flexible multibody dynamics for gradient-based design optimization”. In: *Multibody System Dynamics* 53.4 (2021), pp. 379–409. DOI: 10.1007/s11044-021-09802-z.

[75] Gufler, V., Zwölfer, A., and Wehrle, E. “Analytical derivatives of flexible multibody dynamics with the floating frame of reference formulation”. In: *Multibody System Dynamics* 60.2 (2024), pp. 257–288. DOI: 10.1007/s11044-022-09858-5.

[76] Hairer, E., Wanner, G., and Lubich, C. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Vol. 31. Springer Series in Computational Mathematics. Springer, Berlin, Heidelberg, 2006. DOI: 10.1007/3-540-30666-8.

[77] Han, S. P. “Superlinearly convergent variable metric algorithms for general nonlinear programming problems”. In: *Mathematical Programming* 11.1 (1976), pp. 263–282. DOI: 10.1007/BF01580395.

[78] Han, S. P. “A globally convergent method for nonlinear programming”. In: *Journal of Optimization Theory and Applications* 22.3 (1977), pp. 297–309. DOI: 10.1007/BF00932858.

[79] Hargraves, C. R. and Paris, S. W. “Direct trajectory optimization using nonlinear programming and collocation”. In: *Journal of Guidance, Control, and Dynamics* 10.4 (1987), pp. 338–342. DOI: 10.2514/3.20223.

[80] Haug, E. J. and Arora, J. S. “Optimal mechanical design techniques based on optimal control methods”. In: *ASME Paper No. 64-DTT-10, Proceedings of the 1st ASME Design Technology Transfer Conference*. 1974, pp. 64–74.

[81] Haug, E. J., Wehage, R. A., and Mani, N. K. “Design sensitivity analysis of large-scale constrained dynamic mechanical systems”. In: *Journal of Mechanisms, Transmissions, and Automation in Design* 106.2 (1984), pp. 156–162. DOI: 10.1115/1.3258573.

[82] Held, A., Knüfer, S., and Seifried, R. “Structural sensitivity analysis of flexible multibody systems modeled with the floating frame of reference approach using the adjoint variable method”. In: *Multibody System Dynamics* 40.3 (2017), pp. 287–302. DOI: 10.1007/s11044-016-9540-9.

[83] Held, A. and Seifried, R. “Gradient-based optimization of flexible multibody systems using the absolute nodal coordinate formulation”. In: *Proceedings of the 6th ECCOMAS Thematic Conference on Multibody Dynamics*. 2013.

[84] Held, A. and Seifried, R. “Adjoint sensitivity analysis of multibody system equations in state-space representation obtained by QR decomposition”. In: *Proceedings of the 11th ECCOMAS Thematic Conference on Multibody Dynamics*. Ed. by Ambrósio, J., Flores, P., Quental, C., and Magalhães, H. Instituo Superior Técnico Lisboa, 2023. URL: https://multibody2023.tecnico.ulisboa.pt/prog_MULTIBODY_WEB/MULTIBODY2023_PAPERS/ID_223_548_eccomas2023_full_paper.pdf (visited on 02/02/2024).

[85] Hicks, G. A. and Ray, W. H. “Approximation methods for optimal control synthesis”. In: *The Canadian Journal of Chemical Engineering* 49.4 (1971), pp. 522–528. DOI: 10.1002/cjce.5450490416.

[86] Hu, P. “A note on the adjoint method for neural ordinary differential equation network”. In: *arXiv preprint arXiv:2402.15141* (2024). DOI: 10.48550/arXiv.2402.15141.

[87] Jameson, A. “Aerodynamic design via control theory”. In: *Journal of Scientific Computing* 3.3 (1988), pp. 233–260. DOI: 10.1007/BF01061285.

[88] Johnston, L. and Patel, V. “Second-order sensitivity methods for robustly training recurrent neural network models”. In: *IEEE Control Systems Letters* 5.2 (2021), pp. 529–534. DOI: 10.1109/LCSYS.2020.3001498.

[89] Karush, W. “Minima of functions of several variables with inequalities as side constraints”. Master’s thesis. Department of Mathematics, University of Chicago, 1939.

[90] Keller, H. B. *Numerical Methods for Two-Point Boundary-Value Problems*. Courier Dover Publications, 2018.

[91] Kelley, H. J. “Method of gradients”. In: *Mathematics in Science and Engineering* 5 (1962), pp. 205–254. DOI: 10.1016/S0076-5392(08)62094-9.

[92] Kenway, G. K. W., Mader, C. A., He, P., and Martins, J. R. R. A. “Effective adjoint approaches for computational fluid dynamics”. In: *Progress in Aerospace Sciences* 110 (2019), p. 100542. DOI: 10.1016/j.paerosci.2019.05.002.

[93] Kierzenka, J. and Shampine, L. F. “A BVP solver based on residual control and the Matlab PSE”. In: *ACM Transactions on Mathematical Software* 27.3 (2001), pp. 299–316. DOI: 10.1145/502800.502801.

[94] Kirches, C., Wirsching, L., Bock, H. G., and Schlöder, J. P. “Efficient direct multiple shooting for nonlinear model predictive control on long horizons”. In: *Journal of Process Control* 22.3 (2012), pp. 540–550. DOI: 10.1016/j.jprocont.2012.01.008.

[95] Kirk, D. E. *Optimal Control Theory: An Introduction*. New York: Dover Publications, 2004.

[96] Kraft, D. “A software package for sequential quadratic programming”. In: *Technical Report No. DFVLR-FB 88-28, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*. 1988.

[97] Kuhn, H. W. and Tucker, A. W. “Nonlinear programming”. In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by Neyman, J. 1951, pp. 481–492.

[98] Lagrange, J. L. “Leçon Cinquieme. Sur l’usage des courbes dans la solution des problemes”. In: *Leçons Elémentaires sur les Mathématiques* (1795).

[99] Lichtenecker, D., Eichmeir, P., and Nachbagauer, K. “On the usage of analytically computed adjoint gradients in a direct optimization for time-optimal control problems”. In: *Optimal Design and Control of Multibody Systems*. Ed. by Nachbagauer, K. and Held, A. Vol. 42. IUTAM Bookseries. Springer, Cham, 2024, pp. 153–164. DOI: 10.1007/978-3-031-50000-8_14.

[100] Lichtenecker, D., Krög, G., Gattringer, H., and Müller, A. “Several approaches for the optimization of arm motions of humanoids”. In: *Proceedings of the Joint Austrian Computer Vision and Robotics Workshop 2020*. Ed. by Roth, P. M., Steinbauer, G., Fraundorfer, F., Brandstötter, M., and Perko, R. Verlag der Technischen Universität, September 17-18, Graz, 2020, pp. 59–63. DOI: 10.3217/978-3-85125-752-6-15.

[101] Lichtenecker, D. and Nachbagauer, K. “A discrete adjoint gradient approach for equality and inequality constraints in dynamics”. In: *Multibody System Dynamics* 61.1 (2024), pp. 103–130. DOI: 10.1007/s11044-024-09965-5.

[102] Lichtenecker, D., Rixen, D., Eichmeir, P., and Nachbagauer, K. “On the use of adjoint gradients for time-optimal control problems regarding a discrete control parameterization”. In: *Multibody System Dynamics* 59.3 (2023), pp. 313–334. DOI: 10.1007/s11044-023-09898-5.

[103] Lions, J. L. *Optimal Control of Systems Governed by Partial Differential Equations*. Vol. 170. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, Heidelberg, 1971.

[104] López Varela, Á., Sandu, C., Sandu, A., and Dopico Dopico, D. "Discrete adjoint variable method for the sensitivity analysis of ALI3-P formulations". In: *Multibody System Dynamics* (2023). DOI: 10.1007/s11044-023-09911-x.

[105] Maciąg, P., Malczyk, P., and Frączek, J. "Joint–coordinate adjoint method for optimal control of multibody systems". In: *Multibody System Dynamics* 56.4 (2022), pp. 401–425. DOI: 10.1007/s11044-022-09851-y.

[106] Maciąg, P., Malczyk, P., and Frączek, J. "Adjoint-based feedforward control of two-degree-of-freedom planar robot". In: *Proceedings of the 11th ECCOMAS Thematic Conference on Multibody Dynamics*. Ed. by Ambrósio, J., Flores, P., Quental, C., and Magalhães, H. Instituo Superior Técnico Lisboa, 2023. URL: https://multibody2023.tecnico.ulisboa.pt/prog_MULTIBODY_WEB/MULTBODY2023_ABSTRACTS/ID_166_331_ECCOMAS-2023-abstract-rev.pdf (visited on 02/02/2024).

[107] Mani, N. K. and Haug, E. J. "Singular value decomposition for dynamic system design sensitivity analysis". In: *Engineering with Computers* 1.2 (1985), pp. 103–109. DOI: 10.1007/BF01200068.

[108] Marchuk, G. I. *Adjoint Equations and Analysis of Complex Systems*. Vol. 295. Mathematics and Its Applications. Springer, Dordrecht, 2013. DOI: 10.1007/978-94-017-0621-6.

[109] Martins, J. R. R. A. and Hwang, J. T. "Review and unification of methods for computing derivatives of multidisciplinary computational models". In: *AIAA Journal* 51.11 (2013), pp. 2582–2599. DOI: 10.2514/1.J052184.

[110] Matsubara, T., Miyatake, Y., and Yaguchi, T. "The symplectic adjoint method: Memory-efficient backpropagation of neural-network-based differential equations". In: *IEEE Transactions on Neural Networks and Learning Systems* (2023), pp. 1–13. DOI: 10.1109/TNNLS.2023.3242345.

[111] McNamara, A., Treuille, A., Popović, Z., and Stam, J. "Fluid control using the adjoint method". In: *ACM Transactions on Graphics* 23.3 (2004), pp. 449–456. DOI: 10.1145/1015706.1015744.

[112] Nachbagauer, K. "State of the art of ANCF elements regarding geometric description, interpolation strategies, definition of elastic forces, validation and the locking phenomenon in comparison with proposed beam finite elements". In: *Archives of Computational Methods in Engineering* 21.3 (2014), pp. 293–319. DOI: 10.1007/s11831-014-9117-9.

[113] Nachbagauer, K., Oberpeilsteiner, S., Sherif, K., and Steiner, W. "The use of the adjoint method for solving typical optimization problems in multibody dynamics". In: *Journal of Computational and Nonlinear Dynamics* 10.6 (2015), p. 061011. DOI: 10.1115/1.4028417.

[114] Nadarajah, S. and Jameson, A. "A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization". In: *38th Aerospace Sciences Meeting and Exhibit* (2000). DOI: 10.2514/6.2000-667.

[115] Nocedal, J. and Wright, S. J. *Numerical Optimization*. 2nd Edition. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006. DOI: 10.1007/978-0-387-40065-5.

[116] Ober-Blöbaum, S. “Discrete Mechanics and Optimal Control”. PhD thesis. University of Paderborn, 2008.

[117] Omar, M. A. and Shabana, A. A. “A two-dimensional shear deformable beam for large rotation and deformation problems”. In: *Journal of Sound and Vibration* 243.3 (2001), pp. 565–576. DOI: 10.1006/jsvi.2000.3416.

[118] Otsuka, K., Makihara, K., and Sugiyama, H. “Recent advances in the absolute nodal coordinate formulation: Literature review from 2012 to 2020”. In: *Journal of Computational and Nonlinear Dynamics* 17.8 (2022), p. 080803. DOI: 10.1115/1.4054113.

[119] Petzold, L., Li, S., Cao, Y., and Serban, R. “Sensitivity analysis of differential-algebraic equations and partial differential equations”. In: *Computers & Chemical Engineering* 30.10-12 (2006), pp. 1553–1559. DOI: 10.1016/j.compchemeng.2006.05.015.

[120] Pfeiffer, F. *Mechanical System Dynamics*. Vol. 40. Lecture Notes in Applied and Computational Mechanics. Springer, Berlin, Heidelberg, 2008. DOI: 10.1007/978-3-540-79436-3.

[121] Pi, T., Zhang, Y., and Chen, L. “First order sensitivity analysis of flexible multibody systems using absolute nodal coordinate formulation”. In: *Multibody System Dynamics* 27.2 (2012), pp. 153–171. DOI: 10.1007/s11044-011-9269-4.

[122] Pironneau, O. “On optimum design in fluid mechanics”. In: *Journal of Fluid Mechanics* 64.1 (1974), pp. 97–110. DOI: 10.1017/S0022112074002023.

[123] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F. *The Mathematical Theory of Optimal Processes*. New York: John Wiley & Sons, 1962.

[124] Powell, M. J. D. “A fast algorithm for nonlinearly constrained optimization calculations”. In: *Numerical Analysis*. Ed. by Watson, G. A. Vol. 630. Springer, Berlin, Heidelberg, 1978, pp. 144–157. DOI: 10.1007/BFb0067703.

[125] Powell, M. J. D. “The convergence of variable metric methods for nonlinearly constrained optimization calculations”. In: *Nonlinear Programming 3*. Ed. by Mangasarian, O. L., Meyer, R. R., and Robinson, S. M. Academic Press, 1978, pp. 27–63. DOI: 10.1016/B978-0-12-468660-1.50007-4.

[126] Pytlak, R. *Numerical Methods for Optimal Control Problems with State Constraints*. Vol. 1707. Lecture Notes in Mathematics. Springer, Berlin, Heidelberg, 1999. DOI: 10.1007/BFb0097244.

[127] Qin, L., Peng, H., Huang, X., Liu, M., and Huang, W. “Modeling and simulation of dynamics in soft robotics: A review of numerical approaches”. In: *Current Robotics Reports* 5.1 (2024), pp. 1–13. DOI: 10.1007/s43154-023-00105-z.

[128] Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. “Universal differential equations for scientific machine learning”. In: *arXiv preprint arXiv:2001.04385* (2020). DOI: 10.48550/arXiv.2001.04385.

[129] Rao, A. V. “A survey of numerical methods for optimal control”. In: *Advances in the Astronautical Sciences* 135.1 (2009), pp. 497–528.

[130] Russo, M., Sadati, S. M. H., Dong, X., Mohammad, A., Walker, I. D., Bergeles, C., Xu, K., and Axinte, D. A. “Continuum robots: An overview”. In: *Advanced Intelligent Systems* 5.5 (2023), p. 2200367. DOI: 10.1002/aisy.202200367.

[131] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D. Saisana, M., and Tarantola, S. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, 2008. DOI: 10.1002/9780470725184.

[132] Schiehlen, W. *Multibody Systems Handbook*. Springer, Berlin, Heidelberg, 1990. DOI: 10.1007/978-3-642-50995-7.

[133] Schittkowski, K. *NLPQLP: A fortran implementation of a sequential quadratic programming algorithm with distributed and non-monotone line search, version 4.2*. 2015. URL: https://klaus-schittkowski.de/NLPQLP.pdf (visited on 04/30/2024).

[134] Schneider, S. and Betsch, P. "On optimal control problems in redundant coordinates". In: *Proceedings of the 11th ECCOMAS Thematic Conference on Multibody Dynamics*. Ed. by Ambrósio, J., Flores, P., Quental, C., and Magalhães, H. Instituo Superior Técnico Lisboa, 2023. URL: https://multibody2023.tecnico.ulisboa.pt/prog_MULTIBODY_WEB/MULTIBODY2023_PAPERS/ID_195_541_ECCOMAS_MBD_2023_SS_PB_proceeding.pdf (visited on 02/02/2024).

[135] Seifried, R. *Dynamics of Underactuated Multibody Systems: Modeling, Control and Optimal Design*. Vol. 205. Solid Mechanics and Its Applications. Springer, Cham, 2014. DOI: 10.1007/978-3-319-01228-5.

[136] Shabana, A. A. "An absolute nodal coordinate formulation for the large rotation and deformation analysis of flexible bodies". In: *Technical Report No. MBS96–1-UIC, Department of Mechanical Engineering, University of Illinois at Chicago*. 1996.

[137] Shabana, A. A. "Definition of the slopes and the finite element absolute nodal coordinate formulation". In: *Multibody System Dynamics* 1.3 (1997), pp. 339–348. DOI: 10.1023/A:1009740800463.

[138] Shabana, A. A. "Flexible multibody dynamics: Review of past and recent developments". In: *Multibody System Dynamics* 1.2 (1997), pp. 189–222. DOI: 10.1023/A:1009773505418.

[139] Shabana, A. A. "Definition of ANCF finite elements". In: *Journal of Computational and Nonlinear Dynamics* 10.5 (2015), p. 054506. DOI: 10.1115/1.4030369.

[140] Shabana, A. A. *Computational Continuum Mechanics*. John Wiley & Sons, 2018. DOI: 10.1002/9781119293248.

[141] Shabana, A. A. "Continuum-based geometry/analysis approach for flexible and soft robotic systems". In: *Soft Robotics* 5.5 (2018), pp. 613–621. DOI: 10.1089/soro.2018.0007.

[142] Shabana, A. A. *Dynamics of Multibody Systems*. Cambridge: Cambridge University Press, 2020. DOI: 10.1017/9781108757553.

[143] Shabana, A. A. "An overview of the ANCF approach, justifications for its use, implementation issues, and future research directions". In: *Multibody System Dynamics* 58.3 (2023), pp. 433–477. DOI: 10.1007/s11044-023-09890-z.

[144] Shabana, A. A. and Eldeeb, A. E. "Motion and shape control of soft robots and materials". In: *Nonlinear Dynamics* 104.1 (2021), pp. 165–189. DOI: 10.1007/s11071-021-06272-y.

[145] Shabana, A. A., Hussien, H. A., and Escalona, J. L. "Application of the absolute nodal coordinate formulation to large rotation and large deformation problems". In: *Journal of Mechanical Design* 120.2 (1998), pp. 188–195. DOI: 10.1115/1.2826958.

[146] Shabana, A. A. and Schwertassek, R. "Equivalence of the floating frame of reference approach and finite element formulations". In: *International Journal of Non-Linear Mechanics* 33.3 (1998), pp. 417–432. DOI: 10.1016/S0020-7462(97)00024-3.

[147] Shanno, D. F. "Conditioning of quasi-Newton methods for function minimization". In: *Mathematics of Computation* 24.111 (1970), pp. 647–656. DOI: 10.2307/2004840.

[148] Shiri, B., Wu, G. C., and Baleanu, D. “Terminal value problems for the nonlinear systems of fractional differential equations”. In: *Applied Numerical Mathematics* 170 (2021), pp. 162–178. DOI: 10.1016/j.apnum.2021.06.015.

[149] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*. 3nd Edition. Vol. 12. Texts in Applied Mathematics. Springer, New York, 1992. DOI: 10.1007/978-0-387-21738-3.

[150] Sun, J., Tian, Q., and Hu, H. “Structural optimization of flexible components in a flexible multibody system modeled via ANCF”. In: *Mechanism and Machine Theory* 104 (2016), pp. 59–80. DOI: 10.1016/j.mechmachtheory.2016.05.008.

[151] Tasseff, B., Coffrin, C., Wächter, A., and Laird, C. “Exploring benefits of linear solver parallelism on modern nonlinear optimization applications”. In: *arXiv: Optimization and Control* (2019). DOI: 10.48550/arXiv.1909.08104.

[152] The MathWorks Inc. *Optimization Toolbox version: 9.4 (R2022b)*. Natick, Massachusetts, United States, 2022. URL: https://www.mathworks.com (visited on 04/03/2024).

[153] Tortorelli, D. A. and Michaleris, P. “Design sensitivity analysis: Overview and review”. In: *Inverse Problems in Engineering* 1.1 (1994), pp. 71–105. DOI: 10.1080/174159794088027573.

[154] van Keulen, F., Haftka, R. T., and Kim, N. H. “Review of options for structural design sensitivity analysis. Part 1: Linear systems”. In: *Computer Methods in Applied Mechanics and Engineering* 194.30 (2005), pp. 3213–3243. DOI: 10.1016/j.cma.2005.02.002.

[155] Vanderbei, R. J. and Shanno, D. F. “An interior-point algorithm for nonconvex nonlinear programming”. In: *Computational Optimization and Applications* 13.1 (1999), pp. 231–252. DOI: 10.1023/A:1008677427361.

[156] Vanpaemel, S., Vermaut, M., Desmet, W., and Naets, F. “Input optimization for flexible multibody systems using the adjoint variable method and the flexible natural coordinates formulation”. In: *Multibody System Dynamics* 57.3 (2023), pp. 259–277. DOI: 10.1007/s11044-023-09874-z.

[157] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. “SciPy 1.0: Fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[158] von Stryk, O. “Numerical solution of optimal control problems by direct collocation”. In: *Optimal Control: Calculus of Variations, Optimal Control Theory and Numerical Methods*. Ed. by Bulirsch, R., Miele, A., Stoer, J., and Well, K. Vol. 111. ISNM International Series of Numerical Mathematics. Birkhäuser Basel, 1993, pp. 129–143. DOI: 10.1007/978-3-0348-7539-4_10.

[159] von Stryk, O. and Bulirsch, R. “Direct and indirect methods for trajectory optimization”. In: *Annals of Operations Research* 37.1 (1992), pp. 357–373. DOI: 10.1007/BF02071065.

[160] Wächter, A. and Biegler, L. T. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (2006), pp. 15–57. DOI: 10.1007/s10107-004-0559-y.

[161] Waltz, R. A., Morales, J. L., Nocedal, J., and Orban, D. “An interior algorithm for nonlinear optimization that combines line search and trust region steps”. In: *Mathematical Programming* 107.3 (2006), pp. 391–408. DOI: 10.1007/s10107-004-0560-5.

[162] Wehrle, E. and Gufler, V. “Analytical sensitivity analysis of dynamic problems with direct differentiation of generalized-$\alpha$ time integration”. In: *Machines* 12.2 (2024). DOI: 10.3390/machines12020128.

[163] Wilson, R. B. “A Simplicial Method for Convex Programming”. PhD thesis. Harvard University, 1963.

[164] Xavier, M. S., Tawk, C. D., Zolfagharian, A., Pinskier, J., Howard, D., Young, T., Lai, J., Harrison, S. M., Yong, Y. K., Bodaghi, M., and Fleming, A. J. “Soft pneumatic actuators: A review of design, fabrication, modeling, sensing, control and applications”. In: *IEEE Access* 10 (2022), pp. 59442–59485. DOI: 10.1109/ACCESS.2022.3179589.