Technische Universität München
Zentrum Mathematik
Lehrstuhl für Mathematische Optimierung

Modeling and numerical solution of inverse optimal control problems for the analysis of human motions

Sebastian Albrecht

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. Martin Brokate
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. Michael Ulbrich
	2.	Univ.-Prof. Dr.-Ing./(Univ. Tokio) Martin Buss
	3.	Univ.-Prof. Dr. Matthias Gerdts Universität der Bundeswehr München

Die Dissertation wurde am 23.01.2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 09.07.2013 angenommen.

Acknowledgements

This doctoral thesis addresses a class of mathematical problems arising in real-world applications. Since research of various disciplines tackles different aspects of the overall problem, this work is the result of an interdisciplinary effort. Due to the sometimes contradicting goals of these disciplines, the task of finding a consistent approach was challenging. Completing this thesis would not have been possible without the substantial and outright support of certain people:

First, I would like to thank my advisor Prof. Dr. Michael Ulbrich for giving me the opportunity to analyze such an interesting problem of applied mathematics at his chair. His ideas and critical remarks were immensely valuable in finding solution strategies and for advancing the status quo.

Furthermore, I am deeply grateful to my second advisor Prof. Dr.-Ing./(Univ. Tokio) Martin Buss for enabling a very close cooperation with his chair of the electrical engineering department and for agreeing to examine the work. Many thanks also go to Dr.-Ing. Marion Leibold for her steady advice with respect to realizing the numerical approaches in the application examples and for her input in the course of the completion of this thesis.

I also want to express my gratitude to Prof. Dr. Matthias Gerdts for examining my work.

The colleagues of both chairs, the chair of mathematical optimization and the institute of automatic control engineering, always provided a nice and friendly atmosphere which I enjoyed during the long hours of research. I thank them for their discussions and support over the years and for those diverting moments which were not strictly related to mathematics.

Naturally, I want to mention my cooperation partners here: Specifically, Carolina Passenberg from the institute of automatic control engineering, Sven Kraus from the institute of automotive technology, the team of Dr. Stefan Glasauer at the center for sensorimotor research and the intelligent autonomous system group of Prof. Dr./(Yale Univ.) Michael Beetz. They all provided interesting application scenarios for the developed mathematical approach; I am extremely grateful that they took care of the hardware issues and the data recording and processing.

Finally, I want to thank my friends and my family for their understanding and steady support in all those years where research consumed most of my time. Especially, this work would never have been finished without the immense backing of my wife Michaela.

Zusammenfassung

In dieser Arbeit werden inverse Optimalsteuerungsprobleme für unterschiedliche Anwendungen, die alle menschliche Bewegungen betrachten, modelliert, eine Lösungsmethode wird analysiert und numerische Ergebnisse werden diskutiert.

Die der Problemstellung zugrundeliegende Annahme ist, dass menschliche Bewegungen bezüglicher einer unbekannten Kostenfunktion (näherungsweise) optimal sind. Die Kombination einer Kostenfunktion mit der Dynamik des Menschen führt auf ein Optimalsteuerungsproblem und die zugehörige Lösung kann dann mit aufgezeichneten Daten menschlicher Bewegungen verglichen werden. Ziel der Inversion ist es, diejenige Kostenfunktion innerhalb einer gegebenen (parametrisierten) Menge zu bestimmen, die einen minimalen Abstand zwischen den Daten und der Lösung des zugehörigen Optimalsteuerungsproblems liefert. Die in dieser Arbeit verwendete Lösungmethode basiert auf einem Kollokationsansatz zum Diskretisieren des Optimalsteuerungsproblems und einer Reformulierung des resultierenden Bilevel-Problems mittels Optimalitätsbedingungen. Zur Lösung der daraus folgenden nichtlinearen Optimierungsprobleme wird dann auf eine Innere-Punkte-Methode zurückgegriffen.

Ein weiterer wesentlicher Teil der Arbeit ist die Modellierung verschiedener menschlicher Bewegungen und die Diskussion numerischer Lösungen der entsprechenden inversen Optimalsteuerungsprobleme. Die dargestellten Beispiele sind menschliche Armbewegungen in zwei und drei Dimensionen, vom Menschen gesteuerte Spurwechsel von Autos und menschliche Navigationsprobleme, bei denen die allgemeine Pfadplanung beim menschlichen Gehen analysiert wird.

Contents

Notations v

Acronyms ix

Preface 1

1 Introduction 3
- 1.1 Outline of the Work . . . 4

2 Bilevel Optimization and MPECs 11
- 2.1 Lower and Upper Level Programs . . . 12
- 2.2 Existence of a Global Optimistic Solution . . . 13
- 2.3 Solution Strategies for Bilevel Programs . . . 19
- 2.4 Optimality Conditions for MPECs . . . 20
- 2.5 Regularization Strategy for MPECs . . . 25
  - 2.5.1 Relaxation Scheme . . . 25
  - 2.5.2 Further Regularization Approaches . . . 27
  - 2.5.3 Extension for Interior-Point Methods . . . 29

3 Optimal Control 33
- 3.1 Indirect Methods . . . 35
  - 3.1.1 Inequality Constraints . . . 37
  - 3.1.2 Numerical Methods for Boundary Value Problems . . . 39
- 3.2 Direct Methods . . . 41
  - 3.2.1 Collocation Strategies . . . 42
- 3.3 Minimum Jerk Example . . . 49
- 3.4 Existence Results . . . 53

4 Inverse Optimal Control 59
- 4.1 Inverse Optimal Control Problem . . . 59
- 4.2 State of the Art . . . 61
  - 4.2.1 Inverse Optimal Control of Human Car-Steering . . . 62

4.2.2 Inverse Optimal Control of a Neuro-Musculoskeletal System 63
4.2.3 Bilevel Optimal Control of a Rack Feeder 63
4.2.4 Inverse Optimal Control of Human Navigation 65
4.2.5 Inverse Optimal Control of Human Arm Motions 66
4.2.6 Bilevel Optimal Control of Flight Trajectory Optimization 66
4.2.7 Inverse Optimal Control of Human Leg Motions 67
4.3 ULP Distance Measures 67
4.4 Structure of Discretized Inverse Optimal Control Problem 69
4.4.1 Global Optimistic Solution 70
4.4.2 CQ for the Transformed (One-Level) Problem 72

5 Numerical Methods **77**
5.1 Numerical Strategies for Nonlinear Optimization 77
5.1.1 Interior-Point Methods 79
5.1.2 Filter Techniques 82
5.1.3 Optimization Method `IPOPT` 84
5.2 Optimization Method `coreIOC` for Inverse Optimal Control 88
5.2.1 Adaptive Time Discretization 88
5.2.2 Scaling 90
5.2.3 Goal Attainment 93
5.2.4 Reconstruction Tests 95

6 Human Arm Movements **97**
6.1 Introduction to Human Arm Motions 97
6.2 Rigid Body Models 99
6.2.1 Denavit-Hartenberg notation 99
6.2.2 Equations of Motion 103
6.3 Muscle Models 108
6.3.1 A Linear Muscle Model 108
6.3.2 Muscle Model of Stroeve 109
6.4 Cost Functions 112
6.4.1 Smoothness 112
6.4.2 Accuracy 115
6.4.3 Energy, Time and Others 116
6.4.4 Cost Combinations 117
6.5 Arm Models 118
6.5.1 Planar Arm Model 120
6.5.2 Three-dimensional Arm Model 122
6.6 Numerical Results for Inverse Optimal Control 123
6.6.1 Reconstruction of Planar Arm Motions 123

6.6.2 Inversion of Human Planar Arm Motions . . . 125
6.6.3 Reconstruction of Three-Dimensional Arm Motions . . . 127
6.7 Transfer to Robotic Systems . . . 131
6.7.1 Motion Primitives . . . 131
6.7.2 Imitation Learning . . . 132
6.7.3 Human-like Optimal Control . . . 134

7 Human Car Driving **137**
7.1 State of the Art . . . 137
7.2 Experimental Vehicle . . . 138
7.3 Dynamical Car Model . . . 138
7.3.1 Nonlinear Single-Track Model . . . 138
7.3.2 Linear Single-Track Model . . . 142
7.4 Car-Steering Bilevel Problem . . . 144
7.4.1 LLP Formulation . . . 145
7.4.2 ULP Formulation . . . 146
7.4.3 Numerical Results . . . 147

8 Human Locomotion **153**
8.1 Locomotion Dynamics . . . 153
8.2 Cost Functions . . . 154
8.2.1 Modeling the Interferer . . . 155
8.3 Human Experiments . . . 156
8.3.1 Distance Measures . . . 156
8.4 Model Predictive Control . . . 157
8.5 Optimization Results . . . 158
8.5.1 Reconstruction of Multiple Locomotions . . . 158
8.5.2 Inversion of U-Turn Motions . . . 160
8.5.3 Reconstruction of MPC Navigation . . . 161

Summary **165**

A Nonlinear Optimization **167**
A.1 First-Order Optimality Conditions . . . 168
A.2 Second-Order Optimality Conditions . . . 171

B State of the Art on Human Arm Motions **173**
B State of the Art on Human Arm Motions . . . 173
B.1 Research Goals of Disciplines . . . 173
B.2 Motion Characteristics . . . 174
B.2.1 Fitts' Law . . . 175

B.2.2 Donders' Law and Listing's Law 176
B.2.3 Two-Thirds Power Law 176
B.3 Motor Control 176
B.4 Motion Generation 178
B.4.1 Redundancy 178
B.4.2 Motion Control 178
B.5 Open-Loop and Closed-Loop Control 180
B.6 Adaptation 182
B.7 Discussion of Implications 184

Bibliography **185**

Notations

Scalars:

$(x)_+$	(scalar)	the value $\max\{0, x\}$ for a scalar x
$(x)_-$	(scalar)	the value $\max\{0, -x\}$ for a scalar x
$\|x\|$	(scalar)	the absolute value of the scalar x
x^n	(scalar)	the n-th power of the scalar x
t	(scalar)	the time variable (the independent variable)

Vectors:

$x \in \mathbb{R}^n$	(vector)	column vector in $\mathbb{R}^n$
x^*	(vector)	the optimal vector for an optimization problem
x_i	(scalar)	i-th component of the vector x
$x_{(i)} \in \mathbb{R}^m$	(vector)	i-th m-dimensional subvector of the vector $x \in \mathbb{R}^n$
$^{<j>}x$	(vector)	vector x stated with respect to the j-th coordinate system
$(x)_+$	(vector)	$((x_1)_+, \ldots, (x_n)_+)^T$ for a vector $x \in \mathbb{R}^n$
$(x)_-$	(vector)	$((x_1)_-, \ldots, (x_n)_-)^T$ for a vector $x \in \mathbb{R}^n$
x^T	(row-vector)	transposed vector
$x_{\mathbb{X}}$	(vector)	vector consisting of the components x_i with $i \in \mathbb{X} \subseteq \mathbb{N}$
$\mathbb{1}$	(vector)	vector of ones (i.e., $\mathbb{1}_i = 1 \ \ \forall i$)
$e^{(i)}$	(vector)	i-th unit vector (i.e., $e^{(i)}_j = 0 \ \ \forall j \neq i$ and $e^{(i)}_i = 1$)
$\|\|x\|\|$	(scalar)	Eucledian norm of the vector x
$\|\|x\|\|_n$	(scalar)	n-norm of the vector x
$\mathcal{V}(x)$	(matrix)	diagonal matrix with diagonal x
$\left(x^T, y^T\right)^T \in \mathbb{R}^{n+m}$	(vector)	concatenation vector of $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$

$\min\{x, y\}$ $\max\{x, y\}$	(vector) (vector)	the vectors with the (componentwise) minimal or maximal values of the vectors x and y
$x \geq y$		componentwise ordering (i.e., $x_i \geq y_i \ \ \forall i$)
$x > y$		componentwise ordering (i.e., $x_i > y_i \ \ \forall i$)
$x \ \perp \ y$		orthogonality condition (i.e., $\min\{x, y\} = 0$)

Matrices:

$x_{i,j}$	(scalar)	(i, j)-element of the matrix x
$x_{\cdot,j}$	(vector)	j-th column of the matrix x
${}^{<j>}x_{<i>}$	(matrix)	transformation matrix from the i-th to the j-th coordinate system
$\mathcal{U}$	(matrix)	identity matrix

Sets:

$\mathbb{N}$	(set)	the set of the natural numbers
$\mathbb{R}$	(set)	the set of the real numbers
$\mathbb{R}_{>0}$	(set)	the set of the strictly positive real numbers
$\mathbb{R}_{\geq 0}$	(set)	the set of the non-negative real numbers
$\mathbb{R}^n$	(set)	the n-dimensional space of real numbers
$\mathbb{R}^n_{>0}$	(set)	the strictly positive orthant of $\mathbb{R}^n$
$\mathbb{R}^n_{\geq 0}$	(set)	the non-negative orthant of $\mathbb{R}^n$
$\mathbb{U}_\varepsilon(x)$	(set)	the ball with radius ε about x
(a, b)	(set)	the open interval between a and b in $\mathbb{R}$
$[a, b]$	(set)	the closed interval between a and b in $\mathbb{R}$
$\mathbf{\Delta}$	(set)	the partition of a real interval, i.e., strictly ordered elements
$\lvert\mathbb{X}\rvert$	(scalar)	the cardinality of the set $\mathbb{X}$
$\mathbb{X} \subseteq \mathbb{Y}$	(set)	the set $\mathbb{X}$ is a subset of the set $\mathbb{Y}$
$\mathbb{X} \subset \mathbb{Y}$	(set)	the set $\mathbb{X}$ is a proper subset of the set $\mathbb{Y}$
$\mathbb{X} \cap \mathbb{Y}$	(set)	the intersection of the sets $\mathbb{X}$ and $\mathbb{Y}$
$\mathbb{X} \cup \mathbb{Y}$	(set)	the union of the sets $\mathbb{X}$ and $\mathbb{Y}$
$\mathbb{X} \backslash \mathbb{Y}$	(set)	the complement of $\mathbb{Y}$ in $\mathbb{X}$
$(\mathbb{X} \cap \mathbb{Y})(x)$	(set)	the intersection of the sets $\mathbb{X}(x)$ and $\mathbb{Y}(x)$
$(\mathbb{X} \cup \mathbb{Y})(x)$	(set)	the union of the sets $\mathbb{X}$ and $\mathbb{Y}(x)$
$(\mathbb{X} \backslash \mathbb{Y})(x)$	(set)	the complement of $\mathbb{Y}(x)$ in $\mathbb{X}(x)$
$\widehat{\mathbb{P}}(\mathbb{X})$	(set)	the power set of a set $\mathbb{X}$

Sequences:

$x^{(i)}$	the i-th element of a sequence
$(x^{(i)})$	a sequence with elements $x^{(i)}$
$(x^{(i)})_{\mathbb{X}}$	the subsequence of $(x^{(i)})$ corresponding to the index set $\mathbb{X} \subseteq \mathbb{N}$
$x^{(i)} \to \overline{x}$	a convergent sequence with limit $\overline{x}$

Functions:

$Df(x)$	the Jacobian matrix of the function f at the point x (e.g., $Df(x) \in \mathbb{R}^{n \times m}$ for a function $f : \mathbb{R}^n \to \mathbb{R}^m$)
$D_x f(x, y)$	the Jacobian matrix of the function f with respect to x at the point given by x and y
$\nabla f(x)$	$(Df(x))^T$, gradient of the function f at the point x
$\nabla_x f(x, y)$	$(D_x f(x, y))^T$, gradient of the function f with respect to x at the point given by x and y
$\nabla^2 f(x)$	the Hessian matrix of the function f at the point x
$\nabla^2_{xx} f(x, y)$	the Hessian matrix of the function f with respect to x at the point given by x and y
$f'(x)$	the first derivative of the function f given the scalar x
$f''(x)$	the second derivative of the function f given the scalar x

Function Spaces:

$\mathcal{C}$	the function space of the continuous functions
$\mathcal{C}_p$	the function space of the piecewise continuous functions
$\mathcal{C}^1$	the function space of the continuously differentiable functions
$\mathcal{C}^1_p$	the function space of the piecewise continuously differentiable functions
$\mathcal{C}^1_c$	the function space of the piecewise continuously differentiable functions that are continuous
$\mathcal{C}^2$	the function space of the twice continuously differentiable functions
$\mathbb{P}_i$	the space of the polynomials of degree $i - 1$
$\mathbb{P}_{i,\Delta}$	the space of the splines of degree $i - 1$, i.e., piecewise combination of polynomials in $\mathbb{P}_i$
$\mathbb{P}^n_i$	the space of the vector-valued functions where each of the n components is a polynomial of degree $i - 1$
$\mathbb{P}^n_{i,\Delta}$	the space of the vector-valued functions build of splines of degree $i - 1$ in each of the n image dimensions

Acronyms

ACQ	(p. 169)	Abadie constraint qualification
BL-C	(p. 15)	bilevel compactness assumption
BL-MFCQ	(p. 15)	bilevel Mangasarian-Fromowitz constraint qualification
CQ	(p. 169)	constraint qualification
DAE		differential algebraic equation
DH	(p. 99)	Denavit-Hartenberg (notation)
GCQ	(p. 169)	Guignard constraint qualification
GULP	(p. 12)	general upper level problem
KKT	(p. 169)	Karush-Kuhn-Tucker (conditions)
LICQ	(p. 171)	linear independence constraint qualification
LLP	(p. 12)	lower level program
MFCQ	(p. 170)	Mangasarian-Fromowitz constraint qualification
MPCC	(p. 11)	mathematical problem with complementarity constraints
MPEC	(p. 21)	mathematical problem with equilibrium constraints
MPEC-LICQ	(p. 23)	linear independence constraint qualification for an MPEC
MPEC-SOSC	(p. 24)	second-order sufficient condition for an MPEC
ODE		ordinary differential equation
PLICQ	(p. 170)	positive linear independence constraint qualification
RNLP	(p. 24)	relaxed nonlinear optimization problem
RNLP-SOSC	(p. 24)	second-order sufficient for relaxed nonlinear problem
SOSC	(p. 172)	second-order sufficient condition
SSOSC	(p. 24)	strong second-order sufficient condition
SQP	(p. 78)	sequential quadratic programming
ULP	(p. 13)	upper level program
a.e.		almost everywhere

Preface

Stereotypic human motions are observed for a broad range of tasks in daily life. In consequence, the identification of the basic principles governing these motions is the goal of a significant amount of research. The characteristics observed in various experimental settings seem to be the outcome of an efficient learning scheme and a fast adaptation to new settings is reported. One central idea discussed in literature is that, after an initial learning phase, the stereotypic human motions are approximately optimal with respect to an unknown criterion and thus several of such cost functions have been proposed.

If the goal is to simulate human motions being optimal with respect to one of these cost functions, a model of the dynamics of the human system is needed. This means that a differential equation has to relate the values that can be controlled, the *controls*, to the temporal change of the values representing the current state of system, the *states*; we focus here on dynamics that can be posed in form of a system of ordinary differential equations. Combining one hypothesized cost function, a model of the dynamics of the human and conditions stating the motion task with respect to the state values at the start and end of the motion, a mathematical optimal control problem is obtained. Assuming that the model captures the dynamics sufficiently well and that the correct cost function is known, the solution of the optimal control problem should correspond to the observed human motion. Even if this is possible, one has to note that, from the biological perspective, such an optimal control problem should not be identified with the real biological system, but is a modeling tool within a range of validity which has to be specified.

The research presented in this work has been carried out within the cluster of excellence `CoTeSys` at the Technische Universität München where scientists of various disciplines ranging from psychology over sport sciences to electrical engineering work on the topic of *cognitive technical systems*. The ultimate goal is that humans and robots cooperate intuitively in an environment of daily life, e.g., jointly setting a table in a kitchen. Since an environment of daily life is non-static and unforeseen changes of mind of cooperating humans occur, an adaptable control strategy has to be used for the robot control. Consequently, a reasonably precise model of standard human motions is needed to anticipate movements. Additionally, knowing the underlying principles of human motions, one could control an humanoid robot accordingly and thereby increase the anticipation of the robot's motions by cooperating humans.

Naturally, the question arises which cost function does a human optimize when doing a certain task in daily life? We address this problem by stating an optimization problem: Find the cost function for the optimal control problem that minimizes the distance between the corresponding optimal state and the recorded human data. We assume that the set of feasible cost functions for this data matching problem is a continuously parameterized family of cost

functions, i.e., the goal is to determine a vector of parameters such that the corresponding cost function yields a state minimizing the distance measure. In consequence, two optimization problems are obtained where one is part of the constraints of the other; such a combination is called a *bilevel problem*. Because one of the two problems is an optimal control problem, our problems fall into the class of *bilevel optimal control problems*; in this work we use the term *inverse optimal control problem* since the data fitting problem is a standard problem of inverse optimization.

The presented solution strategy is based on discretizing the optimal control problem by a suitable collocation approach and then transform the bilevel problem into a standard (one-level) optimization problem via the first-order necessary optimality conditions. The resulting problem is a mathematical program with complementarity constraints (MPEC) which needs further modifications in order to be solved with a standard interior-point algorithm. Consequently, theory on nonlinear optimization, bilevel programs, MPECs and optimal control has to be reviewed in order to state our solution strategy in full detail. Both analysis of the problem structure and numerical experiences with our optimization method `coreIOC` are discussed.

Three application examples are presented in this work: the central one is the problem of human arm motions. For this example we discuss in detail the state of the art in the related disciplines to clearly define where our approach fits into the main research lines and to highlight the limits of our open-loop optimal control modeling of human motions. All elements needed to model the dynamics of the human arm as a combination of rigid bodies and muscles are introduced and, in addition to a standard planar arm model, a three-dimensional arm model is derived. Two scenarios to use the result of the inverse optimal control problem for robot control are discussed.

This is followed by the second application example where characteristics of lane changes on a highway are analyzed. Optimization results of the inverse optimal control problem for two dynamics, a linear and a nonlinear single-track model of the car's dynamics, are discussed. Cost functions describing human behavior can be used in this scenario to control an autonomous car and thereby to raise the acceptance by both the passengers and the other traffic participants.

The third class of application examples considers the human locomotion problem where the task is to walk from a start to an end position while avoiding a collision with a crossing person. The goal is to describe the overall motion by a model considering one rigid body, neglecting the details of the dynamics of individual steps. A characteristic element observed in human locomotion is the adaptation process during the motion to account for changes in the environment, i.e., the changes in the position of the interfering person. In consequence, we consider a model predictive control approach where the overall motion is split into a sequence of submotions. Thus a system of optimal control problems has to be considered in the inverse optimal control approach and numerical results are presented showing that our solution strategy can handle this problem class, too.

A more detailed introduction stating the central aspects of each chapter and relation between them can be found in the next chapter.

Introduction

Chapter 1

The problems of inverse optimal control, which form a special class of bilevel optimal control problems, are the central topic of this work. Each bilevel problem is a combination of a *lower level program (LLP)* and an *upper level program (ULP)* where the LLP is part of the upper level constraints. In case of a bilevel optimal control problem at least one of the two problems is an optimal control problem.

In [59] such problems are first introduced in the context of Stackelberg games and related publications deal, for example, with systems with feedback (e.g. [10, 228]) or with dynamic games (e.g. [22, 39]). Recently, two works on bilevel optimal control problem have been presented. First, the problem of a rack feeder is discussed in [177, 178] where the task is to optimally control a ceiling-attached rack feeder in a high rack. Modeling the dynamics as a mathematical pendulum and considering, for example, the minimization of the controls or the oscillations of the load handling device, bilevel optimal control problems are obtained which range from parametric optimization in the upper level to a combination of several optimal control problems. The solution strategy used to solve these bilevel optimal control problems is a hybrid method combining the indirect approach for optimal control problems with the direct one for the overall problem (cf. section 3). Second, the problem of optimizing the track of an air race is addressed in [95]. Combining a complex model of the plane dynamics and assuming time-optimal control, a bilevel optimal control problem is solved where in the ULP the gate positions of the track are optimized in order to maximize safety- and fairness-related cost functions. The optimal control problem is solved by a direct method and sensitivity analysis of the optimal solution is used to solve the upper level problem.

If the ULP is a data fitting problem and the LLP the optimal control problem, an inverse optimal control problem results. In addition to our publications [4, 5, 6, 7, 187] such problems are discussed in [26, 34, 50, 217]. First, the problem of determining the best combination of three given cost functions for double lane changes of a car is discussed in [50] where a simplified single-track model is used for the dynamics of the car. The sensitivity information of the LLP solution is determined by solving a linear-quadratic optimal control problem and this information is used to solve the bilevel problem with a variant of the Levenberg-Marquardt algorithm. Second, in [34] a solution strategy is explained which is similar to the one presented in this work. However, numerical results are not presented for an optimal control problem in the LLP, but only standard bilevel problems are solved. Note that neither details on discretizing the optimal control problem nor on solving the resulting nonlinear problem are given. Third, the inverse optimal control problem of human locomotion is addressed in [217] and the goal is to use the optimal cost function to control a humanoid robot. The human walking problem is considered on the level of trajectories of position and orientation, but

individual steps or rigid body and muscle dynamics are not of interest. The resulting optimal control problem in the lower level is solved by a multiple shooting method and in the upper level a derivative-free optimization approach is used. This approach is also used in [26] to analyze planar arm motions. For further details on these related works in bilevel and inverse optimal control see section 4.2.

The solution strategy realized in this work reformulates the inverse optimal control problem as a nonlinear optimization problem and, consequently, differs from the approaches named above which use two separate optimization methods for the two problems. In our case a parameterized family of feasible cost functions for the optimal control problem is given in form of convex combinations of basic cost functions (see for example section 6.4.4). The inversion problem in the upper level is to find the optimal vector of parameters corresponding to one lower level cost function such that the corresponding optimal state has minimal distance to the given data. The issue of finding a distance measure suitable for each problem is addressed in section 4.3.

The first step in the solution process of the inverse optimal control problem is to address the optimal control problem in the lower level on its own. Considering the special structure of the dynamics of the examples, it is shown in section 3.4 by using the existence theorem of Filippov that under certain assumptions an absolute minimum of the optimal control problem exists. To numerically solve the problem, we use a collocation technique to discretize the problem to obtain a nonlinear optimization problem. Following the line of [330], it is shown in section 3.2.1 that the optimality conditions for the discretized optimal control problem converge to the (continuous) optimality conditions of optimal control theory if the discretization step size goes to zero. Consequently, the polynomial basis of the collocation approach has to be chosen in accordance with an adaptive time discretization (section 5.2.1) to obtain a good approximation of the continuous solution using a small number of discretization points.

Using the discretized optimal control problem in the lower level, a standard bilevel optimal control problem results. A standard technique to solve such a problem is to replace the lower level by its first-order necessary optimality conditions; however, the resulting problem is in general not equivalent to the original inverse optimal control problem. In consequence of adding the KKT-conditions to the constraints of the upper level, a problem with complementarity conditions (MPEC) is obtained if inequality conditions are part of the problem formulation of the optimal control problem. Therefore, if necessary, we adapt the relaxation approach of [324] to solve the MPEC by a sequence of nonlinear optimization problems without complementarity constraints (cf. chapter 2).

In order to evaluate our solution strategy we discuss in section 5.2.4 a reconstruction framework. The idea is to generate artificial data by solving the optimal control problem for a known cost function

1.1 Outline of the Work

In the following the structure of this work is described, i.e., the central aspects of each chapter are summarized and their implication on the overall solution strategy is discussed. Additionally, the dependence of the individual chapters is highlighted.

In chapter 2 the structure of (nonlinear) bilevel programs and the related MPEC problems are discussed. Since both problem types are closely related to standard nonlinear optimization problems, we refer to the appendix A for a short review of relevant theory on nonlinear

optimization. Especially, the concept of constraint qualifications (CQs) with respect to the first-order and second-order optimality conditions are discussed. Necessary extensions of the CQ concept to bilevel programs and MPECs are covered in chapter 2 which is structured in the following way:

First, bilevel programs are introduced as the combination of two nonlinear optimization problems where one is part of the constraints of the other (see section 2.1); consequently, the one is termed the lower level program (LLP) that is constraining the other one, the so-called upper level program (ULP). The definition of the bilevel problem is closely related to the concept of optimistic and pessimistic solutions which describe the process of determining the solution in LLP if the solution set has more than one element. We constrain ourselves in this work to the optimistic case, where in case of multiple LLP solutions the one is chosen that is best to minimize the ULP.

Furthermore, a proof of the existence of a global optimistic solution under suitable conditions following the outline given by [74] is stated in detail in section 2.2. The semicontinuity concept for the set-valued mapping is a central element in the proof. This theorem is later used to prove the existence of a global optimistic solution for the discretized inverse optimal control problem under certain assumptions. The state of the art on solution strategies for bilevel programs is discussed in section 2.3. One of the approaches is based on the optimal value function of the LLP and this concept is used in [352] to deduce (non-smooth) optimality conditions for bilevel optimal control problem. Another approach to solve the bilevel program is to replace the LLP by its first-order necessary optimality condition; this approach is used in this work to solve the inverse optimal control problem.

If the bilevel problem is transformed by this approach into a (one-level) optimization problem, a mathematical program with complementarity constraints results. The term *MPEC* is the abbreviation for *mathematical programs with equilibrium constraints* which is a special class of nonlinear optimization problems characterized by the structure of the constraints. Originally, the equilibrium constraints are parametric variational inequalities describing the equilibrium of a system, e.g., a Nash equilibrium in game theory [202]. If suitable assumptions are fulfilled, the variational inequalities can be replaced by a system of complementarity conditions; to emphasize this special type of MPECs, the term *MPCC* is used in literature. Here we will only consider MPECs of the MPCC-type, but still call them MPECs. Since complementarity conditions are part of the necessary optimality conditions for a general problem of nonlinear optimization with inequality constraints (see appendix A), the MPECs analyzed in this work result from bilevel programs (cf. chapter 2) where the lower level program is replaced by its first order necessary optimality conditions. The reason for treating MPECs separately from the standard nonlinear optimization problems of appendix A is that most standard constraint qualifications do not hold for MPECs [106, 245, 276]. Thus, further theory concerning optima of MPECs is discussed in this chapter and adaptations of numerical methods for standard nonlinear optimization problems are described. Naturally, several theoretical results and numerical approaches are presented in literature, but we restrict ourselves to the most common principles for optimality conditions of MPECs. We focus in section 2.5.1 on the regularization scheme of [324] which is used in this work to solve the reformulated one-level problem, but other regularization approaches are summarized in section 2.5.2. Since the MPECs are to be solved with an interior-point method, further considerations are needed to assure that at all instances a non-empty proper interior of the feasible region exists. In this line we review the two-sided relaxation variant of [324] in section 2.5.3. For more details on MPECs see, for example, [202, 242, 278, 324, 353].

In chapter 3 problems of optimal control are discussed which, in the context of our inverse optimal control problem, correspond to the lower level programs. Two general approaches exist to solve this problem of minimizing a cost function subject to the state dynamics and the equality and inequality constraints on both the states and the controls: First, the *indirect approach of optimal control* (cf. section 3.1) derives optimality conditions in function spaces, yielding a multi-point boundary value problem. The two most common numerical strategies for solving such boundary value problems are the collocation approach and the multiple-shooting technique (see section 3.1.2). Consequently, the idea of the direct approach is to optimize and then to discretize. Second, the *direct methods of optimal control* (cf. section 3.2) follow the opposite strategy: In the first step, the optimal control problem is discretized and then methods of nonlinear optimization are used (see appendix A) to solve it. The collocation approach of [330] discussed in section 3.2.1 is used in this work to discretize the optimal control problem in the LLP. Convergence results (adapted versions of the ones proposed in [330]) show that the KKT-conditions of the discretized optimal control problem converge in limit to the optimality conditions of the indirect approach of optimal control. In section 3.3 the one-dimensional minimum jerk problem is used as an numerical example to show convergence of states, controls and adjoint variables and to discuss different convergence rates of different collocation types. If the classical setting of piecewise continuously differentiable states in the optimal control problem is weakened to allow for absolutely continuous functions, the existence of an solution can be proven under certain assumptions by Filippovs theorem (cf. section 3.4). We show that a weaker set of assumptions which are sufficient to proof the existence theorem of Filippov are met by the optimal control problems considered in the application examples (chapters 6 to 8).

Inverse optimal control problems are discussed in detail in chapter 4. Following the problem formulation (section 4.1), the state of the art in bilevel optimal control and, especially, inverse optimal control is presented in section 4.2. The measurement of the distance between the LLP state and the data highly influences the solution of the inverse optimal control problem. Therefore, this distance measure has to be selected with care and has to be suitable for the goal of the inverse problem. Consequently, two different distance measures are introduced in section 4.3: The first one compares points of equal relative path length and the other measures the distance between points at equal time instances; both measures are used in different application scenarios (see chapters 6 to 8). The last part of this chapter analyzes the problem structure of the inverse optimal control problem. Under suitable assumptions to the discretization strategy, it is shown (cf. section 4.4) that the discretized inverse optimal control problem fulfills the requirements of the theorem presented in the chapter on bilevel problems (cf. chapter 2) which guarantees the existence of a global optimistic solution. Furthermore, a constraint qualification for the transformed one-level optimization problem is discussed in section 4.4.2 in order to use an interior-point algorithm of nonlinear optimization (cf. section 5.1).

Using the theoretical results of the chapters 2 to 4, some implementation details of the numerical optimization method `coreIOC` are discussed in chapter 5.2. The basic idea is to solve, if necessary, a sequence of relaxed problems by the interior-point method `IPOPT`. Alongside the update strategy for the relaxation parameters, the time discretization of the discretized optimal control problem has to be updated (cf. section 5.2.1) in order to approximate the continuous optimal control problem close enough. Additionally, the issue of scaling the optimization parameters and functions is addressed in section 5.2.2. Numerical tests showed that modifying the one-level problem by adding an additional constraint increases solver

performance for certain problems, e.g., see the example in section 6.6.3. In section 5.2.4 a framework to evaluate the performance of the inverse optimal control approach is discussed. The data is computed by solving the LLP for a given cost function and the goal is to analyze the differences between the true cost function and the optimization result of the inverse optimal control problem using a different starting value and adding noise to the simulated data values. The results discussed in the chapters of the application examples show that the presented optimization approach is well-suited to solve the given problems of inverse optimal control.

In the second part of this work (chapters 6 - 8) three application examples are discussed. All examples have in common that they describe tasks where humans select characteristic controls leading to stereotypic motions. However, the problems differ in the complexity of dynamics, constraints and cost functions; consequently, different aspects of modeling problems of daily life and of solving the corresponding problem are addressed. Application scenarios for the solutions of the inverse optimal control problem are discussed for all examples. In the following we want to briefly introduce each of the examples and discuss the structure of the according chapters.

In chapter 6 the inverse optimal control problems of human arm motions are addressed; the idea to use a cost function optimized in human movements to control a robotic arm was the starting point for the research presented in this thesis. Consequently, the state of the art of various related disciplines is discussed in section 6.1 and in more detail in appendix B to show where the inverse optimal control strategy fits into the general lines of research and to discuss the limits of our approach from an engineering or biological perspective. For example, psychologists and biologists use experiments to deduce the underlying (biological) structures and principles of human motion with the goal to understand the mechanism determining the human actions, in contrast to the approach used in this work where the focus is on finding an (optimal control) model that describes the observations. In consequence, one has to distinguish between a biologically plausible principle and a mathematical model, but nevertheless such a model might be valuable even from the biological perspective if it makes correct predictions in different settings (cf. section B.1).

Results of various experimental studies are presented in literature and several basic properties of human arm movements are discussed (compare section B.2); one of the most prominent relations is Fitts' law [96] which relates motion time to target accuracy. Central research questions are how human arm motions are actually generated by the human body (cf. section B.3) and how the motions are planned (cf. section B.4). In this work we use the hypothesis that human motions are generated by controlling the forces of the muscles. Two further aspects related to human arm movements are closed-loop control (see section B.5) and adaptation (see section B.6). Several experiments show that humans use feedback while doing arm motions and learning processes are observed which improve the performance for a given task. Our basic assumption is that the arm motions analyzed in this work are controlled in an open-loop manner and that the human movements are already (approximately) optimal as a results of a finished learning process; since only standard motions of everyday life are analyzed, such assumptions seem to be justified.

Since optimal control of arm movements asks for a dynamical model capturing the main features of the human arm dynamics, a combination of dynamical models for the bones and several lumped muscles is used. In section 6.2 the Denavit-Hartenberg notation [143] for chains of rigid bodies, which are used to model the human bones, is discussed and the corresponding equations of motions are motivated by the Newton-Euler equations resulting in

a recursive framework (see section 6.2.2). To model the dynamics of human muscles is a topic of current research (cf. section 6.3) and various models of different complexities capturing different levels of observed effects have been presented. Two such models are specified in section 6.3: First, a mass-damper model resulting in a linear ODE and second, a nonlinear muscle model introduced in [291, 292].

The basic cost functions needed to formulate the inverse optimal control problem are discussed in section 6.4. Several of these basic cost functions are proposed in literature and they yield arm motions reasonably similar to observed human ones in certain settings. However, each of the models has certain limitations and a cost function describing the broad range of human arm motions is not known. Consequently, we consider the convex combinations of these basic cost functions and try to determine the weighting parameters of the convex combinations that yield the closest fit between data and LLP result. Combining the rigid-body dynamics, the muscle dynamics and the LLP cost functions, two dynamical arm models with different numbers of degrees of freedom are derived in section 6.5: A planar arm model and a three-dimensional arm model. In both cases, human data of experiments suitable for the model is available and numerical results of the inverse optimal control problem are presented in section 6.6.

Finally, two technical scenarios are sketched where such an optimal LLP cost function could be used. On the one hand, the optimal control model can be used to predict motions of the human arm and consequently, the effects of the time-delay between measurements and actions could be reduced in a telepresence setup. On the other hand, the information can be used to control a humanoid robot according to the cost functions of human demonstrations. The goal of such a transfer is to raise acceptance of a robot in a human environment and to increase the anticipation level, allowing a closer human-robot interaction (for more details see section 6.7).

The second application example for our inverse optimal control approach is the problem of steering a car on a highway (see chapter 7) which is introduced in [50]. Focusing on lane changes the general goal is to get the optimal cost function for various driving situations and tasks. The knowledge of the cost functions used by humans can then be deployed to control an autonomous vehicle in a human-like fashion, improving the acceptance by both the other traffic participants and the car occupants. Two models of the car dynamics are discussed in section 7.3. A simple linear one which is frequently used in literature and a more recent nonlinear single-track model in accordance with [124] which allows to simulate many details of car driving. The formulation of the inverse optimal control problem is given in section 7.4 and numerical results for data examples of human-steered lane changes can be found in section 7.4.3.

In chapter 8 the third application example is discussed, where the task is to walk from a given start position to a designated goal position; inverse optimal control for this type of problem is introduced in [217]. The central idea is to maintain a macroscopic perspective and model the human as a mass point with an orientation. If only velocities in forward direction are allowed, this leads to the so-called unicycle model (see section 8.1). Naturally, one can model more details of the human walking process if the application scenario demands more elaborate information about the plant; thus, a brief overview of other models is given. The scenario we are focusing on is the general path planning problem which, for example, has to be solved in mobile robotic systems. Since our interest lies in locomotion in the context of daily life where obstacles in the workspace are the standard case, human motions with crossing persons are discussed. Therefore, new models are introduced (cf. section 8.2) and in consequence a set

of nonlinear parameters has to be optimized in the upper level problem in addition to the weights of the convex combination of lower level cost functions. In section 8.4 the standard optimal control problem is extended to a model predictive control setup to account for the adaptation done by humans to react to positional changes of obstacles. In model predictive control the solution of an optimal control problem is only realized for a given time horizon and then a new optimal control problem is solved with the current position as the starting point. Consequently, the optimal control problem in the lower level is replaced by a system of optimal control problems and the task of finding one combination of basic cost function is extended to the case of finding one combination for the motion parts where no obstacle has to be considered and one for the collision avoidance. This separation mimics the human behavior observed in experiments where the participants react to a crossing interferer only at the last possible instance. Numerical results for the inverse optimal control problem of human locomotion are presented in section 8.5.

The last chapter states a short summary combined with a critical review of the presented results. Finally, an outlook on possible subsequent research completes this work.

Bilevel Optimization and MPECs

Chapter 2

The combination of two optimization problems where one is part of the constraints of the other is called a *bilevel program*. The numerical solution strategy developed in this work is used to solve inverse optimal control problems representing a special class of bilevel programs: On the one hand, a (discretized) optimal control problem has to be solved and on the other hand, a parameter combination minimizing a distance measure with respect to given data is sought.

Therefore, the central concepts of the bilevel programming theory are introduced in this chapter; the presentation follows the line of Dempe [74]. In section 2.1 the general structure of bilevel programs is introduced; followed in section 2.2 by the discussion of the requirements for a global optimal solution of the bilevel program. In the subsequent section 2.3 different solution strategies presented in literature are discussed including the approach used in this work. Using first-order optimality conditions, this approach transforms the bilevel problem into a mathematical program with complementarity conditions; consequently, problems of this structure are discussed in the second part of this chapter addressing *MPECs*.

The term MPEC is the abbreviation for *mathematical programs with equilibrium constraints* which is a special class of nonlinear optimization problems characterized by the structure of the constraints. Originally, the equilibrium constraints are parametric variational inequalities describing the equilibrium of a system, e.g., a Nash equilibrium in game theory [202]. If suitable assumptions are fulfilled, the variational inequalities can be replaced by a system of complementarity conditions; to emphasize this special type of MPECs, the term *MPCC* is occasionally used in literature. Here we will only consider MPECs of the MPCC-type, but still call them MPECs.

The reason for treating MPECs separately from standard nonlinear optimization problems (appendix A) is that most standard constraint qualifications do not hold for MPECs [106, 245, 276]. Thus, further theory concerning optima of MPECs is discussed in section 2.4 and adaptations of numerical methods for standard nonlinear optimization problems are described in 2.5. Naturally, several theoretical results and numerical approaches are presented in literature, but we restrict ourselves to the most common principles for optimality conditions of MPECs and focus on the relaxation approach of [324]. For more details on MPECs see, for example, [202, 242, 278, 324, 353].

2.1 Lower and Upper Level Programs

Problems where more than one optimization take place are introduced by von Stackelberg in [329]. Considering different market participants, a game-theoretic problem with opponents optimizing different criteria is obtained. If this problem class is reduced to its most simple case, the interaction of two market participants has to be analyzed and in case of a hierarchical structure, e.g., a market leader controlling the prices (*leader*) and small business searching for its niche (*follower*), a mathematical problem results which fits into the bilevel framework derived in the following. Posing the problem in the optimization setup, the works of Bracken and McGill [36, 37] start an intensive investigation of bilevel programming. Since only fundamental properties of bilevel programs can be discussed here, we refer, for example, to the books [11, 21, 74, 212] and the literature cited therein for more details.

The first element to define is the *lower level program*; here we assume that this program is given in the form of a standard nonlinear optimization problem:

Definition 2.1.1. ***(Lower Level Program (LLP))***
Given a nonlinear cost function $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the inequality conditions $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and the equality conditions $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$, all of whom are assumed to be sufficiently smooth, the ***lower level program*** *reads*

$$\min_{x \in \mathbb{R}^n} \phi(x, y) \quad \text{subject to} \quad g(x, y) \leq 0 \text{ and } h(x, y) = 0,$$

where $y \in \mathbb{R}^m$ is a fixed parameter.

The solution set of the lower level program consequently depends on the choice of the parameter y which has to be provided by the second optimization problem, the *upper level program*. Therefore, we denote by $\mathbb{L}(y)$ the parameter-dependent solution set of the lower level program. If more than one element is optimal for the lower level program, it raises the question which element is returned by the follower to the leader; to capture this choosing process we introduce the function $\Upsilon : \mathbb{L} \to \mathbb{R}^n$. Depending on the selection different types of bilevel programs are obtained. In one case, called the optimistic case, the follower returns the element of the solution set corresponding to the best possible choice with respect to the cost function and the constraints of the leader. In another case, the pessimistic one, the follower chooses the element out of the solution set that results in the worst values of cost function and constraints for the leader.

Consequently, the most general form of an upper level program is given by the following definition:

Definition 2.1.2. ***(General Upper Level Program (GULP))***
Given a nonlinear cost function $\Phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the inequality conditions $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{\underline{p}}$ and the equality conditions $H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{\underline{q}}$, all of whom are assumed to be sufficiently smooth, the ***general upper level program*** *reads*

$$\min_{y \in \mathbb{R}^m} \Phi(x(y), y)$$

subject to

$$\begin{aligned} G(x(y), y) &\leq 0, \\ H(x(y), y) &= 0, \\ x(y) &= \Upsilon(\mathbb{L}(y)). \end{aligned}$$

In the following, the discussion is restricted to the special type of bilevel programs where the upper level constraints are independent of the LLP state, thus the solution of the lower level program influences only the ULP cost function. Consequently, the feasible set for the upper level program is defined by

$$\mathbb{Y} = \{y \mid H(y) = 0,\ G(y) \leq 0\},$$

where the functions H and G are the upper level constraints reduced to an y input only. In addition to this simplification of the ULP constraints, only the case of optimistic solutions is considered for the bilevel program here.

Definition 2.1.3. ***(Optimistic Solution)***
A point $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^m$ is called a ***local optimistic solution*** *of the bilevel program if y^* is feasible, i.e., $y^* \in \mathbb{Y}$, and $x^* \in \mathbb{L}(y^*)$ with*

$$\Phi(x^*, y^*) \leq \Phi(x, y^*) \quad \forall x \in \mathbb{L}(y^*)$$

and there exists an open neighborhood $\mathbb{U}_\varepsilon(y^*)$, $\varepsilon > 0$, with

$$\min_x\{\Phi(x, y^*) \mid x \in \mathbb{L}(y^*)\} \leq \min_x\{\Phi(x, y) \mid x \in \mathbb{L}(y)\}$$

for all $y \in \mathbb{Y} \cap \mathbb{U}_\varepsilon(y^*)$. If the conditions hold true for all $\varepsilon > 0$, the point is called a ***global optimistic solution****.*

In consequence, the following form of the upper level program is used throughout this chapter:

Definition 2.1.4. ***(Upper Level Program (ULP))***
Given the cost function $\Phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the inequality conditions $G : \mathbb{R}^m \to \mathbb{R}^p$ and the equality conditions $H : \mathbb{R}^m \to \mathbb{R}^q$, the ***upper level program*** *reads*

$$\min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} \Phi(x, y)$$

subject to $G(y) \leq 0$, $H(y) = 0$ and $x \in \mathbb{L}(y)$, $y \in \mathbb{Y}$, where the feasible set $\mathbb{Y}$ is assumed to be a closed set.

2.2 Existence of a Global Optimistic Solution

The goal of this section is to prove the existence of a global optimistic solution for the bilevel program; the presentation is according to [74]. We start with the introduction of (upper) semicontinuity and the definition of two conditions for the successive theorems. Denote by $\widehat{\mathbb{P}}(\mathbb{M})$ the power set of a given set $\mathbb{M}$, i.e., $\widehat{\mathbb{P}}(\mathbb{M})$ is the family of all subsets of $\mathbb{M}$.

Definition 2.2.1. ***(Upper Semicontinuity)***
A mapping $\Gamma : \mathbb{R}^m \to \widehat{\mathbb{P}}(\mathbb{R}^n)$ is called ***upper semicontinuous*** *at a point $w \in \mathbb{R}^m$ if the following condition holds:*
For all open neighborhoods $\mathbb{O}$ of $\Gamma(w)$, i.e., $\mathbb{O}$ is open and $\Gamma(w) \subset \mathbb{O}$, there exists a scalar $\varepsilon > 0$ such that

$$\Gamma(\widehat{w}) \subset \mathbb{O} \quad \forall \widehat{w} \in \mathbb{U}_\varepsilon(w).$$

A mapping is called upper semicontinuous if and only if it is upper semicontinuous at any point of its domain. An alternative definition of the upper semicontinuity of a set-valued mapping Γ is given by the following lemma:

Lemma 2.2.2.
Let $\Gamma(w)$ be compact for a vector $w \in \mathbb{R}^m$. The mapping Γ is upper semicontinuous in w if and only if

$$\forall \varepsilon_x > 0 \;\; \exists \varepsilon_y > 0 : \;\; \Gamma(\widehat{w}) \subset \mathbb{U}_{\varepsilon_x}(\Gamma(w)) \;\; \forall \; \widehat{w} \in \mathbb{U}_{\varepsilon_y}(w). \tag{2.1}$$

The ε-ball of a set $\mathbb{M}$ is defined by $\mathbb{U}_\varepsilon(\mathbb{M}) := \{\widehat{z} \mid \exists \; z \in \mathbb{M} : \; \widehat{z} \in \mathbb{U}_\varepsilon(z)\}$.

Proof. On the one hand the set $\mathbb{U}_{\varepsilon_x}(\Gamma(w))$ is an open neighborhood of $\Gamma(w)$ and the definition of the upper semicontinuity 2.2.1 yields then the condition (2.1) by setting $\varepsilon_y := \varepsilon$.

On the other hand each open neighborhood of a compact set $\Gamma(w)$ contains a ball $\mathbb{U}_{\varepsilon_x}(\Gamma(w))$ for a scalar $\varepsilon_x > 0$. Consequently, the condition (2.1) guarantees for this specific choice for ε_x the set inclusion of the definition 2.2.1. □

Note that for a single-valued function the condition (2.1) guarantees continuity; however a main property of a continuous single-valued function, which is the convergence of the images of a convergent sequence in its domain against the image of the corresponding limit, does not hold true for set-valued function fulfilling (2.1). Therefore the following complementary definition of semicontinuity for a set-valued functions is given:

Definition 2.2.3. ***(Lower Semicontinuity)***
*A mapping $\Gamma : \mathbb{R}^m \to \widehat{\mathbb{P}}(\mathbb{R}^n)$ is called **lower semicontinuous** at a point $w \in \mathbb{R}^m$ if the following condition holds:*
For each open set $\mathbb{O}$ with $\mathbb{O} \cap \Gamma(w) \neq \emptyset$ there is a scalar $\varepsilon > 0$ such that $\mathbb{O} \cap \Gamma(\widehat{w}) \neq \emptyset$ for each $\widehat{w} \in \mathbb{U}_\varepsilon(w)$.

The following lemma gives an alternative definition for lower semicontinuity of a set-valued mapping:

Lemma 2.2.4.
The mapping Γ is lower semicontinuous at $w \in \mathbb{R}^m$ if and only if for all $z \in \Gamma(w)$ and for all sequences $(\widehat{w}^{(k)}) \subset \mathbb{R}^m$ converging to w, there exists a sequence $(\widehat{z}^{(k)}) \subset \mathbb{R}^n$ and a $\underline{k} \in \mathbb{N}$ with

$$\widehat{z}^{(k)} \in \Gamma(\widehat{w}^{(k)}) \;\; \forall k \geq \underline{k} \;\; \text{and} \;\; \widehat{z}^{(k)} \to z \text{ for } k \to \infty. \tag{2.2}$$

Proof. Both implications are addressed separately here; first it is shown that the condition (2.2) is guaranteed by the definition of a lower semicontinuous function 2.2.3:

Let $w \in \mathbb{R}^m$ and $z \in \Gamma(w)$ be given; consider a radius $\varepsilon_x > 0$, then $\mathbb{U}_{\varepsilon_x}(z)$ is an open set and $z \in \mathbb{U}_{\varepsilon_x}(z) \cap \Gamma(w) \neq \emptyset$. Consequently, there exists an $\varepsilon_y > 0$ such that

$$z \in \mathbb{U}_{\varepsilon_x}(z) \cap \Gamma(\widehat{w}) \neq \emptyset \;\; \forall \widehat{w} \in \mathbb{U}_{\varepsilon_y}(w).$$

Let $\left(\widehat{w}^{(k)}\right)$ be a sequence converging to w as given in lemma 2.2.4, then an index $\underline{k} \in \mathbb{N}$ exists such that $\widehat{w}^{(k)} \in \mathbb{U}_{\varepsilon_y}(w)$ for all $k \geq \underline{k}$. Thus $\mathbb{U}_{\varepsilon_x}(z) \cap \Gamma(\widehat{w}^{(k)}) \neq \emptyset$ for all $k \geq \underline{k}$, which means that elements $\overline{z}^{(k)} \in \Gamma(\widehat{w}^{(k)})$ fulfilling $||\overline{z}^{(k)} - z|| \leq \varepsilon_x$ exist for all $k \geq \underline{k}$.

Note that the sequence $(\overline{z}^{(k)})$ depends implicitly on the chosen $\varepsilon_x > 0$, therefore denote them in the following with $\overline{z}^{(k)}(\varepsilon_x)$ and $\underline{k}(\varepsilon_x)$. Consequently, define the sequence $(\widehat{z}^{(k)})$ by using the decreasing sequence $(\varepsilon_x^{(j)})$ with $\varepsilon_x^{(j)} := 2^{-j}$, $j \in \mathbb{N}$:

$$\underline{k} := \underline{k}(\varepsilon_x^{(1)}) \;\text{ and }\; \widehat{z}^{(k)} := \overline{z}^{(k)}(\varepsilon_x^{(j)}) \;\text{ for }\; \underline{k}(\varepsilon_x^{(j)}) \leq k < \underline{k}(\varepsilon_x^{(j+1)}),\; j \in \mathbb{N}.$$

This construction assures that the convergence $\widehat{z}^{(k)} \to z$ holds as well as $\widehat{z}^{(k)} \in \Gamma(\widehat{w}^{(k)})$ for all $k \geq \underline{k}$.

Second, we show that the definition of a lower semicontinuous function 2.2.3 implicates the condition (2.2):

Let $\mathbb{U}$ be an open set with $\mathbb{U} \cap \Gamma(w) \neq \emptyset$, then an element $z \in \Gamma(w)$ exists with $z \in \mathbb{U}$. Since $\mathbb{U}$ is an open set, a scalar $\varepsilon > 0$ exists with $\mathbb{U}_\varepsilon(z) \subset \mathbb{U}$. The lower semicontinuity is proven by contradiction; therefore assume that no $\varepsilon_y > 0$ fulfills:

$$\Gamma(w) \cap \mathbb{U} \neq \emptyset \;\; \forall \widehat{w} \in \mathbb{U}_{\varepsilon_y}(w).$$

This means that for all $\varepsilon_y > 0$ there exists a $\widehat{w} \in \mathbb{U}_{\varepsilon_y}(w)$ such that $\Gamma(\widehat{w}) \cap \mathbb{U} = \emptyset$. Considering the sequence $(\varepsilon_y^{(k)}) \subset \mathbb{R}$ with $\varepsilon_y^{(k)} := 2^{-k}$ for $k \in \mathbb{N}$, a corresponding sequence $(\widehat{w}^{(k)})$ is obtained with

$$\widehat{w}^{(k)} \in \mathbb{U}_{\varepsilon_y^{(k)}}(w) \;\text{ and }\; \Gamma(\widehat{w}^{(k)}) \cap \mathbb{U} = \emptyset.$$

By construction the sequence $(\widehat{w}^{(k)})$ converges to w for $k \to \infty$. Consequently, there exists a sequence $(\widehat{z}^{(k)})$ and a scalar $\underline{k} \in \mathbb{N}$ such that

$$\widehat{z}^{(k)} \in \Gamma(\widehat{w}^{(k)}) \;\; \forall k \geq \underline{k} \;\text{ and }\; \widehat{z}^{(k)} \to z \;\text{ for }\; k \to \infty.$$

Since $\mathbb{U}_\varepsilon(z) \subset \mathbb{U}$ and $\widehat{z}^{(k)} \in \Gamma(\widehat{w}^{(k)})$ for all $k \geq \underline{k}$, but also $\Gamma(\widehat{w}^{(k)}) \cap \mathbb{U} = \emptyset$ holds, the following inequality results:

$$||z - \widehat{z}^{(k)}|| \geq \frac{\varepsilon}{2} \;\; \forall k \geq \underline{k}.$$

This leads in limit $k \to \infty$ to a contradiction to $\widehat{z}^{(k)} \to z$, which concludes the proof. □

The following two regularity conditions are used to prove the (upper) semicontinuity of the point-to-set mapping $\mathbb{L}(\cdot)$ relating the upper level state to the set of (global) optimal values for the respective lower level program.

Definition 2.2.5. ***(Compactness Assumption (BL-C))***
The set $\{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m \mid g(x,y) \leq 0,\; h(x,y) = 0\}$ *is non-empty and compact.*

Definition 2.2.6. ***(Mangasarian-Fromowitz constraint qualification (BL-MFCQ))***
The BL-MFCQ is fulfilled at a point (x,y) *in the bilevel setup if there exists a direction* $d \in \mathbb{R}^n$ *satisfying*

$$\begin{aligned} \nabla_x g_i(x,y)^T d &< 0, \quad \forall i \in \{j \mid g_j(x,y) = 0\}, \\ \nabla_x h_i(x,y)^T d &= 0, \quad \forall i = 1, \ldots, q, \end{aligned}$$

and the gradients $\nabla_x h_i(x,y)$, $i = 1, \ldots, q$, *are linearly independent.*

To state the next theorem, the following mappings have to be defined. The first one is the mapping of the ULP state to the corresponding feasible set $\mathbb{X} : \mathbb{R}^m \to \widehat{\mathbb{P}}(\mathbb{R}^n)$,

$$\mathbb{X}(y) := \{x \in \mathbb{R}^n \mid h(x,y) = 0,\; g(x,y) \leq 0\}.$$

Then, the optimal value function $\varpi : \mathbb{R}^m \to \mathbb{R}$ is defined as a function of the upper level state y by

$$\varpi(y) := \min\{\phi(x, y) \mid x \in \mathbb{X}(y)\}$$

and $\varpi(y) := \infty$ if $\mathbb{X}(y) = \emptyset$. Finally, the global solution mapping $\mathbb{L} : \mathbb{R}^m \to \widehat{\mathbb{P}}(\mathbb{R}^n)$ is given by

$$\mathbb{L}(y) := \{x \in \mathbb{X}(y) \mid \phi(x, y) = \varpi(y)\}.$$

In order to show that the optimal value function $\varpi(\cdot)$ is continuous under certain assumptions, the following lemma addressing the upper semicontinuity of the feasible set mapping $\mathbb{X}(\cdot)$ is used:

Lemma 2.2.7.
Let the functions g_j, $j = 1, \ldots, p$, and h_i, $i = 1, \ldots, q$, be continuous and let assumption BL-C be fulfilled. Then the feasible set mapping $\mathbb{X}(\cdot)$ is upper semicontinuous.

Proof. The condition BL-C assures that $\mathbb{X}(y)$ is compact for all feasible upper level states y. Accoring to lemma 2.2.2 it has to be proven that

$$\forall \varepsilon_x > 0 \;\; \exists\, \varepsilon_y > 0 : \;\; \mathbb{X}(\widehat{y}) \subset \mathbb{U}_{\varepsilon_x}(\mathbb{X}(y)) \;\; \forall \widehat{y} \in \mathbb{U}_{\varepsilon_y}(y).$$

Using the contradiction approach, we assume that

$$\exists\, \varepsilon_x > 0 \;\; \forall\, \varepsilon_y > 0 : \;\; \mathbb{X}(\widehat{y}) \not\subset \mathbb{U}_{\varepsilon_x}(\mathbb{X}(y)), \;\; \widehat{y} \in \mathbb{U}_{\varepsilon_y}(y).$$

Let $(\varepsilon_y^{(k)})$ be a sequence with $\varepsilon_y^{(k)} > 0$, $k \in \mathbb{N}$ and $\varepsilon_y^{(k)} \to 0$ for $k \to \infty$, then the assumption yields a sequence $(\widehat{y}^{(k)})$ with $||\widehat{y}^{(k)} - y|| < \varepsilon_y^{(k)}$ and a sequence $(\widehat{x}^{(k)})$ with $\widehat{x}^{(k)} \in \mathbb{X}(\widehat{y}^{(k)})$, but $\widehat{x}^{(k)} \notin \mathbb{U}_{\varepsilon_x}(\mathbb{X}(\widehat{y}))$. Consequently,

$$||\widehat{x}^{(k)} - x|| \geq \frac{\varepsilon_x}{2} \;\; \forall x \in \mathbb{X}(y), \;\; k \in \mathbb{N}. \tag{2.3}$$

The compactness assumption BL-C guarantees that the sequence $((\widehat{x}^{(k)})^T, (\widehat{y}^{(k)})^T)^T$ has an accumulation point $(\underline{x}^T, \underline{y}^T)^T$. Without loss of generality identify the subsequence converging to the accumulation point with the original one. Note that the assumption $||\widehat{y}^{(k)} - y|| < \varepsilon_y^{(k)}$ yields $\underline{y} = y$. The continuity of the functions g_j, $j = 1, \ldots, p$, results in

$$g_j(\underline{x}, y) = g_j\left(\lim_{k\to\infty} \widehat{x}^{(k)}, \lim_{k\to\infty} \widehat{y}^{(k)}\right) = \lim_{k\to\infty} g_j\left(\widehat{x}^{(k)}, \widehat{y}^{(k)}\right) \leq 0, \quad j = 1, \ldots, p,$$

since $\widehat{x}^{(k)} \in \mathbb{X}(\widehat{y}^{(k)})$. Analogue, for the equality constraints follows

$$h_i(\underline{x}, y) = 0, \quad i = 1, \ldots, q.$$

This shows that $\underline{x} \in \mathbb{X}(y)$, but this contradicts

$$||\underline{x} - x|| \geq \frac{\varepsilon_x}{2} \;\; \forall x \in \mathbb{X}(y),$$

which is a consequence of equation (2.3). □

Theorem 2.2.8.
Consider the lower level problem 2.1.1 with a given parameter $y \in \mathbb{R}^m$ with $\mathbb{X}(y) \neq \emptyset$. If the assumptions BL-C and BL-MFCQ are fulfilled at all points (x, y) with $x \in \mathbb{X}(y)$, the global solution mapping $\mathbb{L}(\cdot)$ is upper semicontinuous and the optimal value function $\varpi(\cdot)$ is continuous in y.

Proof. The first part of the proof is to show the continuity of the mapping $\mathbb{X}(\cdot)$ meaning that both the upper semicontinuity and the lower semicontinuity hold true. In the second part these properties are used to deduce the continuity of the optimal value function $\varpi : \mathbb{R}^m \to \mathbb{R}$. The upper semicontinuity of the global solution mapping $\mathbb{L}(\cdot)$ is shown in the last part of the proof.

According to lemma 2.2.7, the upper semicontinuity of the mapping $\mathbb{X}(\cdot)$ results from assumption BL-C and from the continuity of the functions g_j, $j = 1, \ldots, p$, and h_i, $i = 1, \ldots, q$.

The first step to prove the lower semicontinuity of the $\mathbb{X}(\cdot)$ is to reduce the problem to strictly feasible points: The sets of strictly feasible points are defined by

$$\mathbb{X}^s(y) := \{x \in \mathbb{R}^n \mid h(x, y) = 0,\ g(x, y) < 0\}.$$

The condition BL-MFCQ implies that for each $\underline{x} \in \mathbb{X}(\underline{y})$ there exists a sequence $\left(x^{(k)}\right) \subset \mathbb{X}^s(\underline{y})$ converging to $\underline{x}$. Thus, the inclusion $\mathbb{X}(\underline{y}) \subset \overline{\mathbb{X}^s(\underline{y})}$ holds, i.e., each open set with non-empty intersection with $\mathbb{X}(\underline{y})$ has also an non-empty intersection with $\mathbb{X}^s(\underline{y})$. Consequently, we assume that $\underline{x} \in \mathbb{X}^s(\underline{y})$ and obtain due to the continuity of g that constants $\varepsilon_x > 0$ and $\varepsilon_y > 0$ exist such that

$$g(x, y) < 0 \quad \forall (x, y) : ||x - \underline{x}|| \leq \varepsilon_x,\ ||y - \underline{y}|| \leq \varepsilon_y.$$

Furthermore, the condition BL-MFCQ assures that the gradients $\nabla_x h_i(x, y)$, $i = 1, \ldots, q$, are linearly independent. Using the implicit function theorem, this independence guarantees the existence of a continuous function $x(\cdot)$ defined on an open set $\mathbb{U}_{\varepsilon'_y}(\underline{y})$, $\varepsilon'_y > 0$, with $h(x(y), y) \equiv 0$ for $y \in \mathbb{U}_{\varepsilon'_y}(\underline{y})$ and $x(\underline{y}) = \underline{x}$. As a consequence of the continuity, for all $\varepsilon_x > 0$ there exists a constant $\varepsilon''_y > 0$ with $\varepsilon''_y < \min\{\varepsilon_y, \varepsilon'_y\}$ such that

$$||x(y) - \underline{x}|| < \varepsilon_x \quad \forall y \quad \text{with} \quad ||y - \underline{y}|| < \varepsilon''_y.$$

Thus, $x(y) \in \mathbb{X}^s(y) \subset \mathbb{X}(y)$ for all $||y - \underline{y}|| < \varepsilon''_y$ and because each open set $\mathbb{U}$ with $\underline{x} \in \mathbb{U}$ contains a ε_x-ball about $\underline{x}$ for an $\varepsilon_x > 0$, the set mapping $\mathbb{X}(\cdot)$ is lower semicontinuous.

Having shown that both upper and lower semicontinuity hold for the feasible set mapping, the next step is to deduce continuity of the optimal value function $\varpi(\cdot)$. Let $\left(y^{(k)}\right) \subset \mathbb{R}^m$, $k \in \mathbb{N}$, be a sequence converging to $\underline{y}$ and define a corresponding sequence $\left(x^{(k)}\right) \subset \mathbb{R}^n$ by $x^{(k)} \in \mathbb{L}(y^{(k)})$ for all k. Note that without loss of generality $\mathbb{L}(y^{(k)}) \neq \emptyset$, since $\mathbb{L}(y) \neq \emptyset$ and $y^{(k)} \to y$; due to the upper semicontinuity of $\mathbb{X}(\cdot)$ an open neighborhood $\mathbb{U}(\underline{y})$ of $\underline{y}$ exists with non-empty $\mathbb{X}(y)$ for all $y \in \mathbb{U}(\underline{y})$. As a consequence of assumption BL-C, this sequence has at least one accumulation point $\underline{x}$ and each accumulation point fulfills $\underline{x} \in \mathbb{X}(\underline{y})$ due to the upper semicontinuity of $\mathbb{X}(\cdot)$: Since the sequence $(y^{(k)})$ converges to $\underline{y}$, it results:

$$\forall \varepsilon > 0 \quad \exists\, \underline{k} \in \mathbb{N} : \quad y^{(k)} \in \mathbb{U}_\varepsilon(\underline{y}) \quad \forall k \geq \underline{k}.$$

By lemma 2.2.2 the upper semicontinuity of $\mathbb{X}(\cdot)$ guarantees

$$\forall \varepsilon_x > 0 \quad \exists\, \varepsilon_y > 0 : \quad \mathbb{X}(\widehat{y}) \subset \mathbb{U}_{\varepsilon_x}(\mathbb{X}(\underline{y})) \quad \forall \widehat{y} \in \mathbb{U}_{\varepsilon_y}(\underline{y}).$$

Consequently, for all $\varepsilon_x > 0$ there exists a $\underline{k} \in \mathbb{N}$ such that $x^{(k)} \in \mathbb{U}_{\varepsilon_x}(\mathbb{X}(\underline{y}))$ for all indices $k \geq \underline{k}$. This shows that $\underline{x} \in \mathbb{X}(\underline{y})$, because $\mathbb{X}(\underline{y})$ is closed.

This results in

$$\liminf_{k\to\infty} \varpi(y^{(k)}) = \liminf_{k\to\infty} \phi(x^{(k)}, y^{(k)}) \geq \varpi(\underline{y}).$$

Now, to prove the complementary lim sup-inequality, let be $\underline{x} \in \mathbb{L}(\underline{y})$ and let $(y^{(k)})$ be a sequence converging to $\underline{y}$. Then using lemma 2.2.4 the lower semicontinuity of $\mathbb{X}(\cdot)$ implies that a sequence $x^{(k)} \in \mathbb{X}(y^{(k)})$, $k \in \mathbb{N}$, exists which converges to $\underline{x}$. Consequently,

$$\limsup_{k\to\infty} \varpi(y^{(k)}) \leq \limsup_{k\to\infty} \phi(x^{(k)}, y^{(k)}) = \varpi(\underline{y}),$$

which implies the continuity of the optimal value function.

Finally, the continuities of the cost function of the lower level program and the optimal value function imply that each accumulation point $\underline{x}$ of a sequence $x^{(k)} \in \mathbb{L}(y^{(k)})$, $k \in \mathbb{N}$, satisfies $\phi(\underline{x}, \underline{y}) = \varpi(\underline{y})$. Thus, $\underline{x} \in \mathbb{L}(\underline{y})$ and in combination with the assumption BL-C the upper semicontinuity of $\mathbb{L}(\cdot)$ results by using lemma 2.2.2.

The general layout of the presented proof has been sketched in [74]. □

In general, the dependence of the solution set of the lower level program on the upper level state is not lower semicontinuous; see, for example, the following simple problem:

Example 2.2.9. Using a scalar LLP state $x \in [-1, 1]$ and a scalar ULP state y, the lower level program is defined by:

$$\min_x \phi(x, y) := xy \quad \text{s.t.} \quad -1 \leq x \leq 1, \quad -1 \leq y \leq 1.$$

The solution set is given by

$$\mathbb{L}(y) = \begin{cases} \{-1\} & \text{if} \quad y > 0, \\ [-1, 1] & \text{if} \quad y = 0, \\ \{1\} & \text{if} \quad y < 0, \end{cases}$$

which is not lower semicontinuous since for $(x, y) = (0, 0)$ open neighborhoods of x exist which have no element in common with the solution sets for $y \neq 0$.

Theorem 2.2.10.
Let the conditions BL-C and BL-MFCQ hold at all points $(x, y) \in \mathbb{R}^n \times \mathbb{Y}$ with $x \in \mathbb{X}(y)$. Given a feasible point for the bilevel program, a global optimistic solution exists.

Proof. The conditions BL-C and BL-MFCQ are assumed to hold at all points $(x, y) \in \mathbb{R}^n \times \mathbb{Y}$ with $x \in \mathbb{X}(y)$ which means that the assumptions of theorem 2.2.8 are fulfilled for each upper level state y with $\mathbb{X}(y) \neq \emptyset$. Consequently, the solution set mapping $\mathbb{L}(\cdot)$ is upper semicontinuous for all $y \in \mathbb{Y}$ with $\mathbb{X}(y) \neq \emptyset$. This property in combination with the closed sets $\mathbb{L}(y)$ guarantees that the set $\{(x, y) \mid x \in \mathbb{L}(y)\}$ is closed:

Let $\left(x^{(k)}\right) \subset \mathbb{R}^n$ and $\left(y^{(k)}\right) \subset \mathbb{R}^m$ be convergent sequences with $x^{(k)} \in \mathbb{L}(y^{(k)})$ for all $k \in \mathbb{N}$ and $x^{(k)} \to \underline{x}$, $y^{(k)} \to \underline{y}$. To show that $\underline{x} \in \mathbb{L}(\underline{y})$, we note that $\mathbb{L}(\underline{y})$ is a closed set and define

$$\mathbb{U}_\delta(\mathbb{L}(\underline{y})) := \left\{x \in \mathbb{R}^n \mid ||x - \widetilde{x}|| < \delta, \ \widetilde{x} \in \mathbb{L}(\underline{y})\right\}$$

for a $\delta > 0$. Since $\mathbb{U}_\delta(\mathbb{L}(\underline{y}))$ is open and $\mathbb{L}(\underline{y}) \subset \mathbb{U}_\delta(\mathbb{L}(\underline{y}))$, the upper semicontinuity of $\mathbb{L}(\cdot)$ yields the existence of $\epsilon_y(\delta) > 0$ such that $\mathbb{L}(y) \subset \mathbb{U}_\delta(\mathbb{L}(\underline{y}))$ for all $y \in \mathbb{U}_{\epsilon_y(\delta)}(\underline{y})$.

Consequently, an index $\underline{k}(\delta)$ exists which guarantees $y^{(k)} \in \mathbb{U}_{\epsilon_y(\delta)}(\underline{y})$ for all $k \geq \underline{k}(\delta)$. Thus $\mathbb{L}(y^{(k)}) \subset \mathbb{U}_\delta(\mathbb{L}(\underline{y}))$ for all $k \geq \underline{k}(\delta)$ and $\operatorname{dist}(x^{(k)}, \mathbb{L}(\underline{y})) \leq \delta$ for all $k \geq \underline{k}(\delta)$. The limit $\delta \to 0$ shows that $\lim_{k\to\infty} x^{(k)} = \underline{x} \in \mathbb{L}(\underline{y})$.

The intersection of the closed set $\{(x, y) \mid x \in \mathbb{L}(y)\}$ with $\mathbb{R}^n \times \mathbb{Y}$ is compact because condition BL-C holds and $\mathbb{Y}$ is closed due to the general assumption in the ULP definition.

The upper level program 2.1.4 describes the minimization of the continuous upper level cost function $\Phi(\cdot, \cdot)$ over this compact set. As a consequence of the Bolzano-Weierstraß theorem, the existence of a global solution of the (optimistic) bilevel program is proven if a feasible point for the bilevel program exists.

The outline of this proof is in accordance with [74]. □

2.3 Solution Strategies for Bilevel Programs

In this section we want to mention some of the basic approaches used in literature to solve bilevel programs. Naturally, a greater number of approaches deal with *linear bilevel programs*, i.e., cost functions and constraints are linear in both the upper level and the lower level, or bilevel programs with the lower level program being a convex optimization problem than with the general nonlinear bilevel program 2.1.2, but nevertheless several different solution strategies are known. Here the basic ideas can only be sketched and we refer for details to [21, 74, 326] and the references cited therein.

The fact, that bilevel programs are in general a complex problem class, can already be seen in the linear case, because it is proven in [20, 139, 169] that solving a linear bilevel program is a strongly NP-hard problem. Typical solution strategies for such an problem are extreme point algorithms [57] and a branch-and-bound algorithms [21, 139]. Note that the bilevel structure is closely related to multi-criteria optimization, but the different problems have distinct differences, see for example [113].

A first group of strategies to solve the general bilevel problem is given by methods that consider the lower level program as a black box for the upper level and thus (iteratively) exploit the bilevel structure by using lower level solutions for different upper level states. Both a method of derivative-free optimization or a descent method using sensitivity information of the lower level solution are numerical strategies for such an problem structure, e.g., [90]. Furthermore, note that the implicit function theorem proves to be a suitable tool to analyze this problem type. Under suitable assumptions a local function $x(y) : \mathbb{R}^m \to \mathbb{R}^n$ is given by this theorem and can be integrated into the upper level program formulations. Consequently, implicitly defined constraints or cost functions are obtained. This approach can be used to prove necessary conditions using Clark subgradients and their structure can be used to deduce a suitable bundle technique of non-smooth optimization, cf. [242].

Another approach is based on the definition of an *optimal value function* $\varpi(y)$ by

$$\varpi(y) := \min_{x \in \mathbb{R}^n} \{\phi(x, y) \mid g(x, y) \leq 0,\ h(x, y) = 0\}.$$

In this case the lower level program can be replaced by the non-differentiable constraint $\phi(x, y) \leq \varpi(y)$ yielding an one-level problem. Note that the differentiability properties of the reformulated problem can be used to obtain under suitable assumptions necessary condition

for the original bilevel program. Such conditions are, for example, proven in [354] using the theory on subgradients of the Clark-type or in [75] utilizing the more general approach of convexificators. Note that this approach is closely related to the approach of [242] mentioned above and consequently, similar subgradient methods can be deduced to solve the bilevel program [241].

The approach we use to reformulate the bilevel program as a one-level problem is based on the KKT-conditions of the lower level program. Note that these necessary optimality conditions are only sufficient if the lower level program is a convex optimization problem. Since this is in general not the case, the solution of the reformulated problem might yield only a lower bound for the optimal cost value of the bilevel program. If the lower level program is substituted by its KKT-conditions a one-level program with equilibrium constraints, a so-called MPEC, is obtained; see the next sections for details on MPECs. This reformulation strategy has been succesfully used in several publications, e.g., [18, 19, 86].

2.4 Optimality Conditions for MPECs

Given two vectors z^I and $z^{II} \in \mathbb{R}^s$, the complementarity of these are denoted in this chapter by

$$0 \leq z^I \ \perp \ z^{II} \geq 0,$$

which means

$$\begin{aligned} 0 &\leq z^I, \\ 0 &\leq z^{II}, \\ 0 &= \min\left\{z^I, z^{II}\right\}. \end{aligned}$$

Note that both the inequalities of a vector with a scalar and the minimum operation on two vectors have to be interpreted elementwise. This complementarity constraint can equivalently be written as

$$\begin{aligned} &(i) \qquad z^I \geq 0, \quad z^{II} \geq 0, \quad \left(z^I\right)^T z^{II} = 0, \\ &(ii) \qquad z_i^I \geq 0, \quad z_i^{II} \geq 0, \quad z_i^I z_i^{II} = 0, \quad i = 1, \ldots, s. \end{aligned}$$

To distinguish between cases where both values are zero and where one value is non-zero, the following terms are introduced:

Definition 2.4.1. ***(Degeneration)***

(i) The tuple $(z_i^I, z_i^{II}) \in \mathbb{R}^2$ for $i \in \{1, \ldots, s\}$ is called ***degenerate*** *if both values equal zero:*

$$z_i^I = z_i^{II} = 0.$$

(ii) If either $z_i^I > 0$ or $z_i^{II} > 0$, then the tuple is called ***non-degenerate****.*

(iii) If all components of z^I and z^{II} are non-degenerate, then ***strict complementarity*** *is fulfilled by the two vectors.*

The following definition states the structure of the MPECs considered in this work:

Definition 2.4.2. ***(MPEC)***
A ***mathematical program with equilibrium constraints*** *is a nonlinear optimization problem with additional complementarity constraints:*

$$\begin{aligned} & \min_x \phi(x) \\ \textit{subject to} \quad & h(x) = 0, \\ & g(x) \leq 0, \\ & 0 \leq z^I \perp z^{II} \geq 0, \end{aligned}$$

where the optimization variable $x \in \mathbb{R}^n$ is the concatenation of a vector $z^o \in \mathbb{R}^{n-2s}$ and the complementary vectors $z^I, z^{II} \in \mathbb{R}^s$:

$$x = \left((z^o)^T, \left(z^I\right)^T, \left(z^{II}\right)^T \right)^T.$$

Again, all functions, i.e., the cost function ϕ, the equality constraints h and the inequality constraints g, are assumed to be at least twice continuously differentiable and the set of feasible points is denoted by $\mathbb{X}$.

Note that standard constraint qualifications do not hold for the complementarity condition. If the condition is formulated using smooth functions, the gradients of the active constraints become linearly dependent, thus the LICQ cannot hold. Additionally, the complementarity condition prohibits strictly feasible points and consequently, the MFCQ is also violated at every feasible point [106, 245, 276]. Furthermore, if a feasible point for the MPEC has degenerate components, then the tangent cone is non-convex and consequently, the ACQ does not hold.

Therefore, we now state standard MPEC-stationarity concepts and start with B-stationarity which is a necessary optimality condition for a local solution of an MPEC:

Definition 2.4.3. ***(B-Stationarity)***
A vector $x \in \mathbb{X}$ is called ***B-stationary*** *(B from Bouligand), if*

$$\nabla \phi(x)^T d \geq 0 \quad \forall d \in \mathbb{T}(\mathbb{X}, x).$$

For the further definitions we introduce the Lagrange multipliers ψ^I and $\psi^{II} \in \mathbb{R}^s$ for the inequalities assuring positivity of z^I and z^{II} and define a suitable version of the Lagrangian for the MPEC:

$$L_M\left(x, \lambda, \mu, \psi^I, \psi^{II}\right) := \phi(x) + \lambda^T g(x) + \mu^T h(x) - \left(\psi^I\right)^T z^I - \left(\psi^{II}\right)^T z^{II}.$$

Note that this version of a Lagrangian does not take the (full) complementarity conditions into account. In addition, we define the active sets for the inequalities of the complementarity condition by

$$\mathbb{A}_I(x) := \{ i \in \mathbb{N} \mid i \leq s,\ z_i^I = 0 \}$$

and

$$\mathbb{A}_{II}(x) := \{ i \in \mathbb{N} \mid i \leq s,\ z_i^{II} = 0 \}.$$

Definition 2.4.4. ***(Stationarity Concepts)***
Consider the following system of equations and inequalities for given values x, λ, μ, ψ^I and ψ^{II}:

$$\begin{array}{rcll} \nabla_x L_M\left(x,\lambda,\mu,\psi^I,\psi^{II}\right) & = & 0, & \\ h(x) & = & 0, & \\ g(x) & \leq & 0, & \\ \lambda & \geq & 0, & \\ g_i(x)\lambda_i & = & 0, & i=1,\ldots,q, \\ z^I & \geq & 0, & \\ z^{II} & \geq & 0, & \\ z_i^I = 0 \quad or \quad z_i^{II} & = & 0, & i=1,\ldots,s, \\ z_i^I\psi_i^I & = & 0, & i=1,\ldots,s, \\ z_i^{II}\psi_i^{II} & = & 0, & i=1,\ldots,s,. \end{array} \tag{2.4}$$

The following stationarity concepts are introduced:

- *A feasible point x is called **C-stationary** (C from Clarke) if multipliers λ, μ, ψ^I and ψ^{II} exist, such that system (2.4) is satisfied and the condition*

$$\forall i \in (\mathbb{A}_I \cap \mathbb{A}_{II})(x): \quad \psi_i^I\psi_i^{II} \geq 0$$

  is fulfilled.

- *A feasible point x is called **M-stationary** (M from Mordukhovich) if multipliers λ, μ, ψ^I and ψ^{II} exist, such that system (2.4) is satisfied and the condition*

$$\forall i \in (\mathbb{A}_I \cap \mathbb{A}_{II})(x): \quad \psi_i^I, \psi_i^{II} \geq 0 \quad or \quad \psi_i^I\psi_i^{II} = 0$$

  is fulfilled.

- *A feasible point x is called **strongly stationary** if multipliers λ, μ, ψ^I and ψ^{II} exist, such that system (2.4) is satisfied and the condition*

$$\forall i \in (\mathbb{A}_I \cap \mathbb{A}_{II})(x): \quad \psi_i^I \geq 0 \quad and \quad \psi_i^{II} \geq 0$$

  is fulfilled.

Note that the set $(\mathbb{A}_I \cap \mathbb{A}_{II})(x)$ consists of the degenerate components of x and if this set is empty, all three stationarity concepts are equal. On the other hand, if the set is non-empty, strong stationarity implies M-stationarity and this in turn implies C-stationarity. Problems are known which have a B-stationary point that is not strongly stationary, however, other problems with M-stationary points exist that are not B-stationary (see [324] for an example). Consequently, further information on the individual MPEC is needed to decide which stationarity concept is suitable to characterize candidates for local solutions [276, 353]. Such information can be provided by suitable constraint qualifications for MPECs; a large variety of such constraint qualifications exists, but we restrict ourselves to the most simple one and refer, for example, to [105, 353] for further details on other ones.

Definition 2.4.5. ***(MPEC-LICQ)***
Let x be a feasible point of the MPEC 2.4.2. Then the ***MPEC-LICQ*** *holds at x if the gradients*

$$\begin{array}{rll} \nabla h_i(x) & \text{for} & i = 1, \ldots, p, \\ \nabla g_i(x) & \text{for} & i \in \mathbb{A}, \\ e^{(n-2s+i)} & \text{for} & i \in \mathbb{A}_I, \\ e^{(n-s+i)} & \text{for} & i \in \mathbb{A}_{II}, \end{array}$$

are linearly independent. Here, $e^{(j)} \in \mathbb{R}^n$ is the j-th unit vector.

Note that the MPEC-LICQ differs from the LICQ for standard nonlinear programming problems in the point that only the conditions $z^I \geq 0$ and $z^{II} \geq 0$ of the complementarity condition are considered, which is in accordance with the definition of the Lagrangian for the MPEC. However, it is possible to construct so-called *relaxed nonlinear programs* (cf. 2.4.8) where the MPEC-LICQ represents the standard LICQ and where strong stationarity corresponds with the standard stationarity of nonlinear optimization problems (see for example [110, 276]). Assuming that the MPEC-LICQ holds, one can prove that B-stationarity implies strong stationarity [276], i.e., in this setting strong stationarity is a necessary optimality condition. More details on constraint qualifications and stationarity conditions can, for example, be found in [105, 353].

In addition to the stationarity concepts, second-order sufficient conditions are now discussed in order to relate local optimality to stationarity. Following again the line of [324] the following sets of critical directions are defined in accordance to [254]:

Definition 2.4.6. ***(Sets of Critical Directions)***
The ***sets of critical directions*** *defined by*

$$\begin{aligned} \mathbb{D}(x, \lambda, \mu, \psi^I, \psi^{II}) \;=\; \Big\{ & d = \left((d^o)^T, \left(d^I\right)^T, \left(d^{II}\right)^T \right)^T \in \mathbb{R}^n \backslash \{0\} \mid \\ & 0 = \nabla h(x)^T d, \\ & 0 = \nabla g_i(x)^T d, \quad \text{if } i \in \mathbb{A}(x) \text{ and } \lambda_i > 0, \\ & 0 \geq \nabla g_i(x)^T d, \quad \text{if } i \in \mathbb{A}(x) \text{ and } \lambda_i = 0, \\ & 0 = d_i^I, \qquad \text{if } i \in (\mathbb{A}_I \backslash \mathbb{A}_{II})(x), \\ & 0 = d_i^I, \qquad \text{if } i \in (\mathbb{A}_I \cap \mathbb{A}_{II})(x) \text{ and } \psi_i^I > 0, \\ & 0 \leq d_i^I, \qquad \text{if } i \in (\mathbb{A}_I \cap \mathbb{A}_{II})(x) \text{ and } \psi_i^I = 0, \\ & 0 = d_i^{II}, \qquad \text{if } i \in (\mathbb{A}_{II} \backslash \mathbb{A}_I)(x), \\ & 0 = d_i^{II}, \qquad \text{if } i \in (\mathbb{A}_{II} \cap \mathbb{A}_I)(x) \text{ and } \psi_i^{II} > 0, \\ & 0 \leq d_i^{II}, \qquad \text{if } i \in (\mathbb{A}_{II} \cap \mathbb{A}_I)(x) \text{ and } \psi_i^{II} = 0 \Big\}, \end{aligned}$$

$$\begin{aligned} \overline{\mathbb{D}}(x, \lambda, \mu, \psi^I, \psi^{II}) \;=\; \Big\{ & d \in \mathbb{D}(x, \lambda, \mu, \psi^I, \psi^{II}) \mid \\ & 0 = \min(d_i^I, d_i^{II}), \quad \text{if } i \in (\mathbb{A}_{II} \cap \mathbb{A}_I)(x) \text{ and } \psi_i^I = \psi_i^{II} = 0 \Big\}, \end{aligned}$$

$$\begin{aligned}\widetilde{\mathbb{D}}(x,\lambda,\mu,\psi^I,\psi^{II}) \;=\; \Big\{ d = \left((d^o)^T, \left(d^I\right)^T, \left(d^{II}\right)^T\right)^T \in \mathbb{R}^n\backslash\{0\} \mid \\ 0 = \nabla h(x)^T d, \\ 0 = \nabla g_i(x)^T d, \quad \text{if } i \in \mathbb{A}(x) \text{ and } \lambda_i > 0, \\ 0 = d_i^I, \quad \text{if } \psi_i^I \neq 0, \\ 0 = d_i^{II}, \quad \text{if } \psi_i^{II} \neq 0 \Big\},\end{aligned}$$

satisfy the relation $\overline{\mathbb{D}}(x,\lambda,\mu,\psi^I,\psi^{II}) \subseteq \mathbb{D}(x,\lambda,\mu,\psi^I,\psi^{II}) \subseteq \widetilde{\mathbb{D}}(x,\lambda,\mu,\psi^I,\psi^{II})$.

These sets of critical directions for the MPEC can now be used to define different types of second-order sufficient conditions.

Definition 2.4.7. ***(Second-Order Sufficient Conditions for MPECs)***

(i) Assume that x with the Lagrange multipliers $(\lambda,\mu,\psi^I,\psi^{II})$ is a strong stationary point of the MPEC 2.4.2. The ***MPEC second-order sufficient condition (MPEC-SOSC)*** *is satisfied at x if the inequality*

$$d^T L_M(x,\lambda,\mu,\psi^I,\psi^{II})d > 0$$

holds for all $d \in \overline{\mathbb{D}}(x,\lambda,\mu,\psi^I,\psi^{II})$.

(ii) If the inequality

$$d^T L_M(x,\lambda,\mu,\psi^I,\psi^{II})d > 0$$

holds for all $d \in \mathbb{D}(x,\lambda,\mu,\psi^I,\psi^{II})$, then the ***second-order sufficient conditions of the relaxed nonlinear program (RNLP-SOSC)*** *are fulfilled at x with $(\lambda,\mu,\psi^I,\psi^{II})$.*

(iii) If the inequality

$$d^T L_M(x,\lambda,\mu,\psi^I,\psi^{II})d > 0$$

holds for all $d \in \widetilde{\mathbb{D}}(x,\lambda,\mu,\psi^I,\psi^{II})$, then the ***strong second-order sufficient conditions(SSOSC)*** *are fulfilled at x with $(\lambda,\mu,\psi^I,\psi^{II})$.*

Remark 2.4.8. Note that the RNLP-SOSC are the standard second-order sufficient conditions A.2.3 applied to the following *relaxed nonlinear problem (RNLP)* belonging to the MPEC 2.4.2:

$$\begin{aligned}&\min_x \phi(x) \\ \text{subject to} \quad &h(x) = 0, \\ &g(x) \leq 0, \\ &z_i^I = 0, \quad z_i^{II} \geq 0, \quad \text{if } i \in (\mathbb{A}_I\backslash\mathbb{A}_{II})(x^*), \\ &z_i^I \geq 0, \quad z_i^{II} = 0, \quad \text{if } i \in (\mathbb{A}_{II}\backslash\mathbb{A}_I)(x^*), \\ &z_i^I \geq 0, \quad z_i^{II} \geq 0, \quad \text{if } i \in (\mathbb{A}_{II} \cap \mathbb{A}_I)(x^*),\end{aligned}$$

where the index sets are fixed with respect to a given point x^*. The theorem stating that the MPEC-SOSC is sufficient to guarantee the local optimality of a strong stationary point x for the MPEC 2.4.2 concludes this section:

Theorem 2.4.9. ***(Strict Local Minimum)***
If x is a strong stationary point of the MPEC 2.4.2 that satisfies the MPEC-SOSC for any Lagrange multipliers fulfilling the strong stationary conditions, then x is a strict local minimum of the MPEC.

Proof. This is also shown in [324] by applying a corresponding result of [277]. □

2.5 Regularization Strategy for MPECs

In literature several strategies are known to regularize the complementarity condition in order to obtain an nonlinear optimization problem fulfilling certain constraint qualifications of standard nonlinear optimization problems which are a basic requirement for standard nonlinear optimization methods. In this section we state the relaxation scheme of [290, 324] which can be seen as a combination of the relaxation approach of [277] and the regularization approach of [111]; consequently, the presentation follows the line of [324] and for further details we refer to this original work.

2.5.1 Relaxation Scheme

The basic idea of the relaxation scheme of [324] is to relax the complementarity condition for each pair z_i^I and z_i^{II}, $i = 1, \ldots, s$, only on a subset of the triangle with the vertices $(0,0)$, $(\delta_i, 0)$ and $(0, \delta_i)$. Note that for a sufficiently small relaxation parameter $\delta_i > 0$ the complementarity conditions are only modified for degenerate components and for the choice of $\delta_i = 0$ the original complementarity condition is obtained. Since the approach is independent for each of the scalar complementarity conditions, it is sufficient to discuss how to handle one of them and therefore, the range of the index i is not mentioned at each instance in the following.

A reparametrization of the complementarity problem

$$z_i^I \geq 0, \quad z_i^{II} \geq 0, \quad z_i^I z_i^{II} = 0$$

into the problem

$$\underline{z} = |\overline{z}|$$

by introducing $\underline{z} := z_i^I + z_i^{II}$ and $\overline{z} := z_i^I - z_i^{II}$ allows to post the relaxation problem as a problem of smoothing the absolute value function within the interval $[-\delta_i, \delta_i]$.

Definition 2.5.1. ***(Kink Smoothing Function)***
A function $\beta : I \to \mathbb{R}$ defined on an open interval I with $[-1,1] \subset I \subset \mathbb{R}$ is called an ***kink smoothing function*** *if the following conditions hold:*

(i) $\beta|_{[-1,1]} \in \mathcal{C}^2([-1,1], \mathbb{R})$,

(ii) $\beta(-1) = \beta(1) = 1$,

(iii) $\beta'(-1) = -1$ *and* $\beta'(1) = 1$,

(iv) $\beta''(-1) = \beta''(1) = 0$,

(v) $\beta''(\overline{z}) > 0 \quad \forall \overline{z} \in (-1, 1)$.

Two of such kink smoothing functions are introduced in [324]:

$$\beta_s(\overline{z}) := \frac{2}{\pi} \sin\left(\frac{(\overline{z}+3)\pi}{2}\right) + 1 \quad \text{and} \quad \beta_p(\overline{z}) := \frac{1}{8}\left(-\overline{z}^4 + 6\overline{z}^2 + 3\right).$$

Using an kink smoothing function for the scaled interval $[-\delta_i, \delta_i]$ and the absolute value function for the complement, the following function $\xi \in \mathcal{C}^2(\mathbb{R} \times \mathbb{R}_{\geq 0}, \mathbb{R})$ with

$$\xi(\overline{z}, \delta_i) := \begin{cases} |\overline{z}| & \text{for} \quad |\overline{z}| \geq \delta_i, \\ \\ \delta_i \beta\left(\delta_i^{-1}\overline{z}\right) & \text{for} \quad |\overline{z}| < \delta_i, \end{cases}$$

is obtained and can be used to write a relaxed version of the complementarity condition $\underline{z} = |\overline{z}|$:

$$\underline{z} \geq -\overline{z}, \quad \underline{z} \geq \overline{z}, \quad \underline{z} \leq \xi(\overline{z}, \delta_i).$$

Switching back to the original variables z_i^I and z_i^{II}, the function $\Xi_i \in \mathcal{C}^2(\mathbb{R} \times \mathbb{R} \times \mathbb{R}_{\geq 0}, \mathbb{R})$ defined by

$$\Xi_i(z_i^I, z_i^{II}, \delta_i) := z_i^I + z_i^{II} - \xi(z_i^I - z_i^{II}, \delta_i)$$

allows to state the complementarity condition for z_i^I and z_i^{II} in the form

$$z_i^I \geq 0, \quad z_i^{II} \geq 0, \quad \Xi_i(z_i^I, z_i^{II}, \delta_i) \leq 0.$$

Combining the individual functions Ξ_i for all $i = 1, \ldots, s$, the definition of the vector-valued function $\Xi \in \mathcal{C}^2(\mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^s_{\geq 0}, \mathbb{R}^s)$ is straightforward and results in

$$z^I \geq 0, \quad z^{II} \geq 0, \quad \Xi(z^I, z^{II}, \delta) \leq 0.$$

Note that the vector $\delta \in \mathbb{R}^s_{\geq 0}$ allows to individually determine a suitable relaxation for each scalar complementarity condition.

In consequence, the following parametric nonlinear optimization problem $\mathcal{R}(\delta)$ is obtained:

Definition 2.5.2. ***(Relaxed Problem*** $\mathcal{R}(\delta)$***)***

$$\begin{aligned} & \min_x \phi(x) \\ \textit{subject to} \quad & 0 = h(x), \\ & 0 \geq g(x), \\ & 0 \leq z^I, \\ & 0 \leq z^{II}, \\ & 0 \geq \Xi(z^I, z^{II}, \delta). \end{aligned}$$

By construction the relaxation properties of the scheme are evident: If a variable x^* is feasible for the original MPEC problem 2.4.2, then it also feasible for the relaxed problem 2.5.2. Denote the feasible set of $\mathcal{R}(\delta)$ by $\mathbb{X}_{\mathcal{R}}(\delta)$; if the inequality $\delta^{II} \leq \delta^I$ holds componentwise, then the following inclusion for the feasible set results:

$$\mathbb{X}_{\mathcal{R}}(\delta^{II}) \subseteq \mathbb{X}_{\mathcal{R}}(\delta^I).$$

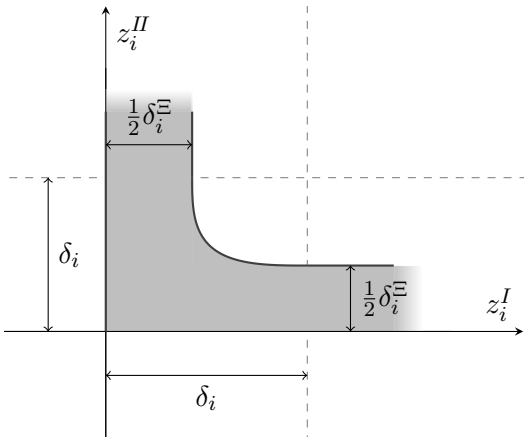


Figure 2.1: Illustration of the relaxation approach of [324] for one scalar complementarity condition using the smoothing function β_p.

Furthermore, if a strict local solution x^* of $\mathcal{R}(\delta^I)$ is feasible for the MPEC problem 2.4.2, then x^* is a strict local solution for all δ^{II} with $0 \leq \delta^{II} \leq \delta^I$. For more details and the proofs of these properties see [324]. The following theorem relates second-order sufficient conditions for the MPEC 2.4.2 to the second-order sufficient conditions of the relaxed problem 2.5.2:

Theorem 2.5.3.
Let x be feasible for the MPEC 2.4.2. Then the following implications hold true:

(i) If x is a strong stationary point for the MPEC satisfying the RNLP-SOSC, then there exist relaxation parameter $\delta > 0$ such that x is a stationary point for $\mathcal{R}(\delta)$ fulfilling the standard second-order sufficient condition A.2.3 for $\mathcal{R}(\delta)$. Consequently, the point x is a strict local minimum of $\mathcal{R}(\delta)$.

(ii) If x is a stationary point for $\mathcal{R}(\delta)$ that satisfies the standard second-order sufficient condition A.2.3, then this point is a strongly stationary point of the MPEC 2.4.2 that satisfies the RLNP-SOSC. Thus, x is a strict local minimum of the MPEC.

Proof. The proof of this theorem can be found in [324]. □

Note that convergence results assuming weaker constraint qualifications for the MPEC are also presented in [324] and extended in [160].

2.5.2 Further Regularization Approaches

In this section we discuss the most relevant regularization schemes for MPECs related to the scheme of [324]; for further details and additional approaches see [120, 324] and the references cited therein. The first regularization scheme that has to be discussed is the approach of [277] where the complementarity condition is relaxed to

$$z_i^I \geq 0, \quad z_i^{II} \geq 0, \quad z_i^I z_i^{II} \leq \delta_i$$

by introducing a parameter $\delta_i > 0$. The feasible region of this regularization approach is displayed in figure 2.2.

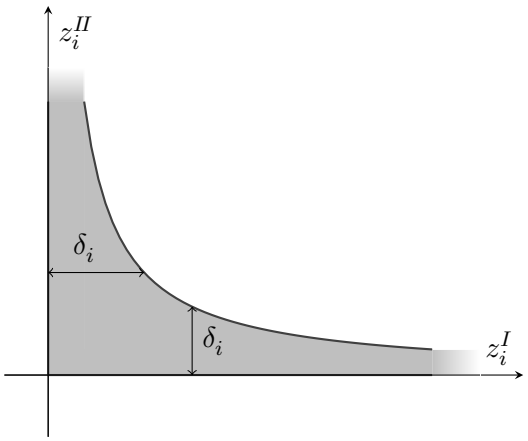


Figure 2.2: Illustration of the relaxation approach of [277] for one scalar complementarity condition.

It is proven in [277] that under assumptions including the MPEC-LICQ the sequence of stationary points for the relaxed problems converge to a C-stationary point with unique multipliers; B-stationarity or M-stationarity follows if additional assumptions are fulfilled. Furthermore, it is shown that a piecewise smooth mapping relating the relaxation parameter δ_i to the corresponding stationary point exists if suitable assumptions are met.

Second, in [111] the MPEC is rewritten in form of a standard nonlinear optimization problem by replacing the complementarity condition by

$$z^I \geq 0, \quad z^{II} \geq 0, \quad \left(z^I\right)^T z_i^{II} \leq 0.$$

Note that this reformulation of the MPEC does not actually change the structure of the MPEC, but it is reported in [110] that standard methods of sequential quadratic programming (SQP-methods) can solve several problems of this type. Under assumptions including the MPEC-LICQ and a second-order sufficient condition superlinear convergence is shown near a strongly stationary point. Several extensions of this approach are discussed in literature, e.g., [13, 193].

The third idea adapted in the relaxation scheme of [324] for interior-point methods is the two-sided relaxation introduced by [72]. Here, the positivity constraints are relaxed in addition to the part of the complementarity constraints given in product form:

$$z_i^I \geq -\widehat{\delta}_i^{\,I}, \quad z_i^{II} \geq -\widehat{\delta}_i^{\,II}, \quad z_i^I z_i^{II} \leq \delta_i, \quad i = 1, \ldots, s,$$

where the positive relaxation parameters $\widehat{\delta}_i^{\,I}$, $\widehat{\delta}_i^{\,II}$ and δ_i, $i = 1, \ldots, s$, are used. A schematic illustration of the feasible set for one of these scalar complementarity conditions is displayed in figure 2.3.

The relaxation parameters are updated such that either δ_i or at least one of $\widehat{\delta}_i^{\,I}$ and $\widehat{\delta}_i^{\,II}$ is reduced, which assures that a strictly feasible set exists that is even in limit non-empty.

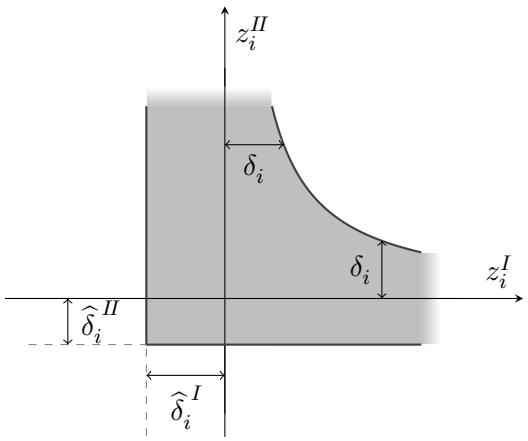


Figure 2.3: Illustration of the relaxation approach of [72] for one scalar complementarity condition.

Under suitable assumptions including MPEC-LICQ and SSOSC it is shown that this scheme converges superlinearly near a strongly stationary point; further details can be found in the next section.

2.5.3 Extension for Interior-Point Methods

In this section an extension of the relaxation scheme of section 2.5.1 is discussed that assures that a non-empty strictly feasible set is maintained at each instance; this characteristic is a necessary requirement for using interior-point methods for solving the relaxed problem (cf. section 5.1.1).

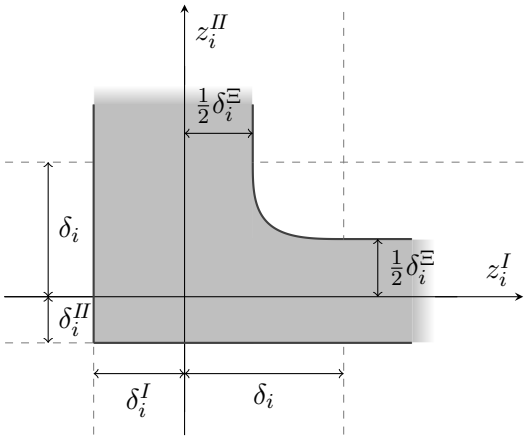


Figure 2.4: Illustration of the two-sided relaxation approach of [324] for one scalar complementarity condition using the smoothing function β_p.

The principal layout of this extension follows the line of [324] which is based on the work [72], but differences in the update procedure of the relaxation parameters resulting in different convergence properties are addressed, too.

The scheme introduced in [72] does not only relax the products $z_i^I z_i^{II} = 0$ of the complementarity conditions by $z_i^I z_i^{II} \leq \delta_i^{\Xi}$, but also the positivity constraints to $z_i^I \geq -\delta_i^I$ and $z_i^{II} \geq -\delta_i^{II}$ using suitable relaxation parameters $\delta_i^{\Xi},\ \delta_i^I,\ \delta_i^{II} \geq 0$. The idea of this two-sided relaxation approach is to reduce either the parameter δ_i^{Ξ} or at least one of the parameters δ_i^I and δ_i^{II} to zero while maintaining a strictly positive value for the other part. Following the line of [72, 324], slack variables are used in the formulation of the two-sided relaxed problem in order to simplify notation:

Definition 2.5.4. ***(Two-Sided Relaxed Problem** $\mathcal{R}(\underline{\delta}, \delta)$**)***

$$\begin{aligned} & \min_x \phi(x) \\ \textit{subject to} \quad & 0 = h(x), \\ & 0 = g(x) + \underline{s}^g, \\ & \delta^I = \underline{s}^I - z^I, \\ & \delta^{II} = \underline{s}^{II} - z^{II}, \\ & \delta^{\Xi} = \Xi(z^I, z^{II}, \delta) + \underline{s}^{\Xi}, \\ & 0 \leq \underline{s}, \end{aligned}$$

where $\underline{s} \in \mathbb{R}^{q+3s}_{\geq 0}$ *is the vector of the slack variables resulting from the concatenation of the slack variables of the individual parts*

$$\underline{s} = \left(\left(\underline{s}^g\right)^T, \left(\underline{s}^I\right)^T, \left(\underline{s}^{II}\right)^T, \left(\underline{s}^{\Xi}\right)^T \right)^T,$$

and $\underline{\delta} \in \mathbb{R}^{q+3s}$ *is the corresponding vector of the relaxation parameters for the two-sided scheme*

$$\underline{\delta} = \left(0, \left(\delta^I\right)^T, \left(\delta^{II}\right)^T, \left(\delta^{\Xi}\right)^T \right)^T.$$

The parameters and the resulting feasible region are illustrated in the figure 2.4. The following theorem states the relation between the stationary points of the relaxed problem $\mathcal{R}(\underline{\delta}, \delta)$ and the strongly stationary points of the MPEC 2.4.2:

Theorem 2.5.5.
Let the point x *with the multipliers* $\lambda,\ \mu,\ \psi^I$ *and* ψ^{II} *be a strongly stationary point of the MPEC 2.4.2 and let the relaxation parameters* $\underline{\delta}$ *and* δ *fulfill the conditions*

$$\begin{aligned} \delta_i^I = 0 \qquad & \textit{if} \quad \psi_i^I > 0, \\ \delta_i^I > 0 \qquad & \textit{if} \quad \psi_i^I \leq 0, \\ \delta_i^{II} = 0 \qquad & \textit{if} \quad \psi_i^{II} > 0, \\ \delta_i^{II} > 0 \qquad & \textit{if} \quad \psi_i^{II} \leq 0, \\ \delta_i^{\Xi} = 0 \qquad & \textit{if} \quad \psi_i^I < 0 \ \textit{ or } \ \psi_i^{II} < 0, \\ \delta_i^{\Xi} > 0 \qquad & \textit{if} \quad \psi_i^I \geq 0 \ \textit{ and } \ \psi_i^{II} \geq 0, \\ 0 \leq \delta_i \leq |z^I - z^{II}| \qquad & \textit{if} \quad \psi_i^I < 0 \ \textit{ or } \ \psi_i^{II} < 0, \\ \delta_i > 0 \qquad & \textit{if} \quad \psi_i^I \geq 0 \ \textit{ and } \ \psi_i^{II} \geq 0, \end{aligned}$$

for all $i = 1, \ldots, s$. *Then the following strict inequalities hold:*

$$\delta_i^I + \delta_i^{\Xi} > 0 \ \textit{ and } \ \delta_i^{II} + \delta_i^{\Xi} > 0, \quad i = 1, \ldots, s.$$

Additionally, if the slack variables and the Lagrange multipliers of the relaxed problem $\mathcal{R}(\underline{\delta}, \delta)$ fulfill the equalities

$$\begin{aligned}
\underline{s} &= \left(-g(x)^T, \left(z^I + \delta^I\right)^T, \left(z^{II} + \delta^{II}\right)^T, \left(\delta^\Xi - \Xi(z^I, z^{II}, \delta)\right)^T\right)^T, \\
\widehat{\psi}^I &= \left(\psi^I\right)_+, \\
\widehat{\psi}^{II} &= \left(\psi^{II}\right)_+, \\
\widehat{\psi}_i^\Xi &= \begin{cases} \left(-\frac{1}{2}\psi^I\right)_+ & \text{if} \quad i \in \left(\mathbb{A}_I \backslash \mathbb{A}_{II}\right)(x), \\ \left(-\frac{1}{2}\psi^{II}\right)_+ & \text{if} \quad i \in \left(\mathbb{A}_{II} \backslash \mathbb{A}_I\right)(x), \\ 0 & \text{if} \quad i \in \left(\mathbb{A}_I \cap \mathbb{A}_{II}\right)(x), \end{cases}
\end{aligned}$$

then $\left(x^T, \underline{s}^T\right)^T$ with the multipliers $\lambda,\ \mu,\ \widehat{\psi}^I,\ \widehat{\psi}^{II}$ and $\widehat{\psi}^\Xi$ is a stationary point of $\mathcal{R}(\underline{\delta}, \delta)$.

Furthermore, if x satisfies the MPEC-LICQ and the SSOSC, then the LICQ and the SOSC for $\mathcal{R}(\underline{\delta}, \delta)$ hold at $\left(x^T, \underline{s}^T\right)^T$, respectively.

Moreover, if the inequality $g(x)+\lambda > 0$ is fulfilled, then $\underline{s}$ and $\left(\lambda^T, \left(\widehat{\psi}^I\right)^T, \left(\widehat{\psi}^{II}\right)^T, \left(\widehat{\psi}^\Xi\right)^T\right)^T$ satisfy strict complementarity.

Proof. The verification of this theorem can be found in [324]. □

A result for the inverse implication is given in the following theorem:

Theorem 2.5.6.
Assume that the relaxation parameter $\delta > 0$ and the relaxation parameters $\underline{\delta}$ satisfying the strict inequalities

$$\delta_i^I + \delta_i^\Xi > 0 \quad \text{and} \quad \delta_i^{II} + \delta_i^\Xi > 0, \quad i = 1, \ldots, s$$

are given. Additionally, the point $\left(x^T, \underline{s}^T\right)^T$ with the Lagrange multipliers $\lambda,\ \mu,\ \widehat{\psi}^I,\ \widehat{\psi}^{II}$ and $\widehat{\psi}^\Xi$ has to be a stationary point of the relaxed problem $\mathcal{R}(\underline{\delta}, \delta)$ and the condition

$$\min\left\{z_i^I, z_i^{II}\right\} = 0 \quad \forall i = 1, \ldots, s$$

has to be fulfilled. Then x is a strongly stationary point of the MPEC 2.4.2 with the Lagrange multipliers λ, μ and ψ^I, ψ^{II} given by the equations

$$\begin{aligned}
\psi_i^I &= \widehat{\psi}_i^I - D_{z_i^I}\Xi_i(z_i^I, z_i^{II}, \delta_i)\widehat{\psi}_i^\Xi, \\
\psi_i^{II} &= \widehat{\psi}_i^{II} - \left(2 - D_{z_i^I}\Xi_i(z_i^I, z_i^{II}, \delta_i)\right)\widehat{\psi}_i^\Xi,
\end{aligned}$$

where $i = 1, \ldots, s$.

Proof. The theorem is proven in [324]. □

These two theorems are not only the basic motivation to solve a sequence of relaxed problems $\mathcal{R}(\underline{\delta}, \delta)$ for decreasing relaxation parameters $\underline{\delta}$ and δ, but they can be used to deduce suitable update strategies for the relaxation parameters. Since appropriate values of the parameters assuring $\min\{z^I, z^{II}\} = 0$ are not a priori known, it is reasonable to start with strictly positive values and analyze the solution in order to use suitable updates for the parameters.

Both [72] and [324] discuss their version of the two-sided relaxation in combination with an interior-point method; the basic idea is to adapt the relaxation parameters within the interior-point optimization resulting in specially-tailored interior-point methods. It is shown that in the course of the optimization the problem can be modified in such a way that under reasonable assumptions superlinear convergence in the vicinity of a strongly stationary point is maintained.

Our approach is not based on this simultaneous procedure, but each problem in the sequence of relaxed problems is solved by a standard interior-point method and the relaxation parameters for the next problem are updated by using the solution of the previous relaxed problem. Thus the standard convergence results for the interior-point method are applicable to the individual optimization run of one relaxed problem. Our update procedure for the relaxation parameters is discussed in the following; naturally, there are differences compared to the update procedures of [72] and [324], but they share the basic idea.

Assume that the j-th relaxed problem given by the relaxation parameters $\underline{\delta}^{(j)}$ and $\delta^{(j)}$ has been solved, which means that a stationary point

$$\left(\left(x^{(j)}\right)^T, \left(\underline{s}^{(j)}\right)^T\right)^T$$

with Lagrange multipliers

$$\lambda^{(j)},\ \mu^{(j)},\ \left(\widehat{\psi}^I\right)^{(j)},\ \left(\widehat{\psi}^{II}\right)^{(j)} \quad \text{and} \quad \left(\widehat{\psi}^{\Xi}\right)^{(j)}$$

is obtained. The goal is now to determine suitable parameters $\underline{\delta}^{(j+1)}$ and $\delta^{(j+1)}$, therefore the reduction factor $c_{\underline{\delta}} \in (0,1)$ und a tolerance for the complementarity condition $\epsilon_{tol} > 0$ is introduced. The two-sided relaxation approach allows the stationary point of $\mathcal{R}\left(\underline{\delta}^{(j)}, \delta^{(j)}\right)$ to either violate at least one of the positivity constraints or to violate the product constraint of the complementarity condition. Note that in both cases only the relaxation parameter corresponding to the violated constraint should be reduced. Consequently, the slack variables can be used to define the following update procedure:

Algorithm 2.5.7. (Relaxation Update)
Let $\underline{\delta}^{(j)}$ and $\delta^{(j)}$ be the current relaxation parameters and assume that a stationary point of the relaxed MPEC $\mathcal{R}(\underline{\delta}^{(j)}, \delta^{(j)})$ is given by $x^{(j)}$ and $\underline{s}^{(j)}$. Then the relaxation parameters are updated to $\underline{\delta}^{(j+1)}$ and $\delta^{(j+1)}$ by using the slack variable information:

$$\delta_i^{(j+1)} = c_{\underline{\delta}}\delta_i^{(j)} \qquad \text{if} \qquad \underline{s}_i - \delta_i < \epsilon_{tol},$$

$$\delta_i^{(j+1)} = \delta_i^{(j)} \qquad \text{if} \qquad \underline{s}_i - \delta_i \geq \epsilon_{tol},$$

for all indices $i = q+1, \ldots, q+3s$.

Optimal Control

Chapter 3

The goal of optimal control theory is to find the time-dependent state function $\overline{x}$ and control function $\overline{u}$ minimizing a given cost function $\overline{\phi}$. The functions have to fulfill the differential equation $\overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}(t))$ describing the dynamical properties of the system. Furthermore, equality and inequality constraints, such as boundary conditions, have to be met.

This leads to the following (classical) problem definition:

Definition 3.0.1. ***(Optimal Control Problem)***
Let $\overline{x} \in \mathcal{C}_c^1([0,1], \mathbb{R}^{\overline{n}})$ be the state function and $\overline{u} \in \mathcal{C}_p([0,1], \mathbb{R}^{\overline{m}})$ the control function. The following functions are assumed to be sufficiently smooth:

the terminal cost term	$\phi_b:$	$\mathbb{R}^{\overline{n}} \times \mathbb{R}^{\overline{n}}$	$\to$	$\mathbb{R}$,
the integral cost term	$\phi_I:$	$\mathbb{R}^{\overline{n}} \times \mathbb{R}^{\overline{m}}$	$\to$	$\mathbb{R}$,
the right hand side of the ODE	$\varphi:$	$\mathbb{R}^{\overline{n}} \times \mathbb{R}^{\overline{m}}$	$\to$	$\mathbb{R}^{\overline{n}}$,
the inequality constraints	$g:$	$\mathbb{R}^{\overline{n}} \times \mathbb{R}^{\overline{m}}$	$\to$	$\mathbb{R}^{\overline{l}}$,
the boundary conditions	$b:$	$\mathbb{R}^{\overline{n}} \times \mathbb{R}^{\overline{n}}$	$\to$	$\mathbb{R}^{\overline{c}}$.

The general optimal control problem is defined by:

$$\min_{\overline{x},\overline{u}} \overline{\phi}(\overline{x}, \overline{u}) := \phi_b(\overline{x}(0), \overline{x}(1)) + \int_0^1 \phi_I(\overline{x}(t), \overline{u}(t)) \, \mathrm{d}t$$

subject to $\overline{n}$ ordinary differential equations

$$\overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}(t)),$$

to $\overline{l}$ inequality constraints

$$g(\overline{x}(t), \overline{u}(t)) \leq 0$$

and to $\overline{c}$ boundary conditions

$$b(\overline{x}(0), \overline{x}(1)) = 0.$$

Remark 3.0.2. Note that the term *sufficiently smooth* is used for functions in problem 3.0.1 because the various statements of the following theorems are based on the continuity of the derivatives of different orders. For the most basic variational results presented in the following the functions have to be at least elements of $\mathcal{C}^1$ or $\mathcal{C}_p^1$.

Remark 3.0.3. We assume that the inequality constraints g are active only in the interior of the time interval, i.e.,

$$g_i(\overline{x}(0), \overline{u}(0)) < 0 \quad \text{and} \quad g_i(\overline{x}(1), \overline{u}(1)) < 0 \quad \text{for all} \quad i = 1, \ldots, \overline{l}.$$

The definition of the state $\overline{x}$ and the control $\overline{u}$ to be *piecewise* continuously differentiable and *piecewise* continuous allows for finitely many time instances where the system behavior can change, e.g., if the differential equation of the state changes. Such time instances can result, for example, if an inequality constraint becomes active.

The term classical optimal control problem is used for the above problem due to the designated spaces for the state variable $\overline{x} \in \mathcal{C}_c^1([0,1], \mathbb{R}^{\overline{n}})$ and the control variable $\overline{u} \in \mathcal{C}_p([0,1], \mathbb{R}^{\overline{m}})$. These smoothness assumptions allow to apply variational approaches in order to derive necessary optimality conditions (see section 3.1). However, the optimal control problem can be defined on larger classes for the state and the control: It is essential that the time-derivative of the state exists and fulfills the fundamental theorem of calculus, but a (Lagrange) integrable function, which needs not to be continuous, is still meaningful for the problem; this leads to the class of absolutely continuous functions $\mathcal{AC}([0,1], \mathbb{R}^{\overline{n}})$ which are introduced in section 3.4. Similarly, the space for the control can be defined by all integrable functions instead of all continuous ones. Consequently, the ordinary differential equation for the time-derivative of the state and the inequality constraints have to be fulfilled almost everywhere (a.e.), a standard construct in the context of Lagrange integration. Using these larger classes, other, more advanced techniques are need to derive necessary optimality conditions for the optimal control problem (e.g. see [58, 123]), but on the other hand these spaces allow to proof the existence of an solution under suitable assumptions (cf. section 3.4).

Remark 3.0.4. In most cases we will consider the boundary conditions to be of the type

$$\overline{x}(0) = \overline{x}^s, \overline{x}(1) = \overline{x}^e$$

for given function values $\overline{x}^s$ and $\overline{x}^e \in \mathbb{R}^{\overline{n}}$.

Remark 3.0.5. In optimal control theory different types of cost functions are considered and the total problem is named accordingly. The cost function of the above problem is a combination of a terminal cost term and an integral cost term; such problems are called *Bolza-problems*. If only terminal costs exist, the problem is called a *Lagrange-problem*. On the other hand, a *Mayer-problem* has only an integral cost term. Note that these distinctions are mainly for simplification of presentation and a problem of one type can be easily transformed into a problem of another problem class.

This optimal control problem can be solved by two different approaches: the direct methods first discretize and then optimize, while the indirect methods first optimize and then discretize. In the next section we will start with the discussion of the indirect approach while focusing on the necessary optimality conditions for the continuous problem. Then, in the section on direct methods the collocation approach used in this work to discretize the optimal control problem is discussed in detail and the convergence properties of the first-order necessary optimality conditions of the discretized problem towards those of the continuous one are addressed. The numerical results of the two approaches are compared for the simple problem of minimizing jerk. Finally, the existence of an optimal solution is discussed.

3.1 Indirect Methods

Indirect methods are based on the derivation of necessary conditions for problem 3.0.1. These conditions are used to obtain a complex boundary value problem, which can be solved by various numerical methods, e.g., the multiple shooting algorithm. More detail on the theory including sufficient conditions can be found in several textbooks on optimal control theory, see for example [44, 192, 287, 311].

In a first step we neglect the inequality constraints, thus the following problem is considered:

Definition 3.1.1. ***(Simplified Optimal Control Problem)***
Given the functions from definition 3.0.1, minimize

$$\overline{\phi}(\overline{x},\overline{u}) = \phi_b(\overline{x}(0),\overline{x}(1)) + \int_0^1 \phi_I(\overline{x}(t),\overline{u}(t))\ \mathrm{d}t$$

subject to $\overline{n}$ ordinary differential equations

$$\overline{x}'(t) = \varphi(\overline{x}(t),\overline{u}(t)), \tag{3.1}$$

and the boundary conditions

$$b(\overline{x}(0),\overline{x}(1)) = 0.$$

The following definition gives us a function combining the cost function $\overline{\phi}$ with the ordinary differential equations using the standard coupling approach of optimal control theory:

Definition 3.1.2. ***(Extended Cost Function)***
The definition of the extended cost function is based on the introduction of Lagrange multipliers or adjoint variables $\overline{\lambda} : [0,1] \to \mathbb{R}^{\overline{n}}$. These are assumed to be piecewise continuously differentiable and are used to couple the ordinary differential equations to the standard cost function $\overline{\phi}$, which yields the extended cost function $\overline{\phi}^+$:

$$\overline{\phi}^+(\overline{x},\overline{u},\overline{\lambda}) \quad := \quad \phi_b(\overline{x}(0),\overline{x}(1)) + \int_0^1 \phi_I^+\left(\overline{x}(t),\overline{u}(t),\overline{x}'(t),\overline{\lambda}(t)\right)\ \mathrm{d}t,$$

where the function

$$\phi_I^+\left(\overline{x}(t),\overline{u}(t),\overline{x}'(t),\overline{\lambda}(t)\right) := \phi_I(\overline{x}(t),\overline{u}(t)) + \overline{\lambda}(t)^T\left(\varphi(\overline{x}(t),\overline{u}(t)) - \overline{x}'(t)\right)$$

states the extended integral costs.

This extended cost function is closely related to the *Hamiltonian* of the problem which enables us to write the necessary conditions in a compact form:

Definition 3.1.3. ***(Hamiltonian)***
The Hamiltonian $H : \mathbb{R}^{\overline{n}} \times \mathbb{R}^{\overline{m}} \times \mathbb{R}^{\overline{n}} \times \mathbb{R} \to \mathbb{R}$ is defined by

$$\begin{aligned} H\left(\overline{x}(t),\overline{u}(t),\overline{\lambda}(t),\alpha\right) \quad := \quad & \alpha\phi_I^+\left(\overline{x}(t),\overline{u}(t),\overline{x}'(t),\overline{\lambda}(t)\right) \\ & - \overline{x}'(t)^T\nabla_{\overline{x}'}\phi_I^+\left(\overline{x}(t),\overline{u}(t),\overline{x}'(t),\overline{\lambda}(t)\right). \end{aligned}$$

Remark 3.1.4. An optimal control problem is called *autonomous* if neither the cost function nor the constraints depend explicitly on the time t, the independent variable. The problem 3.0.1 is autonomous by definition. It can be shown that the Hamiltonian is constant for autonomous problems, which is a good indicator of the obtained accuracy if the problem is solved numerically.

In the setting of 3.1.1 the following Hamiltonian results:

$$H\left(\overline{x}(t),\overline{u}(t),\overline{\lambda}(t),\alpha\right)=\alpha\phi_I(\overline{x}(t),\overline{u}(t))+\overline{\lambda}(t)^T\varphi(\overline{x}(t),\overline{u}(t)).$$

Using methods of variational calculus, it can be shown that the optimal control function $\overline{u}^*(t)$ minimizes the Hamiltonian, which is called the *Pontryagin minimum principle*. This leads to the following necessary conditions:

Theorem 3.1.5.
Let $(\overline{x}^*,\overline{u}^*)$ be a local optimal solution of problem 3.1.1. Then there exist $\alpha^*\geq 0$, $\overline{\lambda}^*\in\mathcal{C}_p^1(\mathbb{R}^{\overline{n}})$ and $\sigma^*\in\mathbb{R}^{\overline{c}}$ not all zero such that the following necessary optimality conditions hold (at time instances t where $\overline{u}^*$ is continuous)

(i) the adjoint differential equations

$$\left(\overline{\lambda}_k^*\right)'(t)=-\frac{\partial H}{\partial\overline{x}_k}\left(\overline{x}^*(t),\overline{u}^*(t),\overline{\lambda}^*(t),\alpha^*\right),\quad k=1,\ldots,\overline{n},\tag{3.2}$$

(ii) the optimality conditions

$$\frac{\partial H}{\partial\overline{u}_k}\left(\overline{x}^*(t),\overline{u}^*(t),\overline{\lambda}^*(t),\alpha^*\right)=0,\quad k=1,\ldots,\overline{m},\tag{3.3}$$

(iii) the transversality conditions

$$\left(\overline{\lambda}^*(0)\right)^T = -\alpha^*\frac{\partial\phi_b}{\partial\overline{x}(0)}\left(\overline{x}^*(0),\overline{x}^*(1)\right)-\left(\sigma^*\right)^T\frac{\partial b}{\partial\overline{x}(0)}\left(\overline{x}^*(0),\overline{x}^*(1)\right),\tag{3.4}$$

$$\left(\overline{\lambda}^*(1)\right)^T = +\alpha^*\frac{\partial\phi_b}{\partial\overline{x}(1)}\left(\overline{x}^*(0),\overline{x}^*(1)\right)+\left(\sigma^*\right)^T\frac{\partial b}{\partial\overline{x}(1)}\left(\overline{x}^*(0),\overline{x}^*(1)\right),\tag{3.5}$$

(iv) the Legendre-Clebsch condition

$$\nabla^2_{\overline{u}\overline{u}}H\left(\overline{x}^*(t),\overline{u}^*(t),\overline{\lambda}^*(t),\alpha^*\right)\quad \textit{positive semidefinite},\tag{3.6}$$

where time t fulfills $0\leq t\leq 1$.

Proof. Various proofs for these conditions exist, see for example [44]. □

Remark 3.1.6. Calculating the optimal control $\overline{u}^*$ by (3.3) and (3.6) yields a multi-point boundary value problem consisting of $2\overline{n}$ differential equations (3.2) and (3.1) and given boundary conditions b or transversality conditions (3.4) and (3.5).

The optimality condition (3.3) is locally uniquely dissolvable with respect to $\overline{u}$ for a regular $\nabla^2 H$. However, if the Hamiltonian depends only linearly on the control, the problem is insolvable as long as the controls are not constrained. In this case, a *switching function* being the derivative of the Hamiltonian with respect to the control is introduced and the sign of a non-zero value of this function determines whether the optimal control value equals the upper or lower bound, so-called *bang-bang control*. If the value of the switching function is zero, the *singular control* has to be determined from a total time-derivative of the switching function. However, in general it cannot be assured that a derivative exists with a non-zero derivative with respect to the control.

3.1.1 Inequality Constraints

To extend the problem defined in definition 3.1.1 to the general problem of the type given in definition 3.0.1, inequality constraints $g : \mathbb{R}^{\overline{n}} \times \mathbb{R}^{\overline{m}} \to \mathbb{R}^{\overline{l}}$ have to be considered:

$$g(\overline{x}(t), \overline{u}(t)) \leq 0, \quad t \in [0, 1].$$

Note that if one or more inequality constraints become active, one has to assume that the system is controllable at every instance, i.e., a feasible control exists for all $t \in [0, 1]$.

In literature one distinguishes between two types of inequality constraints, depending on the derivative with respect to the controls:

- *Control constraint* fulfilling the following causality:

$$g_i(\overline{x}(t), \overline{u}(t)) = 0 \Rightarrow \exists \overline{u}_k : \frac{\partial g_i}{\partial \overline{u}_k}(\overline{x}(t), \overline{u}(t)) \neq 0, \quad k = 1, \ldots, \overline{m}$$

- *State constraint* not explicitly depending on the controls:

$$\forall k = 1, \ldots, \overline{m} : \quad \frac{\partial g_i}{\partial \overline{u}_k}(\overline{x}(t), \overline{u}(t)) = 0.$$

If an inequality constraint is active on an interval, a differential algebraic equation (DAE) has to be fulfilled in addition to the standard ordinary differential equation for the state. Such an DAE changes the problem structure considerably with respect to stability properties and initial values and consequently approaches have been introduced in literature to handle the corresponding problems (see for example [123] and the references therein). To give an impression of central aspects like the order of a constraint and jumps in the adjoint variables, a few results for the case of one inequality constraint and one control are discussed.

3.1.1.1 Control Constraints

Assuming $\overline{l} = 1$ and $\overline{m} = 1$, the following Hamiltonian results

$$H\left(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t), \overline{\mu}(t), \alpha\right) \quad = \quad \alpha\phi_I(\overline{x}(t), \overline{u}(t)) + \overline{\lambda}(t)^T \varphi(\overline{x}(t), \overline{u}(t)) + \overline{\mu}(t) g(\overline{x}(t), \overline{u}(t)),$$

where $\overline{\mu} : \mathbb{R} \to \mathbb{R}$ is the Lagrangian multiplier for the inequality fulfilling the complementarity condition

$$\begin{aligned} \overline{\mu}(t) &= 0 \quad \text{if} \quad g(\overline{x}(t), \overline{u}(t)) < 0, \\ \overline{\mu}(t) &\geq 0 \quad \text{if} \quad g(\overline{x}(t), \overline{u}(t)) = 0. \end{aligned}$$

By using variational calculus it can be proven that in addition to the optimality conditions of theorem 3.1.5 *switching conditions* have to be fulfilled. Let t_1 and $t_2 \in (0, 1)$ be the time instances where the control constraint becomes active and inactive, respectively, the Hamiltonian has to be continuous at both times ($j = 1$ or $j = 2$):

$$\lim_{t \uparrow t_j} H\left(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t), \overline{\mu}(t), \alpha\right) = \lim_{t \downarrow t_j} H\left(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t), \overline{\mu}(t), \alpha\right).$$

Due to the assumption that the state $\overline{x}$ is continuous in $[0, 1]$, the adjoint states $\overline{\lambda}$ and the controls $\overline{u}$ have to be continuous at $t_{j,i}$, too.

3.1.1.2 State Constraints

In the context of state constraints we need to introduce the order of the state constraint:

Definition 3.1.7. ***(State Constraint Order)***
The functions g^k, $k = 1, \ldots, \overline{v}$, are defined recursively:

$$\begin{aligned} g^1(\overline{x}(t), \overline{u}(t)) &:= \frac{\mathrm{d}}{\mathrm{d}t} g(\overline{x}(t)) = \frac{\partial}{\partial \overline{x}} g(\overline{x}(t)) \; \varphi(\overline{x}(t), \overline{u}(t)), \\ g^i(\overline{x}(t), \overline{u}(t)) &:= \frac{\mathrm{d}}{\mathrm{d}t} g^{i-1}(\overline{x}(t)) = \frac{\partial}{\partial \overline{x}} g^{i-1}(\overline{x}(t), \overline{u}(t)) \; \varphi(\overline{x}(t), \overline{u}(t)). \end{aligned}$$

A state constraint is said to be of ***order*** $\overline{v}$ *if*

$$\frac{\partial}{\partial \overline{u}} g^k(\overline{x}(t), \overline{u}(t)) = 0 \quad \forall 0 \leq k \leq \overline{v}-1 \quad \textit{and} \quad \frac{\partial}{\partial \overline{u}} g^{\overline{v}}(\overline{x}(t), \overline{u}(t)) \neq 0. \tag{3.7}$$

Note that the ODE of the state $\varphi(\overline{x}(t), \overline{u}(t))$ and the constraint function $g(\overline{x}(t))$ have to be $(\overline{v}+1)$-times continuously differentiable, if a state constraint of order $\overline{v}$ is considered.

It is assumed that the state constraint becomes active only in the interior of the interval $[0, 1]$ and that only finitely many boundary segments exist. Different approaches exist to couple the inequality condition to the Hamiltonian to obtain the necessary conditions for the constrained problem. The direct coupling strategy used in the following theorem, where the function $g(\overline{x}(t))$ is directly attached to the Hamiltonian, has to be distinguished from the indirect coupling strategy, where $g^p(\overline{x}(t), \overline{u}(t))$ is added to the Hamiltonian.

Theorem 3.1.8. ***(Jacobson, Lele and Speyer [166])***
Let the Lagrange multiplier $\overline{\mu} : [0, 1] \to \mathbb{R}$ be used to couple g to the Hamiltonian:

$$H\left(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t), \overline{\mu}_i(t), \alpha\right) = \alpha \phi_I(\overline{x}(t), \overline{u}(t)) + \overline{\lambda}(t)^T \varphi(\overline{x}(t), \overline{u}(t)) + \overline{\mu}(t) g(\overline{x}(t)).$$

In addition to (3.2) - (3.6), an optimal solution $\left(\overline{x}^*, \overline{u}^*, \overline{\lambda}^*, \overline{\mu}^*, \alpha^*\right)$ has to fulfill

- *Sign conditions*

$$\begin{aligned} \overline{\mu}^*(t) &= 0 \quad \textit{for} \quad g(\overline{x}^*(t)) < 0, \\ \overline{\mu}^*(t) &\geq 0 \quad \textit{for} \quad g(\overline{x}^*(t)) = 0. \end{aligned} \tag{3.8}$$

- *Jump conditions: Let $t_j \in [0, 1]$ be the start or end point of a boundary segment or the time instance of a contact point for the inequality constraint, then the following conditions have to hold:*

$$\overline{\lambda}^*(t_j^+) = \overline{\lambda}^*(t_j^-) - \hat{\gamma} \; \nabla_{\overline{x}} g(\overline{x}^*(t_j), \overline{u}^*(t_j)), \tag{3.9}$$

$$H\left(\overline{x}^*(t_j^+), \overline{u}^*(t_j^+), \overline{\lambda}^*(t_j^+), \overline{\mu}^*(t_j^+), \alpha^*\right) = H\left(\overline{x}^*(t_j^-), \overline{u}^*(t_j^-), \overline{\lambda}^*(t_j^-), \overline{\mu}^*(t_j^-), \alpha^*\right) \tag{3.10}$$

$$\hat{\gamma} \geq 0. \tag{3.11}$$

The control for a boundary segment has to be calculated using $g^p(\overline{x}^*(t), \overline{u}^*(t)) = 0$ and the sign conditions (3.8) and (3.11) can be controlled a posteriori. To calculate the jump parameter $\hat{\gamma}$ the system of differential equations has to be extended by one trivial differential equation.

3.1.2 Numerical Methods for Boundary Value Problems

The necessary optimality conditions of the indirect approach for optimal control problems yield a boundary value problem, i.e., a problem of the general structure:

Definition 3.1.9. ***(Boundary Value Problem)***
Find a function $\overline{r} \in \mathcal{C}_p^1([0,1], \mathbb{R}^\upsilon)$ *that fulfills the ordinary differential equation*

$$\overline{r}'(t) = \zeta(\overline{r}(t)), \quad \forall t \in [0,1],$$

and the boundary conditions

$$b(\overline{r}(0), \overline{r}(1)) = 0$$

for a given function $b \in \mathcal{C}(\mathbb{R}^{\upsilon \times \upsilon}, \mathbb{R}^\upsilon)$.

Various numerical methods exist to solve boundary value problems, see for example [15]; we discuss in the following only those general properties that will also be relevant for the direct approach. Therefore, the related class of initial value problems is introduced:

Definition 3.1.10. ***(Initial Value Problem)***
Find a function $\overline{r} \in \mathcal{C}_p^1([0,1], \mathbb{R}^\upsilon)$ *that fulfills the ordinary differential equation*

$$\overline{r}'(t) = \zeta(\overline{r}(t)), \quad \forall t \in [0,1],$$

and the initial condition $\overline{r}(0) = \overline{r}^s$ *for a given initial value* $\overline{r}^s \in \mathbb{R}^\upsilon$.

A general class of numerical methods to solve initial value problems are the Runge-Kutta methods computing the value for the state at $t + \delta$ given the one at t by using nested evaluations of ζ:

$$\begin{aligned} \overline{r}(t+\delta) &= \overline{r}(t) + \delta \sum_{i=1}^{\kappa} \mathcal{B}_i \mathcal{K}_i, \\ \mathcal{K}_i &= \zeta\left(\overline{r}(t) + \delta \sum_{j=1}^{\kappa} \mathcal{A}_{ij} \mathcal{K}_j\right), \quad i = 1, \ldots, \kappa, \end{aligned}$$

where the characteristic coefficients of the Runge-Kutta method are given by the vector $\mathcal{B} \in \mathbb{R}^\kappa$ and the matrix $\mathcal{A} \in \mathbb{R}^{\kappa \times \kappa}$. Note that the method is called *explicit* if $\mathcal{A}$ has only zero entries in the upper triangular part, otherwise it is called *implicit* and determining $\mathcal{K}_i$ results in a nonlinear problem.

Consequently, the goal is to find a combination of $\mathcal{A}$ and $\mathcal{B}$ that guarantees that the numerical solution of the method converges towards the analytical solution of the initial value problem if the step sizes go to zero. To formally capture this notion, we introduce a strict partition Δ of the time interval $[0,1]$, i.e.,

$$\Delta := \{t_i \mid i = 1, \ldots, \nu\}$$

with $t_1 = 0$, $t_\nu = 1$ and $t_i < t_{i+1}$ for all $i = 1, \ldots, \nu$. The cardinality of Δ is given by $\nu \in \mathbb{N}$ and it is assumed that $\nu \geq 3$. The corresponding lengths of the subintervals are given by $\delta_i := t_{i+1} - t_i > 0$ for $i = 1, \ldots, \nu - 1$ and the maximum is denoted by δ_{max}.

The discretization error related to the discretization Δ is

$$\epsilon_\Delta : \Delta \to \mathbb{R}^\upsilon, \quad \epsilon_\Delta(t) = \overline{r}(t) - \widetilde{x}_\Delta(t),$$

where the mesh function $\widetilde{x}_\Delta(t)$ is obtained by recursively using the Runge-Kutta method for all $t \in \Delta$.

Definition 3.1.11. ***(Convergence of Discretized Problem)***
Assume that a function $\widetilde{x}_\Delta$ exists for all discretizations Δ with a sufficiently small δ_{max}. The family of mesh functions ***converges*** *towards a function $\overline{r} \in \mathcal{C}([0,1], \mathbb{R}^\upsilon)$ if the discretization error fulfills*

$$||\epsilon_\Delta||_\infty := \max_{t \in \Delta} |\epsilon_\Delta(t)| \to 0 \quad for \quad \delta_{max} \to 0.$$

The convergence is of order $\kappa > 0$ if

$$||\epsilon_\Delta||_\infty = \mathcal{O}\left((\delta_{max})^\kappa\right) \quad for \quad \delta_{max} \to 0.$$

The conditions on $\mathcal{A}$ and $\mathcal{B}$ resulting if a certain order of convergence has to be fulfilled are a direct consequence of Taylor's formula; for more details on general Runge-Kutta methods see for example [78, 137].

A basic technique to obtain useful Runge-Kutta methods is the collocation approach where a polynomial $\overline{q}$ is used to approximate the state function on a subinterval $[t, t+\delta]$. It is assumed that the polynomial fulfills the ordinary differential equation at least at κ prescribed time instances:

$$\overline{q}'(t + \mathcal{S}_i\delta) = \zeta(\overline{q}(t + \mathcal{S}_i\delta)).$$

Note that the assumption

$$0 \leq \mathcal{S}_0 < \ldots < \mathcal{S}_i < \ldots \mathcal{S}_\kappa \leq 1$$

is reasonable in this context. Using the Lagrange basis $\{\underline{\mathcal{L}}_j \mid j = 1, \ldots, \kappa\}$ of $\mathbb{P}^\upsilon_{\kappa-1}$, the space of all polynomials with degree smaller than κ, one obtains

$$\overline{q}'(t + t\delta) = \sum_{j=1}^{\kappa} \zeta(\overline{q}(t + \mathcal{S}_i\delta))\underline{\mathcal{L}}_j(t), \quad t \in [0,1].$$

Introducing the intermediate values $\mathcal{K}_i$, $i = 1, \ldots, \kappa$, by

$$\mathcal{K}_i = \zeta(\overline{q}(t + \mathcal{S}_i\delta))$$

and the coefficients of $\mathcal{A}$ and $\mathcal{B}$ by

$$\mathcal{A}_{ij} = \int_0^{\mathcal{S}_i} \underline{\mathcal{L}}_j(t)\ \mathrm{d}t, \quad i, j = 1, \ldots, \kappa,$$

and

$$\mathcal{B}_j = \int_0^1 \underline{\mathcal{L}}_j(t)\ \mathrm{d}t, \quad j = 1, \ldots, \kappa,$$

the collocation approach directly relates the vector $\mathcal{S}$ to a Runge-Kutta method.

Remark 3.1.12. Collocation approaches are also used in the context of the direct approach for optimal control problems (cf. section 3.2); even in this context, keeping the structure of the related Runge-Kutta method in mind proves to be useful.

Having a suitable family of Runge-Kutta methods, the initial value problem 3.1.10 can be solved if issues like step size control and order control are addressed (especially in the considered case of a non-stiff problem); for details we refer to [78, 137].

To solve the boundary value problem 3.1.9, several approaches are known (cf. [15]); we will summarize only the multiple shooting approach and the collocation approach related to the discussed methods for the initial value problems. Note that due to the inherent structure of a boundary value problem, a precise initial value is in general not specified, but the initial values have to be chosen in such a way that the conditions at the final time instance can be met.

The idea of multiple shooting [45, 220] is to divide the total interval $[0, 1]$ into subintervals $I_i := [t_i, t_{i+1})$, guess approximate initial values for each subinterval and then solve the resulting initial value problems. In general, non-zero differences between the final state of one subinterval and the guessed initial state of the next one result. However, a solution of the boundary value problem minimizes these differences. Consequently, methods of nonlinear optimization, e.g., a Newton method, can be used to determine better guesses for the initial values if the derivative information of the final states in dependence on the initial values is known; publications on determining the derivative information are for example [33, 131]. This iterative procedure has to be continued until the differences are below a user-given tolerance. For further information see [15].

Contrary to the multiple shooting method, the concept of the collocation approach avoids the distinction of integrating forward in time. Instead a global approximation of the solution is used by concatenating polynomials for specified subintervals and the goal is to fulfill the right-hand side of the ordinary differential equation at distinct time-instances. For guessed state information at the boundaries of the subintervals, this naturally results in differences between the derivatives of the polynomials and of the corresponding right-hand side values at these time-instances. This leads to a nonlinear optimization problem similar to the one of the multiple shooting method, but the derivative information can be accessed more directly. Further details on the collocation approach can, for example, be found in [15].

3.2 Direct Methods

Contrary to the approach of the indirect methods, where optimality conditions are derived in continuous spaces and the resulting boundary problem is then solved by a numerical integrator, the direct methods discretize the optimal control problem first and use optimization methods to determine an optimum in finite space.

Naturally, the question arises which strategies can be used to discretize the problem and which characteristics are of importance for the choice of the discretization method. Assuming that the optimal controls are known, the optimal control problem is a boundary value problem with a set of ordinary differential equations. Thus, the methods discussed in section 3.1.2 are candidates in this context; most prominently the multiple-shooting methods and the collocation methods. Following the line of von Stryk [330], we focus here on two collocation strategies.

Since the discretization step is a problem modification, it has to be analyzed how the solutions of the discretized problem are related to the originally continuous ones. A few publications discuss the relations for different settings using different techniques, e.g., [133, 136, 203, 330].

In [203] the optimal control problem is discretized in both its states and its controls using

the (forward) Euler method. A large set of assumptions is introduced in order to show the convergence properties: First, differentiability and continuity assumptions are made for the state, the control and the problem-describing functions. Second, assumptions on the coercivity of the Hamiltonian and the rank of the derivative of the constraints have to be fulfilled. Additionally, the solutions of some linearized equations and Ricatti equations have to fulfill certain properties. If all these assumptions are met and if the step size is sufficiently small, then the discrete state, control and adjoint variables differ from the continuous counterparts up to order one.

The application of general (higher order) Runge-Kutta methods to discretize the continuous optimal control problem is discussed in [136] and in addition to the well-known order conditions for Runge-Kutta methods [137] another set of conditions has to be fulfilled to guarantee higher order convergence in the adjoint variables, too. Note that the problem formulation in [136] allows to choose the control variables at each evaluation of the right-hand side of the differential equation independently of previous values, i.e., no coupling is considered which for example would result if the control is assumed to be piecewise linear. Combining the extended conditions for order κ with a smoothness and a coercivity assumption, it is shown that for a sufficiently small step size a local minimum for the discretized problem exists and that the convergence of the states, the controls and the adjoint variables is of order κ. The collocation strategies used in the following fulfill the extended set of conditions of the same order as the classical one.

3.2.1 Collocation Strategies

Two collocation strategies with different degrees of approximation are considered here to transform problem 3.0.1 into a standard nonlinear optimization problem. The strategy with the smaller degree of approximation is more robust with respect to the choice of the starting values for $\overline{x}$ and $\overline{u}$, while the other strategy approximates more accurately for good starting values.

Results from [330] are reviewed that show that the KKT-conditions applied to the discretized problem are equal up to $\mathcal{O}(h)$ to the first order necessary conditions of optimal control theory for problems with control constraints only.

In accordance with the notation of section 3.1.2, the following quantities are used to describe the strict partition

$$\Delta := \{t_i \mid i = 1, \ldots, \nu\}$$

of the total time interval $[0, 1]$:

$$0 = t_1 < t_2 < \ldots < t_\nu = 1.$$

We define the resulting subintervals by $I_i := [t_i, t_{i+1})$ for $i = 1, \ldots, \nu-2$ and by $I_i := [t_i, t_{i+1}]$ for $i = \nu - 1$. Consequently, the mid-point of each subinterval I_i is given by $t_{i+1/2} := \frac{1}{2}(t_i + t_{i+1})$ for $i = 1, \ldots, \nu - 1$. The following quantities describing the properties of the partition are needed to analyze the discretization strategies:

$$\delta_i(\Delta) := t_{i+1} - t_i, \ i = 1, \ldots, \nu - 1, \tag{3.12}$$

$$\delta_{max}(\Delta) := \max\{\delta_i(\Delta), i = 1, \ldots, \nu - 1\}, \tag{3.13}$$

$$\delta_{min}(\Delta) := \min\{\delta_i(\Delta), i = 1, \ldots, \nu - 1\}. \tag{3.14}$$

3.2.1.1 Piecewise Linear State and Piecewise Constant Control Variables

Given the strict partition Δ the control function $\overline{u}$ is approximated by a piecewise constant function $\widetilde{u} \in \mathbb{P}^{\overline{m}}_{1,\Delta}$ such that $\widetilde{u}$ is constant on each subinterval $[t_i, t_{i+1})$:

$$\widetilde{u}(t) = \widetilde{u}(t_{i+1/2}), \quad t \in I_i, \ i = 1, \ldots, \nu - 1. \tag{3.15}$$

The state function $\overline{x}$ is discretized by utilizing a continuous, piecewise linear function $\widetilde{x} \in \mathbb{P}^{\overline{n}}_{2,\Delta} \cap \mathcal{C}([0,1])$:

$$\widetilde{x}(t) = \widetilde{x}(t_i) + \frac{t - t_i}{\delta_i(\Delta)}(\widetilde{x}(t_{i+1}) - \widetilde{x}(t_i)), \quad t \in I_i, \ i = 1, \ldots, \nu - 1. \tag{3.16}$$

The original ordinary differential equation for the state vector is reduced to the condition

$$\widetilde{x}'(t_{i+1/2}) = \varphi(\widetilde{x}(t_{i+1/2}), \widetilde{u}(t_{i+1/2})), \ i = 1, \ldots, \nu - 1. \tag{3.17}$$

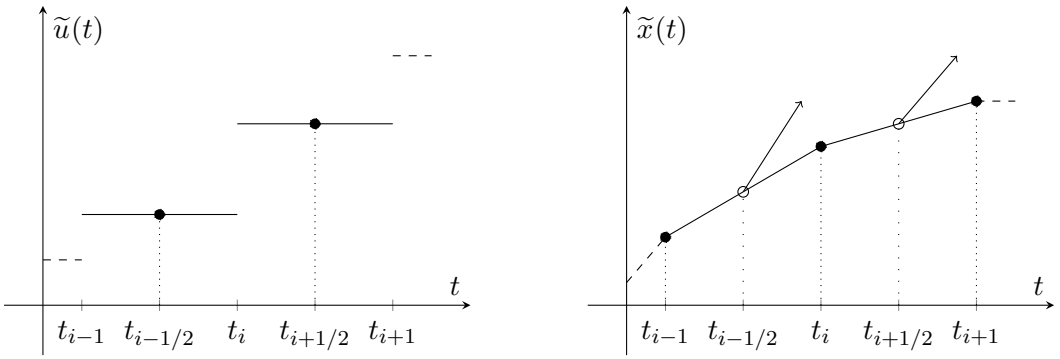


Figure 3.1: Schematic illustration of the approximation approach for state and control; the arrows indicate the slopes given by φ at the intermediate points for the condition (3.17).

The approximating functions $\widetilde{x}$ and $\widetilde{u}$ are utilized to transform the original infinite-dimensional optimal control problem 3.0.1 into a finite-dimensional nonlinear optimization problem:

$$\begin{aligned} & \min && \widetilde{\phi}(\widetilde{x}(t), \widetilde{u}(t)) \\ \text{subject to} \quad & && \widetilde{x}'(t_{i+1/2}) = \varphi(\widetilde{x}(t_{i+1/2}), \widetilde{u}(t_{i+1/2})), \ i = 1, \ldots, \nu - 1, \\ & && b(\widetilde{x}(0), \widetilde{x}(1)) = 0, \\ & && g(\widetilde{x}(t_i), \widetilde{u}(t_i)) \leq 0, \ i = 2, \ldots, \nu - 1, \end{aligned} \tag{3.18}$$

where $\widetilde{\phi}$ is a suitable discrete approximation of the continuous cost function $\overline{\phi}$. Note that $\widetilde{x}(t)$ is only required to fulfill the differential equation in the middle of each subinterval and that the inequality constraints are assumed to be inactive at start and end (cf. 3.0.1).

Several approaches exist to transform the original cost $\overline{\phi}$ for the continuous functions into the approximation $\widetilde{\phi}$, for example, by inserting the approximations $\widetilde{x}$ and $\widetilde{u}$ directly into $\overline{\phi}$ and calculating the resulting integral terms explicitly. In the context of the following analysis of pointwise convergence a linear approximation of the integral cost term proves to be suitable:

$$\widetilde{\phi}_I(t) = \phi_I(\widetilde{x}(t_i), \widetilde{u}(t_i)) + \frac{t - t_i}{\delta_i(\Delta)}(\phi_I(\widetilde{x}(t_{i+1}), \widetilde{u}(t_{i+1})) - \phi_I(\widetilde{x}(t_i), \widetilde{u}(t_i))),$$

where $t \in I_i$, $i = 1, \ldots, \nu - 1$. In consequence, the approximated cost function $\widetilde{\phi}$ reads:

$$\begin{aligned}\widetilde{\phi}(\widetilde{x}(t), \widetilde{u}(t)) &= \phi_b(\widetilde{x}(0), \widetilde{x}(1)) + \int_0^1 \widetilde{\phi}_I(t)\ \mathrm{d}t \\ &= \phi_b(\widetilde{x}(0), \widetilde{x}(1)) \\ &\quad + \sum_{i=1}^{\nu-1} \frac{\delta_i(\Delta)}{2} \left(\phi_I(\widetilde{x}(t_i), \widetilde{u}(t_i)) + \phi_I(\widetilde{x}(t_{i+1}), \widetilde{u}(t_{i+1}))\right).\end{aligned}$$

Our goal is now to rewrite the optimization problem (3.18) in the standard notation of nonlinear optimization to emphasize the finitely many optimization parameters. Therefore, we define the discretized state $x_{(i)} \in \mathbb{R}^{\overline{n}}$ and the discretized control $u_{(i)} \in \mathbb{R}^{\overline{m}}$, $i = 1, \ldots \nu$, by

$$\begin{aligned}x_{(i)} &:= \widetilde{x}(t_i),\ i = 1, \ldots, \nu, \\ u_{(i)} &:= \widetilde{u}(t_i),\ i = 1, \ldots, \nu - 1,\end{aligned}$$

and their respective concatenations by $x \in \mathbb{R}^{\overline{n}(\nu-1)}$ and $u \in \mathbb{R}^{\overline{m}(\nu-1)}$.
From these definitions follow the equations

$$\begin{aligned}\widetilde{x}(t_{i+1/2}) &= \frac{x_{(i+1)} - x_{(i)}}{2}, \\ \widetilde{u}(t_{i+1/2}) &= u_{(i)}, \\ \widetilde{x}'(t_{i+1/2}) &= \frac{x_{(i+1)} - x_{(i)}}{\delta_i(\Delta)}.\end{aligned}$$

Consequently, the optimization problem (3.18) can be rewritten as follows:

Definition 3.2.1. ***(Discrete Optimal Control Problem [Type I])***

$$\begin{aligned}&\min \phi(x, u) \\ &\textit{subject to} \\ &0 = -x_{(i+1)} + x_{(i)} + \delta_i(\Delta)\varphi\left(\frac{x_{(i+1)} - x_{(i)}}{2}, u_{(i)}\right), \quad i = 1, \ldots, \nu - 1, \\ &0 = b(x_{(1)}, x_{(\nu)}), \\ &0 \geq g\left(x_{(i)}, u_{(i)}\right), \quad i = 2, \ldots, \nu - 1.\end{aligned}$$

Here, the cost ϕ is determined by

$$\begin{aligned}\phi(x, u) &= \phi_b(x_{(1)}, x_{(\nu)}) \\ &\quad + \sum_{i=1}^{\nu-1} \frac{\delta_i(\Delta)}{2} \left(\phi_I(x_{(i)}, u_{(i)})) + \phi_I(x_{(i+1)}, u_{(i+1)})\right).\end{aligned}$$

Remark 3.2.2. In problem 3.2.1 the form of the differential equation for the state differs from the version of [330] by the factor $\delta_i(\Delta)$ and consequently, the resulting Lagrange multipliers are scaled versions of those of von Stryk. The advantage of our presentation form is that the relations between the continuous and the discrete Lagrange multipliers of the discrete optimal control problems (type I) and (type II) are identical (compare section 3.2.1.2).

The Lagrangian L of this problem is defined as:

$$\begin{aligned} L(x,u,\lambda,\mu) &:= \phi(x,u) \\ &+ \sum_{i=1}^{\nu-1} \lambda_{(i)}^T \left(\delta_i(\Delta) \varphi \left(\frac{x_{(i+1)} + x_{(i)}}{2}, u_{(i)} \right) - x_{(i+1)} + x_{(i)} \right) \\ &+ \sum_{i=2}^{\nu-1} \mu_{(i)}^T g \left(x_{(i)}, u_{(i)} \right) + \lambda_{(\nu)}^T b \left(x_{(1)}, x_{(\nu)} \right), \end{aligned}$$

where $\lambda_{(i)} \in \mathbb{R}^{\overline{n}}$, $i = 1, \ldots, \nu-1$, $\lambda_{(\nu)} \in \mathbb{R}^{\overline{c}}$ and $\mu_{(i)} \in \mathbb{R}^{\overline{l}}$, $i = 2, \ldots, \nu-1$, are the Lagrange multipliers of the discretized differential equation, the boundary conditions and the inequality constraints, respectively (cf. appendix A).

Theorem 3.2.3. ***(Pointwise Convergence Properties)***
Assume that a CQ is fulfilled for problem 3.2.1 and that in limit, i.e., $\delta_{max} \to 0$, the discrete values μ and λ result in a piecewise continuously differentiable and a piecewise twice continuously differentiable function, respectively. Considering only inequality constraints that ar control constraints, the KKT-conditions of problem 3.2.1 converge pointwise to the first order necessary conditions of the original optimal control problem 3.0.1. Furthermore, the Lagrangian L converges to the extended cost function $\overline{\phi}^+$.

Proof. A proof of this theorem for a problem similar to 3.0.1 can be found in [330]. Here we only state a slight variation of a selected part of the proof to show the basic relationship between the Lagrange multipliers of the discrete problem and the adjoint variables of the continuous problem, i.e., cf. [330] for the stationarity condition.

The derivative of the Lagrangian L with respect to a single state $x_{(j)}$, where $1 < j < \nu$, reads:

$$\begin{aligned} D_{x_{(j)}} L(x,u,\lambda,\lambda_{(\nu)},\mu) &= \frac{\delta_j(\Delta) + \delta_{j-1}(\Delta)}{2} D_{\overline{x}} \phi_I(x_{(j)}, u_{(j)}) \\ &\quad + \delta_{j-1}(\Delta) \left(\frac{\lambda_{(j-1)}}{2} \right)^T D_{\overline{x}} \varphi \left(\frac{x_{(j)} + x_{(j-1)}}{2}, u_{(j-1)} \right) \\ &\quad + \delta_j(\Delta) \left(\frac{\lambda_{(j)}}{2} \right)^T D_{\overline{x}} \varphi \left(\frac{x_{(j+1)} + x_{(j)}}{2}, u_{(j)} \right) \\ &\quad + \mu_{(j)}^T D_{\overline{x}} g \left(x_{(j)}, u_{(j)} \right) - \lambda_{(j-1)}^T + \lambda_{(j)}^T \\ &\stackrel{!}{=} 0. \end{aligned} \tag{3.19}$$

In addition to the approximating functions $\widetilde{x}$ for the state and $\widetilde{u}$ for the control depending on the vectors x and u, respectively, approximating functions for the adjoint variables are needed. The assumptions of the theorem guarantee that functions $\widetilde{\mu} \in \mathcal{C}_p^1([0,1], \mathbb{R}^{\overline{l}})$ and $\widetilde{\lambda} \in \mathcal{C}_p^2([0,1], \mathbb{R}^{\overline{n}})$ exist that fulfill the following conditions:

$$\begin{aligned} \widetilde{\lambda}(t_{j+1/2}) &:= \lambda_{(j)}, \quad j = 1, \ldots, \nu-1, \\ \widetilde{\mu}(t_j) &:= \frac{\mu_{(j)}}{\frac{1}{2}\delta_j(\Delta) + \frac{1}{2}\delta_{j-1}(\Delta)}, \quad j = 2, \ldots, \nu-1. \end{aligned}$$

First, Taylor's formula is used to rewrite the sum of the last two summands in (3.19):

$$\begin{aligned}\lambda_{(j)} - \lambda_{(j-1)} &= \widetilde{\lambda}(t_{j+1/2}) - \widetilde{\lambda}(t_{j-1/2}) \\ &= \widetilde{\lambda}(t_{j+1/2}) - \widetilde{\lambda}\left(t_{j+1/2} - \frac{\delta_j(\Delta) + \delta_{j-1}(\Delta)}{2}\right) \\ &= \frac{\delta_j(\Delta) + \delta_{j-1}(\Delta)}{2}\widetilde{\lambda}'(t_{j+1/2}) + \mathcal{O}(\delta_{max}^2).\end{aligned}$$

Using the Taylor approach in a similar manner, the other four summands of (3.19) result in:

$$\begin{aligned}&\frac{\delta_{j-1}(\Delta) + \delta_j(\Delta)}{2} D_{\overline{x}}\phi_I(x_{(j)}, u_{(j)}) + \mu_{(j)}^T D_{\overline{x}} g\left(x_{(j)}, u_{(j)}\right) \\ &+ \left(\frac{\lambda_{(j-1)}}{2}\right)^T D_{\overline{x}}\varphi\left(\frac{x_{(j)} + x_{(j-1)}}{2}, u_{(j-1)}\right) \\ &+ \left(\frac{\lambda_{(j)}}{2}\right)^T D_{\overline{x}}\varphi\left(\frac{x_{(j+1)} + x_{(j)}}{2}, u_{(j)}\right) \\ = \;& \frac{\delta_{j-1}(\Delta) + \delta_j(\Delta)}{2} D_{\overline{x}}\phi_I\left(\widetilde{x}(t_{j+1/2}), \widetilde{u}(t_{j+1/2})\right) \\ &+ \frac{\delta_{j-1}(\Delta) + \delta_j(\Delta)}{2}\widetilde{\mu}(t_{j+1/2})^T D_{\overline{x}} g\left(\widetilde{x}(t_{j+1/2}), \widetilde{u}(t_{j+1/2})\right) \\ &+ \frac{\delta_{j-1}(\Delta) + \delta_j(\Delta)}{2}\widetilde{\lambda}(t_{j+1/2})^T D_{\overline{x}}\varphi\left(\widetilde{x}(t_{j+1/2}), \widetilde{u}(t_{j+1/2})\right) \\ &+ \mathcal{O}(\delta_{max}^2).\end{aligned}$$

Since $\delta_j(\Delta) > 0$ for all j, the combination of the above equations yields

$$\begin{aligned}\widetilde{\lambda}'(t_{j+1/2})^T &= -D_{\overline{x}}\phi_I\left(\widetilde{x}(t_{j+1/2}), \widetilde{u}(t_{j+1/2})\right) \\ &\quad - \widetilde{\lambda}(t_{j+1/2})^T D_{\overline{x}}\varphi\left(\widetilde{x}(t_{j+1/2}), \widetilde{u}(t_{j+1/2})\right) \\ &\quad - \widetilde{\mu}(t_{j+1/2})^T D_{\overline{x}} g\overline{x}\left(\widetilde{x}(t_{j+1/2}), \widetilde{u}(t_{j+1/2})\right) \\ &\quad + \mathcal{O}(\delta_{max}).\end{aligned}$$

Consequently, for $\delta_{max} \to 0$ this equation converges to the adjoint differential equation (3.2)

$$\overline{\lambda}_k'(t) = -\nabla_{\overline{x}_k} H(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t)), \quad k = 1, \ldots, \overline{n}.$$

□

3.2.1.2 Piecewise Cubic State and Piecewise Linear Control Variables

Considering the strict partition Δ, a piecewise linear function $\widetilde{u} \in \mathbb{P}_{2,\Delta}^{\overline{m}}$ is utilized to approximate the control function $\overline{u}$ such that $\widetilde{u}$ is linear on each subinterval $[t_i, t_{i+1}]$:

$$\widetilde{u}(t) = u_{(i)} + \frac{t - t_i}{t_{i+1} - t_i}\left(u_{(i+1)} - u_{(i)}\right) \quad t \in I_i,\ i = 1, \ldots, \nu - 1,$$

where $u_{(i)} \in \mathbb{R}^{\nu}$ is the discrete control vector.

The approximation of the state function $\overline{x}(t)$ is chosen to be a continuously differentiable, piecewise cubic function $\widetilde{x}(t) \in \mathbb{P}^{\overline{n}}_{4,\Delta} \cap \mathcal{C}^1([0,1], \mathbb{R}^{\overline{n}})$

$$\widetilde{x}(t) = \sum_{j=0}^{3} \underline{\pi}_j^i \left(\frac{t - t_i}{t_{i+1} - t_i} \right)^j, \quad t \in I_i, \ i = 1, \ldots, \nu - 1, \tag{3.20}$$

satisfying the conditions

$$\widetilde{x}(t_i) = x_{(i)} \ \text{ and } \ \widetilde{x}'(t_i) = \varphi(\overline{x}(t_i), \overline{u}(t_i)), \ i = 1, \ldots, \nu. \tag{3.21}$$

These conditions uniquely define the coefficients $\underline{\pi}_j^i$ of the cubic polynomials:

$$\begin{aligned}
j = 0: \quad & \underline{\pi}_j^i = x_{(i)}, \\
j = 1: \quad & \underline{\pi}_j^i = \delta_i \varphi(x_{(i)}, u_{(i)}), \\
j = 2: \quad & \underline{\pi}_j^i = -3x_{(i)} - 2\delta_i \varphi\left(x_{(i)}, u_{(i)}\right) + 3x_{(i+1)} - \delta_i \varphi\left(x_{(i+1)}, u_{(i+1)}\right), \\
j = 3: \quad & \underline{\pi}_j^i = 2x_{(i)} + \delta_i \varphi\left(x_{(i)}, u_{(i)}\right) - 2x_{(i+1)} + \delta_i \varphi\left(x_{(i+1)}, u_{(i+1)}\right).
\end{aligned}$$

The differential equations of the original optimal control problem 3.0.1 will again be transformed into a differential condition for the middle of each interval:

$$\widetilde{x}'(t_{i+1/2}) = \varphi(\widetilde{x}(t_{i+1/2}), \widetilde{u}(t_{i+1/2})). \tag{3.22}$$

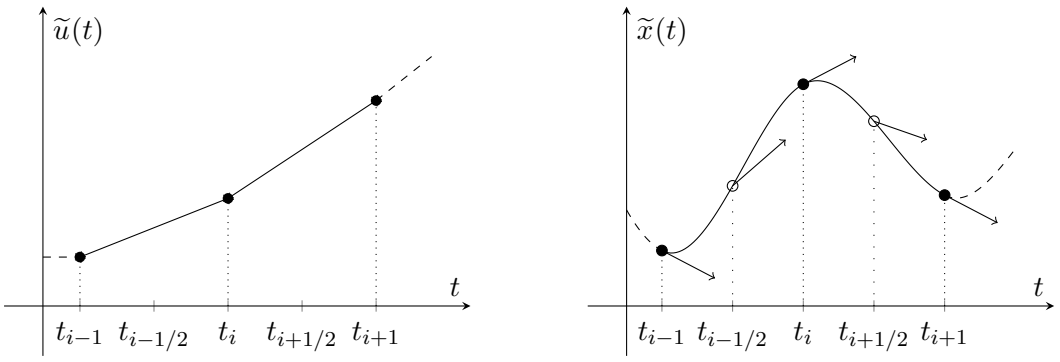


Figure 3.2: Schematic illustration of the approximation approach for state and control; the arrows indicate the slopes given by φ at the boundary points for the construction of the polynomials and at the intermediate points for the condition (3.22).

Remark 3.2.4. This type of approximation is first introduced by Hargraves and Paris [140] to solve optimal control problems and is closely related to the approach of [184] utilizing cubic approximations of the controls.

Remark 3.2.5. The combination of the conditions (3.21) and (3.22) yields a cubic collocation problem. If the interval $[t_i, t_{i+1}]$ is transformed onto the interval $[-1, 1]$ a cubic collocation using Lobatto-points (i.e., -1, 0 and 1) is obtained. In the interior of the interval this approximation is of order h^3. Higher orders of approximation could be realized by using other points for collocation. Gaussian-points, for example, yield an approximation of order h^6, but the approximation would only be continuous and not continuously differentiable at

the points t_i, $i = 2, \ldots, \nu - 1$. The advantage of Lobatto-points is the small number of collocation conditions which consequently leads to a smaller optimization problem. Note that only $2N + 1$ evaluations of the differential equations f are needed, while $3N$ evaluations would be needed if Gaussian-points were used, which yields a big advantage in computational expenses.

Consequently, the following nonlinear optimization problem is gained:

$$\begin{array}{lll} & \min & \widetilde{\phi}(\widetilde{x}(t), \widetilde{u}(t)) \\ \text{subject to} & & \widetilde{x}'(t_{i+1/2}) = \varphi(\widetilde{x}(t_{i+1/2}), \widetilde{u}(t_{i+1/2})),\ i = 1, \ldots, \nu - 1, \\ & & b(\widetilde{x}(0), \widetilde{x}(1)) = 0, \\ & & g(\widetilde{x}(t_i), \widetilde{u}(t_i)) \leq 0,\ i = 2, \ldots, \nu - 1. \end{array} \tag{3.23}$$

From the above definitions of $x_{(i)} = \widetilde{x}(t_i)$ and $u_{(i)} = \widetilde{u}(t_i)$, $i = 1, \ldots, \nu$, follows directly

$$\begin{aligned} \widetilde{u}(t_{i+1/2}) &= \frac{u_{(i+1)} + u_{(i)}}{2}, \\ \widetilde{x}(t_{i+1/2}) &= \frac{x_{(i+1)} + x_{(i)}}{2} + \frac{\delta_i(\Delta)}{8}\left(\varphi\left(x_{(i)}, u_{(i)}\right) - \varphi\left(x_{(i+1)}, u_{(i+1)}\right)\right), \\ \widetilde{x}'(t_{i+1/2}) &= \frac{3}{2}\frac{x_{(i+1)} - x_{(i)}}{\delta_i(\Delta)} - \frac{1}{4}\left(\varphi\left(x_{(i)}, u_{(i)}\right) + \varphi\left(x_{(i+1)}, u_{(i+1)}\right)\right), \\ i &= 1, \ldots, \nu - 1. \end{aligned}$$

Using the vectors x and u, problem (3.23) can be rewritten in the following way:

Definition 3.2.6. ***(Discrete Optimal Control Problem [Type II])***

$$\min \phi(x, u) \quad \textit{subject to}$$

$$\begin{aligned} 0 &= -x_{(i+1)} + x_{(i)} + \delta_i(\Delta)\left(\frac{1}{6}\varphi\left(x_{(i)}, u_{(i)}\right) + \frac{1}{6}\varphi\left(x_{(i+1)}, u_{(i+1)}\right)\right. \\ &\qquad \left. + \frac{2}{3}\varphi\left(x_{m,(i)}, \frac{u_{(i+1)} + u_{(i)}}{2}\right)\right), \quad i = 1, \ldots, \nu - 1, \\ 0 &= b\left(x_{(1)}, x_{(\nu)}\right), \\ 0 &\geq g\left(x_{(i)}, u_{(i)}\right), \quad i = 2, \ldots, \nu - 1, \end{aligned}$$

where for notational reasons the abbreviation $x_{m,(i)}$, $i = 1, \ldots, \nu - 1$, is used for the intermediate point:

$$x_{m,(i)} := \frac{x_{(i+1)} + x_{(i)}}{2} + \frac{\delta_i(\Delta)}{8}\left(\varphi\left(x_{(i)}, u_{(i)}\right) - \varphi\left(x_{(i+1)}, u_{(i+1)}\right)\right).$$

Remark 3.2.7. Again, the differential equations for the state are scaled versions of those in [330]. The form of our version is inspired by the standard form of a Runge-Kutta method and the main advantage is that the resulting relations between discrete and continuous Lagrange multipliers are identical to those of the discrete optimal control problem (type I).

The definition of the Lagrangian L of this problem reads:

$$\begin{aligned} L(x,u,\lambda,\mu) &:= \phi(x,u) + \lambda_{(\nu)}^T b\left(x_{(1)}, x_{(\nu)}\right) + \sum_{i=2}^{\nu-1} \mu_{(i)}^T g\left(x_{(i)}, u_{(i)}\right) \\ &\quad + \sum_{i=1}^{\nu-1} \lambda_{(i)}^T \left(-x_{(i+1)} + x_{(i)} + \delta_i(\Delta)\left(\frac{1}{6}\varphi\left(x_{(i)}, u_{(i)}\right)\right.\right. \\ &\quad \left.\left. + \frac{1}{6}\varphi\left(x_{(i+1)}, u_{(i+1)}\right) + \frac{2}{3}\varphi\left(x_{m,(i)}, \frac{u_{(i+1)} + u_{(i)}}{2}\right)\right)\right). \end{aligned}$$

The convergence properties of the approximation characterized in the following theorem are analyzed by von Stryk [330].

Theorem 3.2.8. ***(Pointwise Convergence Properties)***
Assume that a CQ is fulfilled and that in limit, i.e., $\delta_{max} \to 0$, the discrete values μ and λ result in a piecewise continuously differentiable and a piecewise twice continuously differentiable function, respectively. Considering only inequality constraints that are control constraints, the KKT-conditions of problem 3.2.6 converge pointwise to the first order necessary conditions of the original optimal control problem 3.0.1. Furthermore, the Lagrangian L converges to the extended cost function $\overline{\phi}^+$.

Proof. The basic techniques used by [330] to prove this theorem are identical to those of theorem 3.2.3 for the other, simpler approximation. Consequently, we refer to [330] and state only the points where our approach is slightly different:
In [330] the factor $\frac{2}{3}$ is needed to scale the extended cost function and to relate the discrete Lagrange multipliers to the continuous ones. This factor is a consequence of

$$\frac{\partial \widetilde{x}'(t_{i+1/2})}{\partial x_{(i)}} = -\frac{3}{2}.$$

and the used form of the equality constraints resulting from the ordinary differential equation of the state:

$$\widetilde{x}'(t_{i+1/2}) = \varphi(\widetilde{x}(t_{i+1/2}), \widetilde{u}(t_{i+1/2})),\ i = 1, \ldots, \nu - 1.$$

We use a reformulated version of the equality conditions to avoid this problem and guarantee consistency with other discretization strategies. Consequently, the relations between the discrete Lagrange multipliers λ and μ and the corresponding functions $\widetilde{\lambda}$ and $\widetilde{\mu}$ are in our case given by:

$$\begin{aligned} \widetilde{\lambda}(t_{j+1/2}) &:= \lambda_{(j)},\quad j = 1, \ldots, \nu - 1, \\ \widetilde{\mu}(t_j) &:= \frac{\mu_{(j)}}{\frac{1}{2}\delta_j(\Delta) + \frac{1}{2}\delta_{j-1}(\Delta)},\quad j = 2, \ldots, \nu - 1. \end{aligned}$$

□

3.3 Minimum Jerk Example

A simple numerical example is presented in this section to support the presented convergence results for both discretization strategies. The problem is taken from human arm motions where one of the central principles is the minimization of jerk of the hand, i.e., the third time derivative of the hand position is minimized for a task of moving the hand from a given

start to an end position (cf. section 6). The minimum jerk principle is normally discussed in two dimensions [104], but since the solution is just the combination of two solutions of one-dimensional problems, we consider here the one-dimensional problem. In the following the problem details are presented and the analytical solution is derived using the indirect approach, then the numerical results for both discretization strategies are compared to the analytical ones.

The hand position is denoted by $\overline{p}(t) \in \mathbb{R}$ for a time instance $t \in [0,1]$. The goal is to determine a curve $\overline{p}(\cdot)$ minimizing the integral of squared jerk, where jerk $\overline{j}$ is defined by

$$\overline{j}(t) := \frac{\mathrm{d}}{\mathrm{d}t^3}\overline{p}(t).$$

A common assumption in such physically-motivated problems is that all functions are sufficiently smooth. Since the standard form of an optimal control problem assumes that the differential equation for the state is of first order, the following equation results:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \overline{p}(t) \\ \overline{v}(t) \\ \overline{a}(t) \end{pmatrix} = \begin{pmatrix} \overline{v}(t) \\ \overline{a}(t) \\ \overline{j}(t) \end{pmatrix}, \tag{3.24}$$

where for $t \in [0,1]$ the velocity of the hand is denoted by $\overline{v}(t) \in \mathbb{R}$ and the acceleration of the hand by $\overline{a}(t) \in \mathbb{R}$. The natural choice for the control function $\overline{u}$ of the resulting optimal control problem is the hand jerk itself and the state vector is given by

$$\overline{x}(t) := (\overline{p}(t), \overline{v}(t), \overline{a}(t))^T.$$

To fully state the optimal control task, boundary conditions are needed:

$$\overline{x}(0) = \overline{x}^s \quad \text{and} \quad \overline{x}(1) = \overline{x}^e,$$

where the vectors $\overline{x}^s$ and $\overline{x}^e \in \mathbb{R}^3$ are given. Summing up, the following optimal control problem states the minimum jerk example:

Definition 3.3.1. ***(Minimum Jerk Problem)***

$$\min \int_0^1 \overline{u}(t)^2 \,\mathrm{d}t \qquad \textit{subject to} \qquad \overline{x}' = \begin{pmatrix} \overline{v}(t) \\ \overline{a}(t) \\ \overline{u}(t) \end{pmatrix},$$
$$\overline{x}(0) = \overline{x}^s, \quad \textit{and} \quad \overline{x}(1) = \overline{x}^e.$$

The Hamiltonian of this problem is given by

$$\begin{aligned} \overline{H}(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t)) \;&:=\; \overline{u}(t)^2 + \overline{\lambda}_1(t)\overline{v}(t) \\ &\quad + \overline{\lambda}_2(t)\overline{a}(t) + \overline{\lambda}_3(t)\overline{u}(t), \end{aligned}$$

where the Lagrange multipliers are

$$\overline{\lambda}(t) := \left(\overline{\lambda}_1(t), \overline{\lambda}_2(t), \overline{\lambda}_3(t)\right)^T.$$

The optimality condition $\frac{\mathrm{d}}{\mathrm{d}t}\overline{\lambda}(t) = -\nabla_{\overline{x}}\overline{H}(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t))$ yields the equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{\lambda}_1(t) = 0, \quad \frac{\mathrm{d}}{\mathrm{d}t}\overline{\lambda}_2(t) = -\overline{\lambda}_1(t), \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}t}\overline{\lambda}_3(t) = -\overline{\lambda}_2(t),$$

which lead to the following polynomials if the corresponding starting values $\overline{\lambda}_1^s$, $\overline{\lambda}_2^s$ and $\overline{\lambda}_3^s$ are known:

$$\begin{aligned}\overline{\lambda}_1(t) &= \overline{\lambda}_1^s,\\ \overline{\lambda}_2(t) &= -\overline{\lambda}_1^s t + \overline{\lambda}_2^s,\\ \overline{\lambda}_3(t) &= \frac{1}{2}\overline{\lambda}_1^s t^2 - \overline{\lambda}_2^s t + \overline{\lambda}_3^s.\end{aligned}$$

The second optimality condition $\nabla_{\overline{u}}\overline{H}(\overline{x}(t), \overline{u}(t), \overline{\lambda}(t)) = 0$ allows to determine the control in dependence on the Lagrange multipliers:

$$0 = 2\overline{u}(t) + \overline{\lambda}_3(t).$$

In consequence, the function of the hand position $\overline{p}(\cdot)$ is a polynomial of order five, because the ordinary differential equation (3.24) allows to simply integrate the second order polynomial for the control $\overline{u}(\cdot)$. Since the hand position and its derivatives have to fulfill the given boundary conditions $\overline{x}^s$ and $\overline{x}^e$, a linear equation system is obtained with a unique solution corresponding to the starting values of the Lagrange multipliers.

The following boundary conditions are used in the numerical computations:

$$\overline{x}^s = (0, -5, 0)^T \quad \text{and} \quad \overline{x}^e = (1, -5, 0)^T.$$

Consequently, the hand positions minimizing the squared jerk are given by the polynomial

$$\overline{p}^*(t) = 36t^5 - 90t^4 + 60t^3 - 5t, \quad t \in [0, 1].$$

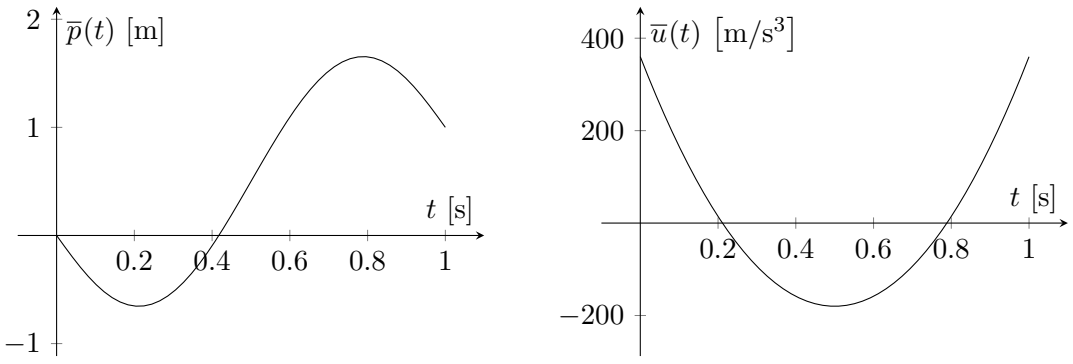


Figure 3.3: The hand path of the analytical solution (left) and the corresponding optimal control (right).

The minimum jerk problem 3.3.1 is now solved by using the two discretization strategies of the previous section. To analyze the convergence properties of both approaches, uniform time discretizations are used, i.e., the number of partitions ν directly corresponds to

$$\delta_{max} = \delta_i = \frac{1}{\nu - 1}, \quad \forall i = 1, \ldots, \nu - 1.$$

Choosing one of the two discretization types and a number of discretization intervals, the optimal values x^*, u^* and λ^* are determined by nonlinear optimization for the discretized

minimum jerk problem. To measure the accuracy of the obtained results we consider the root mean square error between the numerical results and the analytical solution; note that similar results are obtained if the max-norm is used. For example, consider the optimal positions $\overline{p}^*$ given in form of a polynomial and the corresponding discrete values p^*; the root mean square error normalized by the maximal value is then given by

$$\mathrm{RMSE}(\overline{p}^*, p^*) = \sqrt{\frac{\sum_{i=0}^{\nu-1} \left(\overline{p}^*(i\delta_i) - p^*\right)^2}{\max_{[0,1]}(\overline{p}^*)}},$$

where p^* is the i-th component of the vector p^*.

The numerical results for several uniform partitions of the time-interval $[0, 1]$ of different finenesses and for both discretization types discussed in the previous section are presented in figure (3.4).

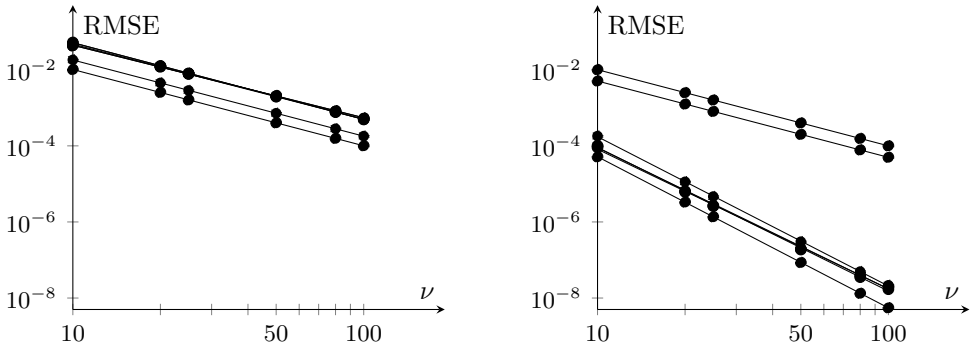


Figure 3.4: The root mean square errors between the analytical and numerical solutions of the normalized state, control and adjoint variables for discretizations of different fineness (left: discretization (type I), right: discretization (type II)). [•: numerical values, ——: lines of best fit]

Using the discretization of (type I) with piecewise linear state and piecewise constant control, one observes that the discrete values for all the states, the control and all the Lagrange multipliers converge to the analytical solutions with quadratic convergence in δ_{max}; curve fitting determines the mean of the exponents to be approximately 1.98 with a standard deviation of 0.05. Note that this observation is in accordance with the theory on Runge-Kutta methods stating that the implicit mid-point rule is of order two. On the other hand, the results for the discretization of (type II) show that the root mean square error is of order $\mathcal{O}\left((\delta_{max})^4\right)$ for all the state variables and the corresponding Lagrange multipliers $\overline{\lambda}_1$ and $\overline{\lambda}_2$, which, again, is the order of the Hermite-Simpson method as a Runge-Kutta method. In this case, the curve fitting determines the exponents to be approximately 3.81 in the mean with a standard deviation about 0.13. However, for the control function $\overline{u}$ and the related Lagrange multiplier $\overline{\lambda}_3$ we only find quadratic convergence, but the absolute values of the errors are smaller than those determined for the discretization of (type I).

This example does not allow to determine convergence characteristics with respect to the optimality conditions of the optimal control problem, because the discrete optimal values fulfill the discrete versions of these conditions for all analyzed uniform discretizations; i.e., the errors are smaller than the solution tolerance of the solver for the optimization problem.

3.4 Existence Results

The existence of a solution of a nonlinear optimal control problem is discussed in this section. Usually, further assumptions on the problem structure and especially on the control structure are needed to obtain such existence results. Here the existence theorem of Filippov using the compactness of certain sets is discussed and the implications for the optimal control problems presented in this work are specified.

As already mentioned when introducing the optimal control problem 3.0.1, the state space and the control space can easily be extended if continuity assumptions on the control and the derivative of the state are dropped. Consequently, absolutely continuous functions are introduced in the following and some of their properties are discussed, since in these spaces the existence of an solution can be proven.

Definition 3.4.1. ***(Absolutely Continuous Function)***
A function $z : [t_a, t_b] \to \mathbb{R}$ *with* $t_a < t_b$ *is* ***absolutely continuous*** *if to every* $\varepsilon > 0$ *there corresponds a* $\delta > 0$ *such that*

$$\sum_{i=1}^{\mathcal{N}} \left| z\left(t_b^{(i)}\right) - z\left(t_a^{(i)}\right) \right| \leq \varepsilon$$

for all finite systems of disjunct intervals $\left[t_a^{(i)}, t_b^{(i)}\right]$, $i = 1, \ldots, \mathcal{N}$, *in* $[t_a, t_b]$ *with*

$$\sum_{i=1}^{\mathcal{N}} \left| t_b^{(i)} - t_a^{(i)} \right| \leq \delta.$$

The set of all absolutely continuous functions $z : [t_a, t_b] \to \mathbb{R}$ *is denoted by* $\mathcal{AC}([t_a, t_b], \mathbb{R})$.

Absolutely continuous functions are discussed in various books on real analysis, e.g., [224, 265, 266]. Some central aspects are mentioned in the following corollary and lemma; for proofs of these statements see these books.

Corollary 3.4.2.

(i) If the function z *is in* $\mathcal{AC}([t_a, t_b], \mathbb{R})$*, then it is continuous.*

(ii) The sum, difference and product of two functions in $\mathcal{AC}([t_a, t_b], \mathbb{R})$ *is absolutely continuous, too. If the denominator of a quotient of two absolutely continuous functions is nonzero everywhere, then the quotient is also absolutely continuous.*

Note that all Lipschitz-continuous functions on an interval $[t_a, t_b]$ are examples of absolutely continuous functions. On the other hand, an example of a function being continuous on a closed interval and having a finite derivative almost everywhere but not being absolutely continuous is $t\cos(\pi/(2t))$. Note that the derivative of this function is not bounded.

Lemma 3.4.3.

(i) If and only if $z \in \mathcal{AC}([t_a, t_b], \mathbb{R})$, then z is differentiable almost everywhere in $[t_a, t_b]$, $z' \in \mathcal{L}^1([t_a, t_b], \mathbb{R})$ and the following equation holds true for all $t \in [t_a, t_b]$:

$$z(t) - z(t_a) = \int_{t_a}^{t} z'(\bar{t}) \, \mathrm{d}\bar{t}.$$

(ii) If the derivative z' of $z \in \mathcal{AC}([t_a, t_b], \mathbb{R})$ is zero almost everywhere in $[t_a, t_b]$, then z is constant.

Remark 3.4.4. The properties of absolutely continuous functions listed above motivate the consideration of $\mathcal{AC}([0, t_f], \mathbb{R}^{\overline{n}})$ as the space for the state of the optimal control problem. Combined with the assumption that the controls are measurable and the integral cost term is (Lagrange)-integrable, a rather general optimal control problem is obtained. Note that in literature related settings are also common; for example, the state could be of bounded variation only (see for example [224] for a definition of bounded variation and note that all absolutely continuous functions have a bounded variation) or in addition to absolute continuity a (a.e.) bounded derivative could be demanded.

The following existence theorem can be found in [58], but it is presented in a slightly modified form because only autonomous systems are of interest here; several sets have to be introduced for the theorem: Let $t_f > 0$ be the final time, $\mathbb{Y} = [0, t_f] \times \mathbb{X} \subset \mathbb{R}^{\overline{n}+1}$ be a subset of image space for time-state-combinations and $\mathbb{U} \subset \mathbb{R}^{\overline{m}}$ be a subset of the image space for control functions, i.e., a feasible combination of a state $\overline{x}$ and a control $\overline{u}$ is assumed to fulfill

$$(t, \overline{x}(t)) \in \mathbb{Y}, \quad \overline{u}(t) \in \mathbb{U}, \quad \text{for} \quad t \in [0, t_f] \text{ a.e.}$$

Denote by $\mathbb{M} = \mathbb{X} \times \mathbb{U}$ the Cartesian product of $\mathbb{X}$ and $\mathbb{U}$ and consequently, the function φ is a given mapping from $\mathbb{M}$ to $\mathbb{R}^{\overline{n}}$. For every $x \in \mathbb{X}$ let $\widetilde{\mathbb{Q}}(x) \subset \mathbb{R}^{\overline{n}}$ be defined by

$$\widetilde{\mathbb{Q}}(x) = \{\varphi(x, u) \mid u \in \mathbb{U}\}.$$

Furthermore, the extended sets $\mathbb{Q}(x) \subset \mathbb{R} \times \widetilde{\mathbb{Q}}(x)$ are introduced by

$$\mathbb{Q}(x) = \{(q_1, \varphi(x, u)) \mid q_1 \geq \phi_I(x, u), u \in \mathbb{U}\}.$$

Let $\mathbb{B}$ be the set of feasible boundary values, i.e.,

$$\forall \ \left(x^{(1)}, x^{(2)}\right) \in \mathbb{B} : \quad b\left(x^{(1)}, x^{(2)}\right) = 0,$$

and consequently, the terminal cost ϕ_b is a given mapping from $\mathbb{B}$ to $\mathbb{R}$.

This leads to the following formulation of an optimal control problem for a state $\overline{x} \in \mathcal{AC}([0, t_f], \mathbb{R}^{\overline{n}})$ and a measurable control $\overline{u} : [0, t_f] \to \mathbb{R}^{\overline{m}}$:

$$\min_{\overline{x}, \overline{u}} \overline{\phi}(\overline{x}, \overline{u}) = \phi_b(\overline{x}(0), \overline{x}(t_f)) + \int_0^{t_f} \phi_I(\overline{x}(t), \overline{u}(t)) \, \mathrm{d}t$$

subject to $\overline{n}$ ordinary differential equations

$$\overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}(t)), \quad t \in [0, t_f] \text{ a.e.},$$

to the boundary conditions

$$(\overline{x}(0), \overline{x}(t_f)) \in \mathbb{B},$$

to the integrability condition

$$\phi_I(\overline{x}(\cdot), \overline{u}(\cdot)) \quad \text{(Lagrange-) integrable in} \quad [0, t_f]$$

and to the set constraints

$$(t, \overline{x}(t)) \in \mathbb{Y}, \quad \overline{u}(t) \in \mathbb{U}, \quad t \in [0, t_f] \text{ a.e.}$$

Remark 3.4.5. Comparing the above defined optimal control problem with the original one 3.0.1, two details have to be mentioned in addition to the different spaces for the state and the control. First, a final time t_f is introduced to account for solutions with different lengths. In the classical setting this problem is solved by using a time transformation, but in the context of an existence analysis such an technique only obscures the underlying structure. Second, the inequality constraints are no longer given in explicit form but in the implicit form of the feasible set $\mathbb{Y}$, which means that in the presented form the inequality constraints are limited to pure state constraints (see [58] for a more general setting).

Theorem 3.4.6. ***(Filippov Existence Theorem)***
Let $\mathbb{Y}$ be compact, $\mathbb{B}$ closed, $\mathbb{M}$ compact, ϕ_b lower semicontinuous on $\mathbb{B}$ and ϕ_I, φ continuous on $\mathbb{M}$. Furthermore, assume that almost all sets $\mathbb{Q}(x)$, $x \in \mathbb{X}$, are convex. Then the cost function $\overline{\phi}(\overline{x}, \overline{u})$ has an absolute minimum if a feasible $(\overline{x}, \overline{u})$-pair exists.

Proof. The proof using the approach of orientor fields can be found in [58]. The basic principle of orientor fields is to consider the constraint $x' \in \widetilde{\mathbb{Q}}(x)$ in the formulation of the optimal control problem. Given the assumptions of the theorem, the representation using orientor fields is equivalent to the standard problem formulation combining an ODE for the states and the bounds for the controls.

A central idea of the proof is to show that a minimizing sequence is uniformly bounded and equicontinuous and then to use the theorem of Arzela-Ascoli to obtain uniform convergence. Finally, it is proven that the limit guarantees absolute continuity of the state and measurability of the control and that it is actually an absolute minimum. □

Remark 3.4.7. Note that the assumptions of $\mathbb{Y}$ and $\mathbb{M}$ being compact are rather strong and the proof actually uses the following weaker property: The minimizing sequence lies in a bounded closed subset $\mathbb{Y}_1 \subset \mathbb{Y}$ and $\mathbb{M}_1 = \mathbb{X}_1 \times \mathbb{U}$, the part of $\mathbb{M}$ corresponding to $\mathbb{X}_1$, is compact.

Following the line of [58], the weaker assumptions given in the previous remark can be guaranteed by various sets of conditions. We make use of the following ones:

Assumption 3.4.8.

(i) The set $\mathbb{Y}$ is closed and the final time t_f is bounded from above, i.e., a constant $\widehat{t}_f > 0$ exists such that $t_f \leq \widehat{t}_f$.

(ii) For every constant $c_{\overline{x}} > 0$ the set $\{(x, u) \in \mathbb{M} \mid ||x|| \leq c_{\overline{x}}\}$ is compact.

(iii) A compact subset $\mathbb{C}$ of $\mathbb{Y}$ exists such that every feasible trajectory $\overline{x}$ has a non-empty intersection set with $\mathbb{C}$.

(iv) A constant $c_\varphi \geq 0$ exists such that $||\varphi(x,u)|| \leq c_\varphi\,(||x||+1)$ for all $(x,u) \in \mathbb{M}$.

Lemma 3.4.9. *If the assumptions 3.4.8 are fulfilled, a bounded closed subset $\mathbb{Y}_1 \subset \mathbb{Y}$ containing the feasible states exists and $\mathbb{M}_1 = \mathbb{X}_1 \times \mathbb{U}$ is compact.*

Proof. Assumption (iv) guarantees that a constant $\widehat{c}_\varphi \geq 0$ exists such that $\sum_{i=1}^{\overline{n}} x_i\varphi_i(x,u) \leq \widehat{c}_\varphi\left(||x||^2+1\right)$, since with $\widehat{c}_\varphi \geq 2\overline{n}c_\varphi$ holds

$$\begin{aligned}\sum_{i=1}^{\overline{n}} x_i\varphi_i(x,u) &\leq \overline{n}||x||\;||\varphi(x,u)|| \\ &\leq \overline{n}||x||\;c_\varphi\,(||x||+1) \\ &\leq 2\overline{n}c_\varphi\left(||x||^2+1\right) \\ &\leq \widehat{c}_\varphi\left(||x||^2+1\right).\end{aligned}$$

This can be used to show that a constant $\widehat{c}_{\overline{x}} > 0$ exists such that every feasible trajectory lies in the set $\left\{(t,x) \mid 0 \leq t \leq \widehat{t}_f,\ ||x|| \leq \widehat{c}_{\overline{x}}\right\}$:
Consider the function $z : [0,t_f] \to \mathbb{R}$ for a given feasible state $\overline{x}$ and control $\overline{u}$ defined by

$$z(t) := ||\overline{x}(t)||^2 + 1.$$

It follows that $z(t) \geq 1$ for all $t \in [0,t_f]$ and

$$\frac{\mathrm{d}}{\mathrm{d}t}z(t) = 2\sum_{i=1}^{\overline{n}} \overline{x}_i(t)\varphi_i(\overline{x}(t),\overline{u}(t)) \leq 2\widehat{c}_\varphi z(t).$$

By assumption (iii) a time instance $t^* \in [0,t_f]$ exists such that $(t^*,\overline{x}(t^*)) \in \mathbb{C}$. Since $\mathbb{C}$ is compact, a constant $\widetilde{c}_{\overline{x}} > 0$ exists such that $||x|| \leq \widetilde{c}_{\overline{x}}$ for all x with $(t,x) \in \mathbb{C}$. Consequently, by integration from t^* to t follows

$$1 \leq z(t) \leq z(t^*)\exp(2\widehat{c}_\varphi|t-t^*|) \leq (||\overline{x}(t^*)||^2+1)\exp(2\widehat{c}_\varphi|t-t^*|) \leq (\widetilde{c}_{\overline{x}}^2+1)\exp(2\widehat{c}_\varphi\widehat{t}_f)$$

and finally

$$||\overline{x}(t)|| \leq \sqrt{z(t)} \leq \sqrt{\widetilde{c}_{\overline{x}}^2+1}\exp(\widehat{c}_\varphi\widehat{t}_f) =: \widehat{c}_{\overline{x}}.$$

Using assumption (i) the set $\mathbb{X}_1 := \{x \in \mathbb{X} \mid ||x|| \leq \widehat{c}_{\overline{x}}\}$ is closed and bounded; furthermore, we have shown that $\mathbb{X}_1$ contains all feasible state values. The same holds true for $\mathbb{Y}_1 := [0,\widehat{t}_f] \times \mathbb{X}_1$, which proves the first part of the lemma.
The second part follows by combining the assumption (ii) with the set $\mathbb{M}_1 = \mathbb{X}_1 \times \mathbb{U}$ which yields that $\mathbb{M}_1$ is compact.
This proof is based on ideas of [58]. □

Several of the assumptions needed in Filippov's existence theorem are directly fulfilled by the optimal control problems discussed in this work. For instance, the set of possible boundary values $\mathbb{B}$ is closed because we consider $\mathbb{B}$ to be the null set of a continuous function b. Furthermore, all cost function ϕ_b and ϕ_I and all right-hand sides of the ordinary differential equations for the state variables φ are continuous in our problems. Note that all problems describe human motions and consequently, it can easily be assumed that the final time is bounded from above.

Since all state constraints are formulated as inequality constraints using continuous functions, the set $\mathbb{X}$ is closed, which yields that $\mathbb{Y}$ is closed, too; this guarantees assumption (i) of 3.4.8. The set of feasible controls $\mathbb{U}$ is given in our examples by upper and lower bounds and therefore $\mathbb{U}$ is compact. From this follows that assumption (ii) is fulfilled, because the relevant set is the Cartesian product of two compact sets. Because all state variables are related to physical quantities, bounds for the initial values that are not fixed by prescribed values can be deduced in order to obtain meaningful quantities and consequently, the compact subset $\mathbb{C}$ of assumption (iii) is given by the set of feasible initial values.

The following lemma addresses one of the remaining assumptions:

Lemma 3.4.10. *Let the set $\mathbb{U}$ be convex, the integral cost function ϕ_I be convex with respect to the controls and the function φ be linear with respect to the controls. Then all sets $\mathbb{Q}(x)$ for $x \in \mathbb{X}$ are convex.*

Proof. Since the function φ is linear with respect to the controls, the following notation is introduced:

$$\varphi(x,u) = \overline{A}(x) + \overline{B}(x)u.$$

Let $q^{(1)}$ and $q^{(2)}$ be elements of $\mathbb{Q}(x)$ for a given $x \in \mathbb{X}$. Consider the convex combination with the parameter $\alpha \in [0,1]$:

$$\begin{aligned}\alpha q^{(1)} + (1-\alpha)q^{(2)} &= \alpha\left(q_1^{(1)}, \varphi\left(x, u^{(1)}\right)^T\right)^T + (1-\alpha)\left(q_1^{(2)}, \varphi\left(x, u^{(2)}\right)^T\right)^T \\ &= \left(\alpha q_1^{(1)} + (1-\alpha)q_1^{(2)}, \alpha\varphi\left(x, u^{(1)}\right)^T + (1-\alpha)\varphi\left(x, u^{(2)}\right)^T\right)^T.\end{aligned}$$

Due to the convexity of ϕ_I with respect to the controls, the first element fulfills

$$\begin{aligned}\alpha q_1^{(1)} + (1-\alpha)q_1^{(2)} &\geq \alpha\phi_I\left(x, u^{(1)}\right) + (1-\alpha)\phi_I\left(x, u^{(2)}\right) \\ &\geq \phi_I\left(x, \alpha u^{(1)} + (1-\alpha)u^{(2)}\right).\end{aligned}$$

Additionally, the following equations hold true for the other elements:

$$\begin{aligned}\alpha\varphi(x,u^{(1)}) + (1-\alpha)\varphi(x,u^{(2)}) &= \alpha\left(\overline{A}(x) + \overline{B}(x)u^{(1)}\right) + (1-\alpha)\left(\overline{A}(x) + \overline{B}(x)u^{(2)}\right) \\ &= \overline{A}(x) + \overline{B}(x)\left(\alpha u^{(1)} + (1-\alpha)u^{(2)}\right) \\ &= \varphi\left(\alpha u^{(1)} + (1-\alpha)u^{(2)}\right).\end{aligned}$$

Consequently, the convex combination $\alpha q^{(1)} + (1-\alpha)q^{(2)} \in \mathbb{Q}(x)$ if the combination

$$\alpha u^{(1)} + (1-\alpha)u^{(2)} \in \mathbb{U}.$$

The latter is a consequence of the convexity of $\mathbb{U}$. □

Note that all examples discussed in this work fulfill all three assumptions of this lemma: The set $\mathbb{U}$ is convex due to the simple upper and lower bounds. All considered ordinary differential equations are linear with respect to the controls and the controls appear in the integral cost terms in a convex manner (if they influence the cost term at all).

The remaining assumption which has to be fulfilled in order to apply the existence theorem is assumption (iv) of 3.4.8. For some dynamics this assumption is directly fulfilled as a

consequence of bounded controls, for example in case of linear dynamics (e.g., compare the linear car model in section 7.3.2):
Let $\varphi(x,u) := \overline{A}x + \overline{B}u$ for constant matrices $\overline{A} \in \mathbb{R}^{\overline{n}\times\overline{n}}$ and $\overline{B} \in \mathbb{R}^{\overline{n}\times\overline{m}}$:

$$||\varphi(x,u)|| \leq ||\overline{A}||\ ||x|| + ||\overline{B}||\ ||u|| \leq \max\left\{||\overline{A}||, ||\overline{B}||\max_{u\in\mathbb{U}}\{||u||\}\right\}(||x|| + 1)\,.$$

In case of nonlinear dynamics one might need further assumptions to assure this condition. If for example the nonlinear models of the human arm are considered (cf. section 6.5), critical cases, i.e., cases where the inequality (iv) is violated, correspond for instance to arm configurations with a singular mass matrix. Note that in most applications precisely these configurations have to be avoided due to the related physical properties. Consequently, it can be assumed that the pure state constraint is already added to the feasible set $\mathbb{X}$ in the problem formulation. Other cases can be solved if bounds are introduced for selected state variables; note that the biological and technical background of the discussed problems naturally results in bounds for most state variables. If bounds exist for all state variables, the feasible set $\mathbb{Y}$ is compact and Filippov's existence theorem can be applied directly.

Remark 3.4.11. If inequalities constraining both the state and the control variables are considered, the presented approach has to be extended to state-dependent sets of feasible controls $\mathbb{U}(x)$; see [58] for details. However, to guarantee the convexity and compactness properties needed in the existence theorem, it has to be assumed that such inequality constraints guarantee the convexity and the compactness of all sets $\mathbb{U}(x)$.

Inverse Optimal Control

Chapter 4

In this chapter we address a generalization of the bilevel programs of chapter 2 where (at least) the lower level program is an optimal control problem (cf. chapter 3). In our terminology such a problem is called a *bilevel optimal control problem*, but in literature the term *bilevel dynamic problem* is alternatively used. We are interested in problems of a special subclass where the upper level state influences only the lower level cost function and the upper level cost function is a distance measure between given data and the optimal lower level state. Such problems result if one assumes that a system is controlled optimally with respect to an unknown cost function and the goal is to determine the cost function within a given parameterized family of cost functions that solves the inverse problem of minimizing the distance to characteristic data of the system. Problems of this subclass are named *inverse optimal control problems* in this work and for details on the problem structure and a summary of our solution strategy see section 4.1.

In section 4.2 the state of the art in bilevel optimal control and inverse optimal control is discussed and the connections to related research topics like *differential games* are presented. Two distance measures used as upper level cost functions are introduced in section 4.3. The structure of the discretized bilevel optimal control problem and the corresponding reformulated one-level problem are discussed in section 4.4.

4.1 Inverse Optimal Control Problem

In the following the inverse optimal control problems considered in this work are defined and the general layout of our solution strategy is summarized by combining the presented concepts of the previous chapters. Further details on the actual realization of this strategy can be found in chapter 5.2.

First, the lower level problem is characterized as an optimal control problem (cf. chapter 3) with a cost function depending on a given parameter $y \in \mathbb{R}^m$:

Definition 4.1.1. ***(LLP of Inverse Optimal Control)***
In accordance with the definition 3.0.1 of an optimal control problem let $\overline{x}$ be the state function and $\overline{u}$ the control function. Generalize the functions $\overline{\phi}$, ϕ_b and ϕ_I to depend additionally on the given parameter $y \in \mathbb{R}^m$ in a continuously differentiable manner. The ***lower level problem of inverse optimal control*** *is given by:*

$$\min_{\overline{x},\overline{u}} \overline{\phi}(\overline{x},\overline{u},y) = \phi_b(\overline{x}(0),\overline{x}(1),y) + \int_0^1 \phi_I(\overline{x}(t),\overline{u}(t),y)\ \mathrm{d}t$$

subject to $\overline{n}$ *ordinary differential equations*

$$\overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}(t)),$$

to $\overline{l}$ *inequality constraints*

$$g(\overline{x}(t), \overline{u}(t)) \leq 0$$

and to $\overline{c}$ *boundary conditions*

$$b(\overline{x}(0), \overline{x}(1)) = 0.$$

Remark 4.1.2. Note that the parameter y influences only the cost function $\overline{\phi}$ of the lower level problem, i.e., the parameter vector specifies a specific cost function within a given parameterized family of cost functions. A simple example of such a parameterized family is obtained if all convex combinations of some basic cost functions are considered; in this case the parameters correspond to the weighting factors of the convex combination (see for example section 6.4.4).

Remark 4.1.3. In general, the upper level variable y can also appear in the ordinary differential equation φ, the inequality conditions g and the boundary conditions b. Note that in this case only minor details in the problem structure change. However, the numerical examples of chapters 6 to 8 address the problems where the upper level variable influence only the objective on the lower level.

The goal of the upper level problem of inverse optimal control is to determine the optimal value of the parameter vector y, i.e., the upper level state, such that the corresponding optimal values $\overline{x}^*(y)$ and $\overline{u}^*(y)$ minimize the distance measure $\overline{\Phi}$ in the upper level problem:

Definition 4.1.4. ***(ULP of Inverse Optimal Control)***
Let $\Lambda^d \in \mathbb{R}^{\underline{m} \times \underline{\nu}}$ *be a matrix of given data values, i.e., each column* $\Lambda^d_{\cdot,i} \in \mathbb{R}^{\underline{m}}$ *corresponds to given measurements at specific time instances* $t_i^d \in [0,1]$ *for* $i = 1, \ldots, \underline{\nu}$*, and let* $\overline{\Phi}$ *be a continuously differentiable cost function. The* ***upper level problem of inverse optimal control*** *is given by*

$$\begin{aligned} \min_{y} \quad & \overline{\Phi}\left(\overline{x}^*, \overline{u}^*, \Lambda^d\right) \\ \text{subject to} \quad & 0 = H(y), \\ & 0 \geq G(y), \end{aligned}$$

where $\overline{x}^*$ *and* $\overline{u}^*$ *are the optimal values for the lower level problem corresponding to* y*.*

Remark 4.1.5. If $\overline{\Lambda}^c$ is a function using the inputs $\overline{x}^*$, $\overline{u}^*$ and a time instance $t \in [0,1]$ to compute the LLP-related values that have to be compared to the measurements, a simple realization of the cost function $\overline{\Phi}$ could be

$$\overline{\Phi}_{diff}(\overline{x}^*, \overline{u}^*, \Lambda^d) := \sum_{i=1}^{\underline{\nu}} \left(\overline{\Lambda}^c(\overline{x}^*, \overline{u}^*, t_i^d) - \Lambda^d_{\cdot,i}\right)^2,$$

computing the sum of squared differences at the data time instances t_i^d, $i = 1, \ldots, \underline{\nu}$. For a more detailed discussion of ULP distance measures see section 4.3.

Our solution strategy for such an inverse optimal control problem is to discretize the optimal control problem using the collocation technique presented in chapter 3. The bilevel problem is then reformulated as a one-level problem by replacing the lower level problem by its KKT-conditions (cf. appendix A); note that these conditions are in general only necessary but

not sufficient optimality conditions and consequently, the reformulated problem we solve is not equivalent to the original inverse optimal control problem. Since this reformulation yields usually a mathematical problem with complementarity constraints (cf. chapter 2), a relaxation approach is used to generate a sequence of standard nonlinear optimization problems which are solved by using an interior-point method (cf. section 5.1.1). This solution strategy is realized in the method `coreDBO`; for details see section 4.4 and section 5.2.

4.2 State of the Art

We will start the review of the state of the art regarding bilevel optimal control problems by stating related problems and then advancing to necessary optimality conditions for bilevel optimal control problems. This is followed by a more detailed discussion of recent works on bilevel and inverse optimal control in chronological order, see sections 4.2.1 to 4.2.7.

The first problems of bilevel optimal control resulted in the context of Stackelberg games. In [59] a *two-person dynamic game* is introduced which combines the problem structure of a *Stackelberg game* with ideas from optimal control. Both players are assumed to choose their individual control functions $\overline{u}^I : [0,1] \to \mathbb{R}^{\overline{m}_1}$ and $\overline{u}^{II} : [0,1] \to \mathbb{R}^{\overline{m}_2}$ in order to minimize their individual cost functions

$$\overline{\phi}^I(\overline{x}, \overline{u}^I, \overline{u}^{II}) := \int_0^1 \phi_I^I(\overline{x}(t), \overline{u}^I(t), \overline{u}^{II}(t), t)\ \mathrm{d}t$$

and

$$\overline{\phi}^{II}(\overline{x}, \overline{u}^I, \overline{u}^{II}) := \int_0^1 \phi_I^{II}(\overline{x}(t), \overline{u}^I(t), \overline{u}^{II}(t), t)\ \mathrm{d}t,$$

where $\overline{x}$ is the common state whose starting value $\overline{x}(0)$ is given and the dynamics are described by the ordinary differential equation

$$\overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}^I(t), \overline{u}^{II}, t).$$

Note that the two cost functions are not necessary opposing each other and consequently, variants like cooperative games are possible. In [59] neither control nor state constraints are considered and it is assumed that the necessary optimality conditions are sufficient for the lower level problem. In [356] a similar strategy is used while assuming that the optimal control problem has linear dynamics and a convex cost function.

Considering dynamic Stackelberg games with feedback, algorithms of dynamic programming are used to solve games with dependent or independent followers and various numbers of leaders [10, 227, 228]. A dynamic game modeling an economic control problem is discussed in [119] and the solution strategy is based on determining a relation between state and adjoint variables. Optimal management of fishery introduced by [62] and refined by [253] is another application of bilevel optimal control where an analytical solution can be derived.

Closely related to bilevel optimal controls problem is the field of dynamic games [22] where various numbers of players with continuous or discretized dynamics are considered. A special type of dynamic games are the *pursuit evasion games* where different players try to force or avoid collision. In [39, 40] this problem class is analyzed and necessary optimality conditions are derived which lead to multi-point boundary value problems. Furthermore, numerical methods based on the multiple-shooting algorithm are presented to solve the problem. A related approach for a space shuttle reentry can be found in [38]. In [87] the pursuit evasion

game is discretized yielding a min-max-problem and solved by a first-order method using sensitivity information of the lower level problem; in a more general setting subgradient methods might be necessary.

The problem of optimally controlling a hypersonic flight while guaranteeing full safety for mission aborts at all time instances is discussed in [56]. The safety constraint can be transformed into a series of further optimal control problems. The problem is solved by a multiple-shooting method. Trying to determine the global solution of a bilevel optimal control problem, the strategy presented in [216] simplifies the problem by replacing the lower level problem by a bound on the lower level cost.

In several publications of Ye, e.g., [351, 352], the derivation of necessary optimality conditions for bilevel optimal control problems is addressed. The presented technique, being similar to their approach for standard bilevel problems [354] (cf. section 2.3), is based on the optimal value function of the lower level problem. Using our notation, the bilevel problems considered in [352] have the following structure: Given the upper level control function $\overline{u}^I$, the lower level problem is a standard optimal control problem:

$$\min_{\overline{x},\overline{u}} \overline{\phi}^{II}(\overline{x},\overline{u}^I,\overline{u}^{II}) = \phi_b^{II}(\overline{x}(0),\overline{x}(1)) + \int_0^1 \phi_I^{II}(\overline{x}(t),\overline{u}^I(t),\overline{u}^{II}(t),t)\ \mathrm{d}t$$

subject to

$$\overline{x}'(t) = \varphi(\overline{x}(t),\overline{u}^I(t),\overline{u}^{II}(t))$$

and a control constraint $\overline{u}^{II}(t) \in \mathbb{U}^{II}(t)$ for a given set of feasible controls $\mathbb{U}^{II}(t)$ for $t \in [0,1]$ almost everywhere. Constraining the upper level control by $\overline{u}^I(t) \in \mathbb{U}^I(t)$, the upper level cost function has the structure:

$$\overline{\phi}^I(\overline{x},\overline{u}^I,\overline{u}^{II}) = \phi_b^I(\overline{x}(0),\overline{x}(1)) + \int_0^1 \phi_I^I(\overline{x}(t),\overline{u}^I(t),\overline{u}^{II}(t),t)\ \mathrm{d}t.$$

For details on the functions and sets in the problem definitions we refer to the original work [352]. This bilevel optimal control problem is reformulated to a one-level problem by using the optimal value function ϖ which is in general a non-smooth function. Consequently, the theory on subgradients of the Clark-type [61] is used to derive the necessary optimality conditions. A variant where in the upper level a finite number of parameters instead of a continuous function can be controlled is presented in [351].

4.2.1 Inverse Optimal Control of Human Car-Steering

The problem of inverse optimal control is introduced by Butz and his advisor von Stryk in [50] where a car-steering problem is modeled by an optimal control approach and the inverse problem of determining the underlying optimization principles of human-steered lane changes is addressed. The models of different complexities for the dynamic system of the car are discussed. For the numerical computations a simplified single-track model of the car is used which assumes, for example, a constant velocity during the maneuver and linearizes certain sine and cosine functions of the total nonlinear single-track model of the car (compare section 7). Several cost functions that can be useful to model the lane change maneuver are introduced; they range from state and control minimization to minimization of the deviation from the center of the lines. The optimal control problems of the lower level are solved by utilizing the direct optimization method `DIRCOL` [330] based on a collocation method (cf. section 3).

For solving the inverse problem a sensitivity problem is analyzed: How does the solution of the discretized optimal control problem depend on the parameters of the discretization? Using the accessory minimum problem known from optimal control theory for neighboring extremals, it is shown that a linear-quadratic optimal control problem has to be solved to determine the sensitivities. It is reported that the introduced approach to determine the sensitivities is more robust than the approach of [49] where the sensitivities of the discretized optimal control problem are analyzed. In consequence, the approach of [50] allows to maintain the bilevel structure of the inverse optimal control problem and to use only the sensitivities of the optimal control problem on the upper level to determine the next iterate of parameters for the lower level using a variant of the Levenberg-Marquardt algorithm.

The actual inverse optimal control problems solved in [50] combine three basic cost functions for the lower level (total path length, deviation from the center of the lines and lateral acceleration) with the simplified single-track model. The analyzed driving maneuver is a double lane change and numerical results of inverse optimization for both synthetic and real measurement data are discussed. Note that the measurement data does not maintain a constant velocity during the maneuver, thus problem characteristics have to be adapted at each discretization instance.

4.2.2 Inverse Optimal Control of a Neuro-Musculoskeletal System

Inverse optimal control problems in the context of the human neuro-musculoskeletal system are discussed by Bottasso, Prilutsky, Croce, Imberti and Sartirana [34]. While in the modeling part of the paper the lower level problem is considered to be an optimal control problem, the presented examples are only static problems. The solution strategy is based on a discretization of the optimal control problem and the usage of the corresponding KKT-conditions to obtain a standard nonlinear optimization problem. Note that neither details on discretizing the optimal control problem nor on solving the resulting nonlinear problem are given; especially, the MPEC structure of the reformulated problem is not discussed. In consequence, this work introduces a general methodology similar to the one presented here, but numerical results on inverse optimal control problems are not given.

For the case of the lower level problem being a static optimization problem two numerical examples are presented: A two-dimensional leg pushing experiment and a planar arm stiffness experiment. The goal is to determine the load of individual muscles given force or stiffness measurements at the foot or the hand, respectively. Note that in this static setup no activation levels have to be modeled to capture the muscle behavior, but only forces have to be determined. In both cases the family of cost functions contains objectives minimizing the states of the model to various powers. Numerical results are reported to be similar to the recorded EMG-data.

The authors state that their tests suggest a potential applicability to more complex problems and that further work especially on dynamic motor problems is needed in order to assess the real usefulness of the proposed methodology.

4.2.3 Bilevel Optimal Control of a Rack Feeder

The works of Knauer and Büskens [177, 178] do not address inverse optimal control but a related bilevel optimal control problem. However, the presented solution approach could be applied to inverse optimal control problems, as well. The basic task is to optimally control a

ceiling-attached rack feeder in a high rack; such a rack feeder is a combination of a traveling trolley and a load handling device. The traveling trolley is fixed to the ceiling by rails along a lane of the high rack and the load handling device is connected to this trolley by four cables, lengths of which can be controlled. Consequently, this system of two bodies is modeled as a mathematical pendulum and the resulting ordinary differential equation is linear in most components. A central aspect in the control of such a rack feeder is to reduce the oscillations of the load handling device resulting from motions of the traveling trolley.

The optimal control problems discussed in [177] include physically-motivated box-constraints for the states and the controls and combinations of boundary conditions characteristic for the different tasks. Several cost functions are introduced for these tasks; in case of a free final time, one of these cost functions is the minimization of the free final time. Another criterion is motivated by maximal controllability, i.e., the sum of the squared controls is minimized, which corresponds to minimization of the jerk of both the position of the traveling trolley and the lengths of the cables. Other cost functions result if oscillations of the load handling device are to be minimized; the criteria used in the presented optimal control problems are linear combinations of these cost functions.

The considered class of bilevel optimal control problems is the combination of the following two problems; the focus of this presentation lies on the general structure. Consequently, details of the domain and image of the individual functions are omitted, but can be found in the original work [177]. The upper level program is given by

$$\begin{aligned} \min_{\overline{w}} \quad & \int_0^{\pi_f} \Phi_I(\overline{y}(\pi), \overline{x}, \overline{w}(\pi), \overline{u}) \, \mathrm{d}\pi \\ \text{s.t.} \quad & \overline{y}'(\pi) = \underline{\varphi}(\overline{y}(\pi), \overline{w}(\pi)), \quad \pi \in [0, \pi_f], \\ & 0 = B(\overline{y}(0), \overline{y}(\pi_f), y^s, y^e), \\ & G(\overline{y}(\pi), \overline{x}, \overline{w}(\pi), \overline{u}) \leq 0, \quad \pi \in [0, \pi_f], \\ & (\overline{x}, \overline{u}) \in \mathbb{L}(\overline{y}, \overline{w}), \end{aligned}$$

where $\overline{y}$ and $\overline{w}$ are the state and the control of the upper level program, respectively; the start and end value for the state is prescribed by the function B using the given values y^s and y^e. The minimized ULP cost function is Φ_I and the right-hand side of the ordinary differential equation of the ULP state is given by $\underline{\varphi}$. The lower level state and control are denoted by $\overline{x}$ and $\overline{u}$ and the solution set of the lower level program is $\mathbb{L}$. Furthermore, constraints linking LLP and ULP quantities are denoted by G.

The corresponding lower level program has the following form:

$$\begin{aligned} \min_{\overline{u}} \quad & \int_{t_0}^{t_f} \phi(\overline{y}, \overline{x}(t), \overline{w}, \overline{u}(t)) \, \mathrm{d}t \\ \text{s.t.} \quad & \overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}(t)), \quad t \in [t_0, t_f], \\ & 0 = b(\overline{x}(t_0), \overline{x}(t_f), x^s(\overline{y}, \overline{w}), x^e(\overline{y}, \overline{w})), \end{aligned}$$

where ϕ is the minimized cost function and φ the right-hand side of the ordinary differential equation for the LLP state. Note that the time interval considered in the lower level program might differ from the one of the upper level and that the start and end values for the LLP state x^s and x^e, which are used by the function b to describe the boundary conditions, might depend on the ULP state and control.

Three scenarios of such bilevel optimal control problems are discussed in [177] for the rack feeder. The first problem is a parametric bilevel problem where on the lower level the squared

control input is minimized depending on a parameter stating the start velocity of the traveling trolley. In the upper level program the value for the parameter minimizing the oscillations of the load is to be chosen. In consequence, no inequality constraints or ordinary differential equations have to be considered in the upper level.

The second problem considers collision avoidance of two rack feeders whose tracks are fixed at different heights, allowing one rack feeder to pass below the other if the corresponding load is high enough. The control problem of the upper rack feeder is the upper level of the bilevel program, because its actions limit the lower rack feeder more than the other way round.

The third problem, which is the initial problem of the work of [177], is the controlled stopping of a rack feeder. To guarantee that at every moment the rack feeder can be stopped within a given time period is essential from the engineering perspective, because safety regulations have to be fulfilled by the system to be commercially usable. Given a current state of the rack feeder, optimally controlling it to a state with zero velocities and accelerations within a given time period is captured by a lower level program. The original optimal control task of moving the rack feeder from one position to another is the upper level program. Note that ideally one would solve this bilevel program with infinitely many lower level programs, i.e., one for each time instance, but to allow for a numerical solution only a limited number of such lower level programs can be considered. Consequently, the bilevel problem solved in [177] considers a few of these problems at time instances equally distributed over the total time interval of the upper level motion. Approximations for intermediate values can be obtained by sensitivity analysis of the optimal control problems on the lower level.

The solution strategy used to solve these bilevel optimal control problems is a hybrid method combining the indirect approach for optimal control problems with the direct one. The lower level problems are replaced by the necessary conditions of optimal control theory, i.e., a multi-point boundary value problem is considered. Thereby the bilevel optimal control problem is transformed into a standard optimal control problem which is then solved by routine `NUDOCCCS` [49] of Büskens.

Note that theoretically this hybrid approach can also be applied to inverse optimal control problems. However, the disadvantages of indirect optimal control concerning the starting values for the numerical solver would hold true for the transformed problem. Due to the more complex dynamical systems considered in the inverse optimal control problems compared to the mathematical pendulum of the rack feeder, these disadvantages might be significant.

4.2.4 Inverse Optimal Control of Human Navigation

The problem of inverse optimal control for human motions is introduced by Mombaur, Truong and Laumond in [217]. Their goal is to find the cost function underlying the human locomotion and use it to control a humanoid robot accordingly. The locomotion tasks are to walk from a start position to a designated end position with given orientation at a comfortable speed. The perspective on the human walking problem is macroscopic which means that one is interested in the trajectories of position and orientation, but individual steps or rigid body and muscle dynamics are not of interest. Consequently, a point model without mass or inertia is used that can be controlled by acceleration in forward and sideward direction and rotational acceleration (cf. chapter 8). The resulting optimal control problem in the lower level is solved by the multiple shooting method `MUSCOD` [79]. This direct method solves the obtained nonlinear problem with an SQP method.

The upper level cost function is the sum of squared norms of differences between computed states and controls of the lower level and the recorded ones at given time instances. The bilevel problem is solved by using separate solvers for both levels, but without using any derivative or sensitivity information of the lower level solution in the upper level. Consequently, the derivative-free method `BOBYQA` [251] is used, which realizes an interpolation-based trust region technique. The considered family of cost functions for the lower level is generated by linear combinations of five basic cost functions: Minimization of the three individual controls, minimization of total motion time and minimization of deviation between current orientation and orientation to the goal position (cf. chapter 8). The numerical results for several locomotion scenarios are discussed and it is reported that a combination of the five cost functions exists reproducing the observed main characteristics of the human locomotion.

4.2.5 Inverse Optimal Control of Human Arm Motions

The basic techniques of the approach of [217] are used by Berret, Chiovetto, Nori and Pozzo [26] to analyze planar human arm motions. A standard dynamical model consisting of two rigid bodies without muscles is used to model the dynamics of the human arm. Several basic cost functions are considered for the generation of the linear family of possible lower level cost functions: On the one hand, integrals over the squared values of hand jerk, angle jerk, angle acceleration, torque change and torque. On the other hand minimization of geodesic length, energy and effort. The resulting optimal control problem is transformed into a standard nonlinear optimization problem by using the collocation method `GPOPS` [255], which is a Gauss pseudospectral method, and the nonlinear problem is then solved with the optimization method `SNOPT` [125].

Two metrics for the upper level problem are discussed: First, comparison of Cartesian and curvature differences between the solution of the lower level and the recorded data. Second, likelihood values using a Gaussian Mixture Model resulting from the recorded data. The upper level problem is then solved by the derivative-free trust-region method `CONDOR` [323]. Numerical results presented suggest that humans might use a composite cost function in the considered planar arm motions.

4.2.6 Bilevel Optimal Control of Flight Trajectory Optimization

The bilevel problem of optimizing the track for an air race in order to minimize safety- and fairness-related cost functions while assuming that the planes are controlled time-optimally is addressed by Fisch [95]. A detailed dynamical model of an air race plane suitable for optimal control is introduced using rigid body dynamics and several simplified versions are deduced. The optimal control problems of minimizing the flight time for a given race track are solved by using a multiple-shooting method where the entire race trajectory is divided into subproblems with continuity conditions at the start and end of each segment. A process of alternating simulation and optimization using plane models of increasing complexity is presented in order to get good starting values for the optimization of the full optimal control problem. To avoid control oscillations and to assure that this alternating process works, a penalty term of squared control derivatives is added to the cost function.

The task of the upper level problem is to optimize the positions of the gates that have to be passed by the planes to maximize audience safety. For each gate a specific flight position is required and the gate position can be varied within given box constraints. Discussed

upper level cost functions are, for example, the minimization of distance or flight time of the plane to the crowd or minimization of flight time differences between different types of planes. These bilevel problems are solved using a simplified model of the plane dynamics. The sensitivity analysis for optimal control problems of [49] is utilized within a gradient-based descent method to solve the upper level problem.

4.2.7 Inverse Optimal Control of Human Leg Motions

In their current research Hatz, Schlöder and Bock address problems of inverse optimal control. Their specific interest lies on identifying cost functions and parameters in human gaits in order to provide tools of optimal control to the field of orthopedics. For example, such an optimal control model could be used to simulate the gait and analyze the interaction of the individual body segments or to predict the outcome of certain surgeries.

In [147] a benchmark problem of a rocket car with friction is used to analyze differences in replacing the lower level problem by its first-order necessary conditions obtained by the direct or the indirect approach. Using the indirect approach, a multi-point boundary value problem is obtained and solved by the optimization method `MUSCOD` [79]. The direct approach is realized by using the multiple-shooting method and replacing the resulting nonlinear lower level problem by its KKT-conditions. The necessary derivative information is obtained by internal numerical differentiation [33] in the course of solving the ordinary differential equation on the sub-intervals by utilizing the multiple-shooting method. Note that the structure of the resulting MPEC is considered by following the line of [109, 111] for SQP-methods (see chapter 2). The optimizaton method `MUSCOD` is used to solve the resulting nonlinear optimization problem. The numerical results for the benchmark problem presented in [147] are reported to be reasonable for both approaches.

4.3 ULP Distance Measures

The goal of the upper level cost function $\overline{\Phi}$ in the inverse optimal control problem is to measure the distance between given data Λ^d and the corresponding quantities resulting from the LLP solution. In some cases it might be necessary to compare several quantities with different physical meanings, e.g., positional information and forces, which results in the problem of determining a suitable (relative) scaling of the quantities. If linear combinations of distance measures for single quantities only are considered, suitable weighting factors have to be chosen.

Since a direct approach is used to solve the optimal control problem, we address here the problem of defining a distance measure Φ. It is assumed that $\underline{\nu}$ measurements at time instances t_i^d, $i = 1, \ldots, \underline{\nu}$, are given in form of the matrix Λ^d. This means that the column $\Lambda^d_{\cdot,i} \in \mathbb{R}^{\underline{m}}$ represents the given data for t_i^d. Furthermore, the state x of the optimal control problem in the lower level yields the state information $x_{(i)}$ for time instances t_i, $i = 1, \ldots, \nu$. We assume that a function Λ^c exists that maps a state of the dynamics model to the quantity measured in the data. In consequence, the distance of the set

$$\left\{ \left(t_i, \left(\Lambda^c \left(x_{(i)} \right) \right)^T \right) \mid i = 1, \ldots, \nu \right\}$$

from the data set

$$\left\{ \left(t_i^d, \left(\Lambda^d_{\cdot,i} \right)^T \right) \mid i = 1, \ldots, \underline{\nu} \right\}$$

is sought. Two distance measures, one neglecting the temporal dimension of these set and one explicitly using this information, are introduced in the following.

If the temporal information in the data and the LLP state is to be used, only time instances in the interval $[\widetilde{t}_{max}, \widetilde{t}_{min}]$ defined by

$$\widetilde{t}_{max} := \max\left\{t_1, t_1^d\right\} \quad \text{and} \quad \widetilde{t}_{min} := \min\left\{t_\nu, t_{\underline{\nu}}^d\right\}$$

are suitable. Consequently, assume that $\widetilde{\nu}$ time instances $\widetilde{t}_j \in \left[\widetilde{t}_{max}, \widetilde{t}_{min}\right]$, $j = 1, \ldots, \widetilde{\nu}$, with $\widetilde{t}_j < \widetilde{t}_{j+1}$, $j = 1, \ldots, \widetilde{\nu}-1$, are given. To compare the two sets, a piecewise linear interpolation is used, because the sets $\{t_i \mid i = 1, \ldots, \nu\}$, $\left\{t_i^d \mid i = 1, \ldots, \underline{\nu}\right\}$ and $\left\{\widetilde{t}_i \mid i = 1, \ldots, \widetilde{\nu}\right\}$ might be disjoint. Given one of the time instances $\widetilde{t}_j$, the following value results for the lower level state:

$$\widehat{\chi}^{(j)} := \Lambda^c\left(x_{(i)}\right) + \frac{\widetilde{t}_j - t_i}{t_{i+1} - t_i}\left(\Lambda^c\left(x_{(i+1)}\right) - \Lambda^c\left(x_{(i)}\right)\right) \quad \text{for } \widetilde{t}_j \in [t_i, t_{i+1}].$$

The term for the corresponding data value has the same structure:

$$\underline{\chi}^{(j)} := \Lambda^d_{\cdot,k} + \frac{\widetilde{t}_j - t_k^d}{t_{k+1}^d - t_k^d}\left(\Lambda^d_{\cdot,k+1} - \Lambda^d_{\cdot,k}\right) \quad \text{for } \widetilde{t}_j \in [t_k^d, t_{k+1}^d].$$

Summing up, the following distance measure compares values at identical time instances:

$$\Phi_{time}(x, \Lambda^d) := \sum_{j=2}^{\widetilde{\nu}} \frac{\widetilde{t}_j - \widetilde{t}_{j-1}}{2}\left(\widehat{\chi}^{(j)} - \underline{\chi}^{(j)}\right)^2 + \sum_{j=1}^{\widetilde{\nu}-1} \frac{\widetilde{t}_{j+1} - \widetilde{t}_j}{2}\left(\widehat{\chi}^{(j)} - \underline{\chi}^{(j)}\right)^2.$$

The second distance measure useful in inverse optimal control is based on the path lengths (of the linear interpolations) of the LLP state $\widehat{l}$ and data $\underline{l}$:

$$\begin{aligned} \widehat{l} &:= \sum_{i=1}^{\nu-1} \left|\left|\Lambda^c\left(x_{(i+1)}\right) - \Lambda^c\left(x_{(i)}\right)\right|\right|, \\ \underline{l} &:= \sum_{k=1}^{\underline{\nu}-1} \left|\left|\Lambda^d_{\cdot,k+1} - \Lambda^d_{\cdot,k}\right|\right|. \end{aligned}$$

Comparing points of equal relative path length, the distance measure uses only geometrical properties of the recorded trajectories, thus it would identify a time-delayed trajectory with the original one, which might be important if the start and the end phase of a motion are less relevant than the middle part. Therefore, we assume that a vector $\widetilde{\sigma} \in \mathbb{R}^{\widetilde{\nu}}$ specifies the relative path lengths where the distance between the data and the LLP state have to be computed. To simplify notation, the relative path lengths $\widehat{\sigma} \in \mathbb{R}^{\nu}$ and $\underline{\sigma} \in \mathbb{R}^{\underline{\nu}}$ corresponding to the LLP states $x_{(i)}$ and the data instances $\Lambda^d_{\cdot,k}$, accordingly, are defined by:

$$\begin{aligned} \widehat{\sigma}_j &:= \frac{1}{\widehat{l}}\left(\sum_{i=1}^{j-1} \left|\left|\Lambda^c\left(x_{(i+1)}\right) - \Lambda^c\left(x_{(i)}\right)\right|\right|\right), \quad j = 1, \ldots, \nu, \\ \underline{\sigma}_j &:= \frac{1}{\underline{l}}\left(\sum_{k=1}^{j-1} \left|\left|\Lambda^d_{\cdot,k+1} - \Lambda^d_{\cdot,k}\right|\right|\right), \quad j = 1, \ldots, \underline{\nu}. \end{aligned}$$

Consequently, the comparison points corresponding to the LLP state

$$\widehat{\psi}^{(j)} := \Lambda^c\left(x_{(i)}\right) + \frac{\widetilde{\sigma}_j - \widehat{\sigma}_i}{\widehat{\sigma}_{i+1} - \widehat{\sigma}_i}\left(\Lambda^c\left(x_{(i+1)}\right) - \Lambda^c\left(x_{(i)}\right)\right) \quad \text{for} \quad \widetilde{\sigma}_j \in [\widehat{\sigma}_i, \widehat{\sigma}_{i+1}],$$

and the data

$$\underline{\psi}^{(j)} := \Lambda^d_{\cdot,k} + \frac{\widetilde{\sigma}_j - \underline{\sigma}_k}{\underline{\sigma}_{k+1} - \underline{\sigma}_k}\left(\Lambda^d_{\cdot,k+1} - \Lambda^d_{\cdot,k}\right) \quad \text{for} \quad \widetilde{\sigma}_j \in [\underline{\sigma}_k, \underline{\sigma}_{k+1}],$$

are introduced for $j = 1, \ldots, \underline{\nu}$. Finally, this yields the distance measure Φ_{path}:

$$\Phi_{path}(x, \Lambda^d) \;:=\; \sum_{j=2}^{\widetilde{\nu}} \frac{\widetilde{\sigma}_j - \widetilde{\sigma}_{j-1}}{2}\left(\widehat{\psi}^{(j)} - \underline{\psi}^{(j)}\right)^2 + \sum_{j=1}^{\widetilde{\nu}-1} \frac{\widetilde{\sigma}_{j+1} - \widetilde{\sigma}_j}{2}\left(\widehat{\psi}^{(j)} - \underline{\psi}^{(j)}\right)^2.$$

4.4 Structure of Discretized Inverse Optimal Control Problem

In this section both the structure of the discretized inverse optimal control problem and the structure of the reformulated one-level problem are discussed to address the existence of a global optimistic solution and the applicability of the interior-point optimization method `IPOPT` to solve the final nonlinear optimization problem.

The collocation approach of chapter 3 for discretizing an optimal control problem yields a nonlinear optimization problem of the following structure if it is applied to the lower level problem of inverse optimal control:

$$\begin{aligned}
&\min \phi(x, u, y) \\
\text{subject to}\quad & \\
&0 = -x_{(i+1)} + x_{(i)} + \Psi\left(x_{(i)}, u_{(i)}, x_{(i+1)}, u_{(i+1)}, \delta_i(\Delta)\right), \quad i = 1, \ldots, \nu - 1, \\
&0 = b\left(x_{(1)}, x_{(\nu)}\right), \\
&0 \geq g\left(x_{(i)}, u_{(i)}\right), \quad i = 2, \ldots, \nu - 1.
\end{aligned}$$

Note that this problem form subsumes both discretization strategies discussed in detail in section 3.2.1; the function Ψ consequently represents the weighted sum of evaluations of the right-hand side φ of the ordinary differential equation for the state.

To write this problem in the form of a standard lower level problem (cf. section 2.1), the following definitions are made:

The lower level state $\underline{x}$ is given by the concatenation of the states $x_{(i)}$ and the controls $u_{(i)}$:

$$\underline{x} := \left(x_{(1)}^T, u_{(1)}^T, \ldots, x_{(\nu)}^T, u_{(\nu)}^T\right)^T,$$

where $n := \nu\left(\overline{n} + \overline{m}\right)$.

Consequently, the inequality constraints g can be defined in the form

$$g(\underline{x}) := \begin{pmatrix} g\left(x_{(2)}, u_{(2)}\right) \\ \vdots \\ g\left(x_{(\nu-1)}, u_{(\nu-1)}\right) \end{pmatrix}$$

and thus $p := \bar{l}\,(\nu - 2)$. Finally, the equality constraints h are considered to be given by

$$h(\underline{x}, y) := \begin{pmatrix} -x_{(2)} + x_{(1)} + \Psi\left(x_{(1)}, u_{(1)}, x_{(2)}, u_{(2)}, \delta_1(\Delta)\right) \\ \vdots \\ -x_{(\nu)} + x_{(\nu-1)} + \Psi\left(x_{(\nu-1)}, u_{(\nu-1)}, x_{(\nu)}, u_{(\nu)}, \delta_{\nu-1}(\Delta)\right) \\ b\left(x_{(1)}, x_{(\nu)}\right) \end{pmatrix}$$

with $q := (\nu - 1)\,\overline{n} + \overline{c}$ and a given partition Δ. Using these notations together with the cost function

$$\phi(\underline{x}, y) := \phi(x, u, y),$$

the discretized optimal control problem has the standard form of a lower level problem (see definition 2.1.1):

Definition 4.4.1. ***(Discretized LLP of Inverse Optimal Control)***

$$\min_{\underline{x}} \phi(\underline{x}, y) \quad \textit{subject to} \quad h(\underline{x}) = 0, \quad g(\underline{x}) \leq 0.$$

4.4.1 Global Optimistic Solution

Since the BL-MFCQ is a requirement for theorem 2.2.10 stating sufficient conditions for the existence of a global optimistic solution for a standard bilevel program, the block structure of the Jacobians for both the equality constraints and the inequality constraints are discussed in the following.

$$\nabla g(\underline{x}) = \left(\begin{array}{c|ccc|c} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \hline \nabla_x g\left(x_{(2)}, u_{(2)}\right) & 0 & \cdots & 0 & 0 \\ \nabla_u g\left(x_{(2)}, u_{(2)}\right) & 0 & \cdots & 0 & 0 \\ \hline 0 & & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & & 0 \\ \hline 0 & 0 & \cdots & 0 & \nabla_x g\left(x_{(\nu-1)}, u_{(\nu-1)}\right) \\ 0 & 0 & \cdots & 0 & \nabla_u g\left(x_{(\nu-1)}, u_{(\nu-1)}\right) \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{array}\right)$$

In order to get a more compact notation, we drop the arguments of Ψ and denote the gradient with respect to input value i by ∇_i, e.g., the derivative $\nabla_{12}\Psi$ in the i-th block row is then the short form for

$$\nabla_{12}\Psi = \begin{pmatrix} \nabla_{x_{(i)}}\Psi\left(x_{(i)}, u_{(i)}, x_{(i+1)}, u_{(i+1)}, \delta_i(\Delta)\right) \\ \nabla_{u_{(i)}}\Psi\left(x_{(i)}, u_{(i)}, x_{(i+1)}, u_{(i+1)}, \delta_i(\Delta)\right) \end{pmatrix}$$

and abusing the notation $\nabla_1 b$ denotes

$$\nabla_1 b = \begin{pmatrix} \nabla_{x_{(1)}} b\left(x_{(1)}, x_{(\nu)}\right) \\ 0 \end{pmatrix} \in \mathbb{R}^{(\overline{n}+\overline{m})\times\overline{c}}.$$

As a result the derivative of the equality constraint can be written as

$$\nabla h(\underline{x}) = \left(\begin{array}{c|c|c|c|c|c} \mathcal{U}+\nabla_{12}\Psi & 0 & \cdots & 0 & 0 & \nabla_1 b \\ \hline -\mathcal{U}+\nabla_{34}\Psi & \mathcal{U}+\nabla_{12}\Psi & \ddots & \vdots & \vdots & 0 \\ \hline 0 & -\mathcal{U}+\nabla_{34}\Psi & \ddots & 0 & 0 & \vdots \\ \hline 0 & 0 & \ddots & \mathcal{U}+\nabla_{12}\Psi & 0 & \vdots \\ \hline \vdots & \vdots & \ddots & -\mathcal{U}+\nabla_{34}\Psi & \mathcal{U}+\nabla_{12}\Psi & 0 \\ \hline 0 & 0 & \ldots & 0 & -\mathcal{U}+\nabla_{34}\Psi & \nabla_2 b \end{array}\right),$$

where $\mathcal{U} \in \mathbb{R}^{(\overline{n}+\overline{m})\times\overline{n}}$ is the identity matrix defined by

$$\mathcal{U}_{i,j} = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases}.$$

Note that for the assumptions of theorem 2.2.10 the BL-MFCQ has to be fulfilled for all feasible points. Given a specific discretization strategy and a particular ordinary differential equation φ, the structure of the above derivatives can be analyzed and the LICQ is fulfilled in certain settings (compare the next example).

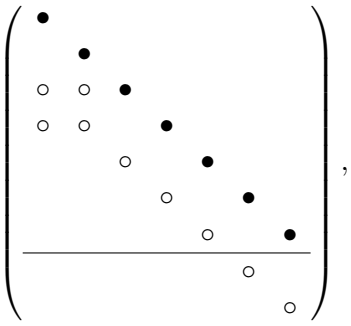
All example problems discussed in this work are based on models of dynamic systems where the input is human-generated and at least theoretic control bounds exist. In consequence, the set of possible states is also bounded if the ordinary differential equation relating the control input to the change of the state satisfies a Lipschitz-condition, i.e., no blow-up is observed [15, 78]; for the discussed examples this assumption does hold. Additionally, all constraints of the discretized lower level problem are at least continuous. Consequently, the feasible set of the lower level problem can be assumed to be compact and non-empty for a well-posed problem, i.e., condition BL-C 2.2.5 is fulfilled. Compare the section 3.4 on the existence of a solution of the optimal control problem.

Summing up, if the problem and the chosen discretization strategy guarantee that both conditions of theorem 2.2.10 are fulfilled, the existence of a global optimistic solution for the discretized inverse optimal control problem results.

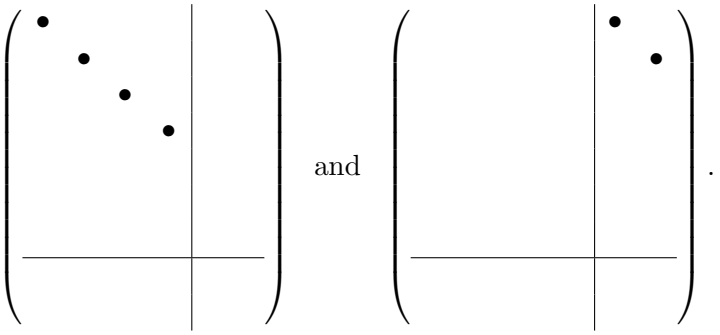
Example 4.4.2.
If the unicycle model used in the context of human navigation (cf. chapter 8) is combined with the discretization strategy (type I) of chapter 3, a simple example results where the

LICQ can be shown. The considered locomotion task is to move from a start position with specified orientation and velocity to a given end position. In addition to the dynamics and the boundary condition, upper and lower bounds on the controls are considered.

First, we state the structures of the blocks in $\nabla h(\underline{x}, y)$ by marking values with an absolute value of one by $\bullet$ and other non-zero values by $\circ$. The block given by $\mathcal{U}+\nabla_{12}\Psi$ and $-\mathcal{U}+\nabla_{34}\Psi$ has the structure



where the derivatives with respect to the two controls are non-zero in all cases. It can be seen that all these blocks have full column-rank. The blocks corresponding to the boundary conditions are structured in the following way:



As a consequence of its structure the gradient of the equality constraints has full column-rank. Additionally, the active inequality constraints have to be considered to show that the LICQ is fulfilled. Note that the upper and lower bound on the controls do not violate the LICQ if they become active.

Summing up, for this simple example the existence of a global optimistic solution results. In case of additional inequality constraints, different boundary conditions, other dynamics or discretization approaches such a structure analysis might not be as simple as in this example and further assumptions might be necessary to assure the linear independence constraint qualification.

4.4.2 CQ for the Transformed (One-Level) Problem

In addition to the question whether the discretized bilevel problem has a global optimistic solution, we now want to address the question whether the MPEC-LICQ holds if the lower level problem is replaced by its first-order necessary conditions. Since an interior-point algorithm is used in this work to solve the nonlinear optimization problems generated by the two-sided relaxation scheme 2.5.4, it has to be analyzed under which assumptions the KKT-conditions are necessary for the relaxed one-level problem.

The following definition states the structure of the transformed one-level problem:

Definition 4.4.3. ***(Transformed Problem of Inverse Optimal Control)*** *The* ***transformed (one-level) problem of inverse optimal control*** *is given by*

$$\begin{aligned} \min \quad & \Phi(\underline{x}, y, \Lambda^d) \\ \textit{subject to} \quad & 0 = \nabla_x \phi(\underline{x}, y) + \nabla g(\underline{x})\lambda + \nabla h(\underline{x})\mu, \\ & 0 = h(\underline{x}) \\ & 0 = H(y) \\ & 0 \leq -g(\underline{x}) \ \perp \ \lambda \geq 0 \\ & 0 \geq G(y), \end{aligned}$$

where Φ is a suitable discrete version of the cost function $\overline{\Phi}$ and the functions H and G define the feasible set of ULP states. Note that the Lagrange multipliers for the lower level constraints λ and μ are optimization variables in addition to the LLP state $\underline{x}$ and the ULP state y.

The following theorem relates assumptions on the structure of the lower and upper level programs to the structure of the MPEC 4.4.3.

Theorem 4.4.4.
Let $\underline{x}^*$ be optimal for the lower level problem of the inverse optimal control problem for a given upper level state y and let λ^* and μ^* be the corresponding Lagrange multipliers.
Assume that the lower level solution fulfills strict complementarity and both the LICQ and the SOSC hold true. If, additionally, the LICQ is fulfilled at the upper level, then the MPEC-LICQ is satisfied for the reformulated problem 4.4.3.

Proof. The following two functions combine the equality and inequality constraints relevant for the MPEC-LICQ:

$$\widetilde{h}(\underline{x}, y, \lambda, \mu) := \begin{pmatrix} \nabla_x \phi(\underline{x}, y) + \nabla g(\underline{x})\lambda + \nabla h(\underline{x})\mu \\ h(\underline{x}) \\ H(y) \end{pmatrix}$$

and

$$\widetilde{g}(\underline{x}, y, \lambda, \mu) := \begin{pmatrix} g(\underline{x}) \\ -\lambda \\ G(y) \end{pmatrix}.$$

The Jacobians of the equality constraints and all active inequality constraints yield the following derivative matrix where the arguments of the functions are dropped for presentation reasons:

$$\left(\nabla \widetilde{h}, \nabla \widetilde{g}_{\mathbb{A}}\right)^T = \begin{pmatrix} D_{xx}\left(\phi + g\lambda + h\mu\right) & D_{xy}\phi & \nabla g & \nabla h \\ Dh & 0 & 0 & 0 \\ 0 & DH & 0 & 0 \\ Dg_{\mathbb{A}} & 0 & 0 & 0 \\ 0 & 0 & -\mathcal{U}_{\mathbb{A}} & 0 \\ 0 & DG_{\mathbb{A}} & 0 & 0 \end{pmatrix}. \tag{4.1}$$

If one assumes that the LICQ holds true for both the lower and the upper level program, the following two matrices have full row-rank:

$$\begin{pmatrix} Dh \\ Dg_{\mathbb{A}} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} DH \\ DG_{\mathbb{A}} \end{pmatrix}.$$

The linear independence of the rows of matrix (4.1) is proven here by contraposition: Assume a vector

$$d := \left(d_x^T, d_h^T, d_H^T, d_g^T, d_G^T\right)^T \in \mathbb{R}^{n+q+\underline{q}+p+\underline{p}} \backslash \{0\}$$

exists such that $d^T \left(\nabla \widetilde{h}, \nabla \widetilde{g}_{\mathbb{A}}, 0\right)^T = 0$, then the following equations result:

$$\begin{aligned}
d_x^T D_{xx} \left(\phi + g\lambda^* + h\mu^*\right) + d_h^T Dh + d_g^T \begin{pmatrix} Dg_{\mathbb{A}} \\ 0 \end{pmatrix} &= 0, && (4.2)\\
d_x^T D_{xy}\phi + d_H^T DH + d_G^T \begin{pmatrix} DG_{\mathbb{A}} \\ 0 \end{pmatrix} &= 0, && (4.3)\\
d_x^T \begin{pmatrix} \nabla g_{\mathbb{A}}, & \nabla g_{\mathbb{I}} \end{pmatrix} + d_g^T \begin{pmatrix} 0, & -\mathcal{U}_{\mathbb{A}} \end{pmatrix} &= 0, && (4.4)\\
d_x^T \nabla h &= 0, && (4.5)
\end{aligned}$$

where for notation reasons it is assumed that the indices of the LLP inequality constraints are reordered such that first all active constraints and then all inactive constraints are stated. Note that for a vector with $d_x \neq 0$ the following two inequalities have to hold:

$$\begin{aligned}
d_x^T \nabla g_{\mathbb{A}} &= 0,\\
d_x^T \nabla h &= 0.
\end{aligned}$$

Consequently, the vector d_x is an element of the cone $\mathbb{T}^+(g, h, \underline{x}^*, \mu^*)$ (cf. section A.2) and thus

$$d_x^T D_{xx} \left(\phi + g\lambda^* + h\mu^*\right) d_x > 0$$

as a consequence of the second-order sufficient condition for the lower level solution.
However, this yields a contradiction to equation (4.2), because

$$\begin{aligned}
& \left(d_x^T D_{xx} \left(\phi + g\lambda^* + h\mu^*\right) + d_h^T Dh + d_g^T \begin{pmatrix} Dg_{\mathbb{A}} \\ 0 \end{pmatrix}\right) d_x\\
= \; & \underbrace{d_x^T D_{xx} \left(\phi + g\lambda^* + h\mu^*\right) d_x}_{>0} + d_h^T \underbrace{Dh d_x}_{=0} + d_g^T \underbrace{\begin{pmatrix} Dg_{\mathbb{A}} d_x \\ 0 \end{pmatrix}}_{=0}\\
> \; & 0.
\end{aligned}$$

Therefore, the assumption $d_x \neq 0$ does not hold and the equation system (4.2) - (4.5) simplifies to

$$\begin{aligned}
d_h^T Dh + d_g^T \begin{pmatrix} Dg_{\mathbb{A}} \\ 0 \end{pmatrix} &= 0,\\
d_H^T DH + d_G^T \begin{pmatrix} DG_{\mathbb{A}} \\ 0 \end{pmatrix} &= 0,\\
d_g^T \begin{pmatrix} 0, & -\mathcal{U}_{\mathbb{A}} \end{pmatrix} &= 0.
\end{aligned}$$

Note that this equation system can only be fulfilled by $d = 0$, since the LICQ is assumed to be fulfilled for both the lower and the upper level program. This contradicts the initial assumption and consequently, the MPEC-LICQ is fulfilled for problem 4.4.3. □

Remark 4.4.5. If all upper level states are weighting factors in a convex combination of basic lower level cost functions, the following constraints have to be fulfilled:

$$\mathbb{1}^T y = 1 \quad \text{and} \quad y \geq 0.$$

This yields the derivative matrix $(\mathbb{1}, \mathcal{U})$ and the LICQ is consequently fulfilled, because at least one value $y_i > 0$.

Numerical Methods

Chapter 5

In the previous chapters discretization and reformulation techniques have been discussed which allow to solve optimal control problems and inverse optimal control problems with numerical solvers for general nonlinear optimization problems. Therefore, such solvers are addressed in the first section of the following chapter with the focus on interior point methods (cf. section 5.1.1).

Since the implementation `IPOPT` [333] of an interior point method is used as the basic solver for the inverse optimal control code `coreIOC`, some details relevant for the performance analysis our approach are introduced in section 5.1.3. For the numerical realization of the solution strategy for inverse optimal control problems discussed in chapters 2 to 4 some details on time discretization, scaling and goal attainment are discussed in section 5.2.

5.1 Numerical Strategies for Nonlinear Optimization

For nonlinear optimization problems an evolved optimality theory exists and several theorems characterizing local solutions are known (see appendix A). However, analytically solving a nonlinear problem is in general not possible. Consequently, various numerical approaches have been proposed to solve such problems; some of these approaches are specially tailored for a rather specific subclass of problems whereas others consider the general problem. The general idea of most methods is to iteratively solve the problem by using update strategies for the current variable of the problem until a termination condition is fulfilled. In the following we will briefly discuss some of the most common approaches for nonlinear optimization problems and than have a more detailed look on interior-point methods in section 5.1.1.

Since a wide range of well-developed numerical methods for unconstrained optimization exist [122, 230], one of the most basic approaches to solve a constrained nonlinear optimization problem is to approximate it by a sequence of unconstrained nonlinear optimization problems where a violation of the constraints is penalized. Considering an equality-constrained problem, i.e., $q = 0$, the *penalty function* going back to [64] is defined by

$$P(x, \widetilde{\alpha}) := \phi(x) + \frac{\widetilde{\alpha}}{2}||h(x)||^2,$$

where $\widetilde{\alpha} > 0$ is the so-called *penalty parameter*. Note that for all feasible points $x \in \mathbb{X}$ the penalty function is identical to the objective function ϕ. However, a non-feasible point is penalized by the additional term $\frac{\widetilde{\alpha}}{2}||h(x)||^2$. Consequently, the unconstrained minimum of the penalty function for a fixed penalty parameter $x_P^*(\widetilde{\alpha}^{(i)})$ can be seen as an approximation of the solution of the equality-constrained problem x^*. The approximation improves if the

penalty parameter $\widetilde{\alpha}$ is increased and to assure equality between the two minimizer the limit $\widetilde{\alpha} \to \infty$ has to be analyzed. Note that the general case including inequality constraints can be addressed by adding the penalty term

$$\frac{\widetilde{\alpha}}{2} \sum_{i=1}^{q} \left(\max\{0, g_i(x)\}\right)^2 .$$

For detail on the convergence properties of the penalty methods see, for example, [122, 230].

The idea of the penalty method is to use a sequence of strictly increasing penalty parameters $\widetilde{\alpha}^{(i)}$ and determine a sequence of minimal values $x^*(\widetilde{\alpha}^{(i)})$ for the parameters $\widetilde{\alpha}^{(i)}$ by using the previous value $x_P^*(\widetilde{\alpha}^{(i)})$ as a starting point. However, the condition of the penalty function $P(x, \widetilde{\alpha})$ worsens for an increasing penalty parameter $\widetilde{\alpha}$, which causes numerical problems for the numerical solvers of the unconstrained problem. In addition to the quadratic penalty term from above other *exact penalty terms* exist that reduce this numerical problem at the cost of a non-differentiable penalty function P (cf. [122, 230]).

Note that by construction the penalty methods generates a sequence $x_P^*(\widetilde{\alpha}^{(i)})$ which is infeasible for the original equality constrained problem as long as $x_P^*(\widetilde{\alpha}^{(i)}) \neq x^*$. A related class of methods are the *barrier methods* having an penalty term which is non-zero even for feasible points of the original problem and goes to infinity if the boundary of the feasible set $\mathbb{X}$ is approached. Consequently, all iterates of a barrier methods are strictly feasible for the original problem. Two common barrier functions to couple an g to the cost function ϕ are the *logarithmic barrier function*

$$-\theta \sum_{i=1}^{q} \ln(-g_i(x))$$

and the *inverse barrier function*

$$-\theta \sum_{i=1}^{q} \frac{1}{g_i(x)},$$

where θ is the *barrier parameter*. Since to structure of the barrier functions is unsuitable for considering equality constraints, one possibility is to use a standard penalty term to couple the equality constraints to the objective, which results in a so-called *penalty-barrier approach*.

The general structure of the barrier method is similar to the structure of penalty methods and for a strictly decreasing sequence of barrier parameters related convergence properties can be proven, see for example [230]; however, numerical problems occur likewise for barrier parameters converging to zero. Note that the *primal interior-point methods* in recent research are essentially barrier methods; the more advanced *primal-dual interior-point methods* approximating not only the optimal state but also the corresponding Lagrange multipliers prove to be a suitable class of methods for many nonlinear optimization problems (see section 5.1.1 for more details).

The class of *SQP-methods* is an alternative class of methods for the nonlinear optimization problem and one of the most efficient ones; several well-established optimization methods are based on this class, e.g., `FilterSQP` [107] and `SNOPT` [125]. Note that a large number of variants of the SQP-methods with different advantages exists. One way to motivate SQP-methods is to consider the KKT-conditions for the equality-constrained nonlinear optimization problem and define the corresponding function $J : \mathbb{R}^{n \times p} \to \mathbb{R}^{n \times p}$ by

$$J(x, \mu) := \begin{pmatrix} \nabla_x L_{eq}(x, \mu) \\ h(x) \end{pmatrix},$$

where L_{eq} is the Lagrangian of the equality-constrained nonlinear optimization problem:

$$L_{eq}(x, \mu) := \phi(x) + h(x)^T \mu.$$

If this problem is solved with the nonlinear Newton method, the *Lagrange-Newton method* results and at iteration j with the iteration values $x^{(j)}$ and $\mu^{(j)}$ the Newton system reads

$$DJ\left(x^{(j)}, \mu^{(j)}\right) d_N^{(j)} = -J\left(x^{(j)}, \mu^{(j)}\right),$$

where $d_N^{(j)} \in \mathbb{R}^{n \times p}$ is the corresponding *Newton step*. This equation system can be interpreted as the KKT-conditions of the quadratic optimization problem

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla\phi\left(x^{(j)}\right)^T d + \frac{1}{2} d^T \nabla^2_{xx} L_{eq}\left(x^{(j)}, \mu^{(j)}\right) d \\ \text{subject to} \quad & h\left(x^{(j)}\right) + \nabla h\left(x^{(j)}\right)^T d = 0. \end{aligned}$$

Considering the linearized structure of the equality constraints, the problem can be generalized to nonlinear optimization problems by adding a linearized version of the inequality constraints as an further constraint. This directly leads to the (local) SQP-method: A each iteration of the SQP-method a quadratic approximation of the problem is solved and the values of the variables x, μ and λ are updated accordingly; this is the reason for the term *sequential quadratic programming (SQP)*.

Note that a straightforward modification of the SQP-method is obtained by using approximations of the Hessian of the Lagrangian. Other modifications have to be introduced to obtain a global version of the SQP-method or to avoid problem like infeasible quadratic subproblems or the Maratos effect [207]. Details on this modifications and corresponding convergence proofs can be found in [122, 230].

5.1.1 Interior-Point Methods

The goal of this section is to introduce the basic idea of interior-point methods and their connection to barrier problems. Instead of the general optimization problem A.0.1 a nonlinear problem with simplified inequality constraints is considered in this section to shorten the notation:

Definition 5.1.1. ***(Modified Nonlinear Optimization Problem)***

$$\min_{x \in \mathbb{R}^n} \phi(x) \quad \textit{subject to} \quad h(x) = 0,\ x \geq 0.$$

Note that the problem A.0.1 can be rewritten in the form of 5.1.1 by introducing *slack variables* $s \in \mathbb{R}^q_{\geq 0}$ and separating the positive and negative parts of x:

$$\min_{(x)_+ \in \mathbb{R}^n, (x)_- \in \mathbb{R}^n, s \in \mathbb{R}^q} \phi((x)_+ - (x)_-)$$

$$\begin{aligned} \text{subject to} & \\ h((x)_+ - (x)_-) &= 0, \\ g((x)_+ - (x)_-) + s &= 0, \\ (x)_+ &\geq 0, \\ (x)_- &\geq 0, \\ s &\geq 0, \end{aligned}$$

where the positive part $(x)_+ \in \mathbb{R}^n_{\geq 0}$ is defined by $(x)_+ := \max\{0, x\}$ and the negative part $(x)_- \in \mathbb{R}^n_{\geq 0}$ analog by $(x)_- := \max\{0, -x\}$. Consequently, we can define a variable $\widehat{x}$, a cost function $\widehat{\phi}$ and equality constraints $\widehat{h}$ guaranteeing the structure of the modified problem 5.1.1: the variable $\widehat{x} \in \mathbb{R}^{\widehat{n}}$ is the concatenation of $(x)_+$, $(x)_- \in \mathbb{R}^n$ and $s \in \mathbb{R}^q$, thus $\widehat{n} = 2n + q$. The function $\widehat{h} : \mathbb{R}^{\widehat{n}} \to \mathbb{R}^{\widehat{p}}$ with $\widehat{p} = p + q$ is the combination of all the equality constraints of this reformulation and the definition of the cost function $\widehat{\phi} : \mathbb{R}^{\widehat{n}} \to \mathbb{R}$ is straightforward.

Remark 5.1.2. Linear problems are the starting point for theory on interior-point methods and consequently, there exists a detailed and well-established theory on the linear case and most phenomena are understood [348]. The nonlinear case on the other hand is still an open research area.

There are two main ways to motivate interior-point methods and both viewpoints create different insights: One approach is solving systems of perturbed optimality conditions and analyzing the limit of the results if the perturbation is reduced. Alternatively, one can interpret interior-point methods as barrier methods, which will be the way we start with.

Barrier methods solve a sequence of the following *barrier problems* for a decreasing *barrier parameter* θ, that is finally driven towards zero. The basic assumption of such a barrier approach is that strictly feasible points exist.

Definition 5.1.3. ***(Barrier Problem)***
Given a barrier parameter $\theta \geq 0$ the following problem is to solve:

$$\min_{x \in \mathbb{R}^n} \phi_\theta(x) \quad \textit{subject to} \quad h(x) = 0,$$

where the ***barrier objective*** *is defined by*

$$\phi_\theta(x) := \phi(x) - \theta \sum_{k=1}^{n} \ln(x_k).$$

Remark 5.1.4. Note that the objective ϕ_θ is only defined for $x \in \mathbb{R}^n_{>0}$. Thus, there is the implicit assumption that each element of x is strictly positive.

For a local solution $x^* \in \mathbb{R}^n_{>0}$ of 5.1.3 fulfilling a CQ the following KKT-conditions A.1.7 result:

$$\begin{aligned} \nabla\phi(x) - \theta(\mathcal{V}(x))^{-1}\mathbb{1} + \nabla h(x)\mu &= 0, \\ h(x) &= 0, \end{aligned}$$

where the function $\mathcal{V} : \mathbb{R}^i \to \mathbb{R}^{i \times i}$ maps a vector on the corresponding diagonal matrix, i.e., the (k,k)-element of $\mathcal{V}(x)$ equals $x^{(k)}$ for all $k = 1, \ldots, i$, and the vector $\mathbb{1} \in \mathbb{R}^n$ is a vector of ones.
Introducing $\lambda \in \mathbb{R}^n$ by $\lambda = \theta(\mathcal{V}(x))^{-1}\mathbb{1}$ allows us to rewrite these equations. Note that by using λ the original primal equation system is transferred into a primal-dual system.

$$\begin{aligned} \nabla\phi(x) + \nabla h(x)\mu - \lambda &= 0, \\ h(x) &= 0, \\ \mathcal{V}(x)\mathcal{V}(\lambda)\mathbb{1} - \theta\mathbb{1} &= 0. \end{aligned} \tag{5.1}$$

The implicit assumption is that $x > 0$ is resulting in $\lambda > 0$.

The other approach to introduce interior-point methods is to consider relaxations of the KKT-conditions of the problem 5.1.1. In the end we will obtain a system similar to (5.1), but the perspective is different. The (non-relaxed) KKT-conditions read:

$$\begin{aligned} \nabla\phi(x) + \nabla h(x)\mu - \lambda &= 0, \\ h(x) &= 0, \\ \mathcal{V}(x)\mathcal{V}(\lambda)\mathbb{1} &= 0, \\ x &\geq 0, \\ \lambda &\geq 0. \end{aligned}$$

If a relaxation parameter $\theta > 0$ is used in $\mathcal{V}(x)\mathcal{V}(\lambda)\mathbb{1} = 0$, we end up with a system identical to (5.1) with the additional assumptions $x > 0$ and $\lambda > 0$:

$$\begin{aligned} h_\theta(x, \mu, \lambda \mid \theta) &= 0, \\ x &\geq 0, \\ \lambda &\geq 0, \end{aligned} \tag{5.2}$$

where the function of the equality constraints $h_\theta : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}^{2n+p}$ for a given value of $\theta \in \mathbb{R}_{\geq 0}$ is defined by

$$h_\theta(x, \mu, \lambda \mid \theta) = \begin{pmatrix} \nabla\phi(x) + \nabla h(x)\mu - \lambda \\ h(x) \\ \mathcal{V}(x)\mathcal{V}(\lambda)\mathbb{1} - \theta\mathbb{1} \end{pmatrix}.$$

The scalar $\frac{1}{n}x^T\lambda$ corresponding to the last block of h_θ is called the *complementarity gap* and measures the deviation from the original complementarity conditions.

Naturally, the question arises how the (primal) solutions of the barrier problems $x^*(\theta)$ are related to the solution of the original problem x^*. It can be shown (see e.g. [115]) under some assumptions (including the MFCQ and strict complementarity at x^*) that the solutions $x(\theta)$ for decreasing barrier parameters θ not only converge to x^* but also define a differentiable path. This path is called *barrier trajectory* in the context of barrier problems; we, however, will use the term *central path*, which is introduced in the context of interior-point problems.

Definition 5.1.5. ***(Central Path)***
Let $(x(\theta), \mu(\theta), \lambda(\theta))$ be a solution of the system (5.2) for a given value of the parameter θ. Define the ***central path*** *as the following set:*

$$\mathbb{C} := \{(x(\theta), \mu(\theta), \lambda(\theta)) \mid \theta > 0\}.$$

Note that in the general case the set $\mathbb{C}$ might not be a path in its original sense but just a set of points, because problem (5.2) can have multiple solutions. The concept of the central path is the basis for many interior-point methods, which use regularly steps towards the central path or try to stay within a neighborhood of the central path for all iterates.

A common approach to get an interior-point method solving problem (5.2) is to use the equation $h_\theta(x, \mu, \lambda \mid \theta) = 0$ to compute a search direction for the current position and to use the inequality constraints to determine the step size.

For a given parameter θ the Newton direction $d \in \mathbb{R}^{2n+p}$ of the equation at the point (x, μ, λ) is determined by solving

$$Dh_\theta(x, \mu, \lambda)d = -h_\theta(x, \mu, \lambda). \tag{5.3}$$

The Jacobian of h_θ is given by

$$Dh_\theta(x, \mu, \lambda) = \begin{pmatrix} \nabla_{xx}L(x, \mu, \lambda) & \nabla h(x) & -\mathcal{U} \\ \nabla h(x)^T & 0 & 0 \\ \mathcal{V}(\lambda) & 0 & \mathcal{V}(x) \end{pmatrix}. \tag{5.4}$$

Remark 5.1.6. If the value of θ is kept constant, the Newton direction is a direction towards a point on the central path. Note that in general the complementarity gap is not reduced in this direction, but the individual products $x_k\lambda_k$ tend towards to the mean value. Therefore, this direction is called the *centering direction*.

On the other hand, a decrease of the parameter θ changes the direction towards a point given by the KKT-conditions of 5.1.3 without considering the inequality bounds $x \geq 0$ and $\lambda \geq 0$. The direction corresponding to the parameter $\theta = 0$ is the *affine scaling direction* which consequently would allow to reduce the complementarity gap considerably, but in most cases the inequality constraints require a small step.

The last part missing in the formulation of a generic interior-point algorithm is the termination condition. This condition should take the numerical error in $h_\theta(x, \mu, \lambda \mid \theta) = 0$ and the value of the parameter θ or of the complementarity gap into account.

In the following we will introduce the filter technique used to enhance algorithms of nonlinear optimization and then discuss some details of `IPOPT`, the interior-point optimization method of Wächter and Biegler [333].

5.1.2 Filter Techniques

The filter method introduced by Fletcher and Leyfer [108] is a technique for global convergence proofs of nonlinear optimization. After the first work on linear sequential programming [112] the technique is successfully used in the context of sequential quadratic programming [107], as well. Since the filter approach is not limited to a special technique of nonlinear optimization, it is also used in the context of interior-point methods [313, 332].

Our final goal is to solve problem 5.1.1, but it can be assumed here that the inequality constraints are guaranteed by the numerical technique generating the iterates. Consequently, the following problem is to be considered:

$$\min_{x \in \mathbb{R}^n} \phi(x) \;\text{ subject to }\; h(x) = 0.$$

Remark 5.1.7. In the context of barrier methods the cost function ϕ could be replaced by ϕ_θ without changing the problem structure.

Obviously, the problem has two competing aims: minimizing the cost function and minimizing the violation of the equality constraints. The question is how to quantify whether a new iterate is better than the iterates obtained before. The idea of [108] is to interpret the problem as a bi-criteria optimization problem with the functions ϕ and $\chi : \mathbb{R}^n \to \mathbb{R}$ with

$$\chi(x) := ||h(x)||.$$

This bi-criteria problem naturally has a certain emphasis on the function χ, since we are interested in a feasible point with a low cost value instead of a point minimizing the cost but violating the constraints.

Remark 5.1.8. The filter approach tries to build the *efficient border* of the bi-criteria problem, this means that the *non-dominated* iterates of the optimization run are stored. In multi-criteria optimization a tuple is *dominated* by another if all its elements are larger than or equal to the corresponding element of the other tuple. Note that if one point is dominated by another, it probably is of no further interest for the optimization and can therefore be disregarded.

So far, the filter $\mathbb{F}$ as a set of tuples allows for clustering of iterates at an infeasible point. To avoid the problem of accepting points with marginal distance to points already in the filter, a further acceptance condition is introduced by [107].

Let x be the new iterate and $x^{(k)} \in \mathbb{F}$; the following inequality has to be fulfilled by x to be accepted by the filter in the version of [107]:

$$\max\left\{\phi\left(x^{(k)}\right) - \phi(x), \chi\left(x^{(k)}\right) - \chi(x)\right\} > c_{\mathbb{F}}\chi\left(x^{(k)}\right), \tag{5.5}$$

where the constant $c_{\mathbb{F}} \in (0, \frac{1}{2})$.

Naturally, there are more ways to define additional conditions like (5.5) to guarantee that only iterates with certain properties are added to the filter. These additional conditions and the approaches to generate iterates fulfilling them are closely connected to the techniques used to prove global convergence of the respective algorithm. Since we will use the interior-point algorithm `IPOPT` of Wächter and Biegler [333] in our applications, some details of their variant of the filter technique are discussed in the following (the barrier parameter θ is a given constant throughout this section).

Remark 5.1.9. Two equivalent ways exist to represent a filter: The most common way is to use a set of tuples as it is presented here. In [332] another approach is used for notational reasons, where the filter is identified with the parts of $\mathbb{R}^2_{\geq 0}$ corresponding to non-acceptable points.

In the filter version of [332] a new iterate x is accepted by the filter $\mathbb{F}$ if the following inequality holds for all $x^{(k)} \in \mathbb{F}$:

$$\phi_\theta(x) \leq \phi_\theta\left(x^{(k)}\right) - c_\phi\chi\left(x^{(k)}\right) \quad \text{or} \quad \chi(x) \leq (1 - c_\chi)\chi\left(x^{(k)}\right), \tag{5.6}$$

where $c_\phi \in (0, 1)$ and $c_\chi \in (0, 1)$ are fixed constants.

Remark 5.1.10. Due to the fact that the version of [332] does not add every accepted iterate to the filter (see below for details), the above inequality (5.6) must also hold with respect to the last iterate.

Having discussed the basic layout of the filter, a common detail of the mentioned filter versions [107, 332] is to introduce a *switching condition*, which means that the above filter approach is only used if the condition does not hold.

In the setup of [332] the new (primal) iterates $x^{(j+1)}(\varsigma)$, which are to be tested by the filter, are generated along a line given by the last accepted iterate $x^{(j)}$ and a directional vector $d \in \mathbb{R}^n$ (corresponding to a part of the Newton direction, see section 5.1.3 for details):

$$x^{(j+1)}(\varsigma) := x^{(j)} + \varsigma d,$$

where $\varsigma > 0$. For such an iterate the switching condition reads:

$$\nabla\phi_\theta\left(x^{(j)}\right)^T d < 0 \quad \text{and} \quad \varsigma\left(-\nabla\phi_\theta\left(x^{(j)}\right)^T d\right)^{\overline{c}_\phi} > c_{\mathbb{F}}\left(\chi\left(x^{(j)}\right)\right)^{\overline{c}_\chi}, \tag{5.7}$$

where the constants fulfill $c_{\mathbb{F}} > 0$, $\overline{c}_\chi > 1$ and $\overline{c}_\phi \geq 1$.

Remark 5.1.11. The first condition assures that the linearized version of the cost function ϕ_θ decreases in the direction of d, i.e., d is in particular a descent direction of the objective. The second condition guarantees a large cost reduction (in the linearized model) if the infeasibility of the iterate is large.

This condition is used in the following way: If $\chi(x^{(j)}) \leq \chi_{min}$ for a given $\chi_{min} \in (0, \infty]$ and condition (5.7) is true for $x^{(j+1)}(\varsigma)$, the Armijo condition

$$\phi_\theta\left(x^{(j+1)}(\varsigma)\right) \leq \phi_\theta\left(x^{(j)}\right) + c_A\varsigma\nabla\phi_\theta\left(x^{(j)}\right)^T d \tag{5.8}$$

has to be satisfied instead of (5.6) for $x^{(j+1)}(\varsigma)$ to be acceptable for the filter, where the Armijo parameter $c_A \in (0, \frac{1}{2})$ is constant.

Remark 5.1.12. The motivation for this definition of the switching condition is given by [332]: Condition (5.7) becomes true (under some assumptions assured by the interior-point method, cf. [332]) if a feasible but non-optimal point is approached. In this case enforcing the decrease of the cost function by the Armijo condition (5.8) prevents the method from converging to such a point.

Summing up, the filter approach used by [333] has the following structure: If $\chi(x^{(j)}) > \chi_{min}$ or if condition (5.7) does not hold, the new iterate has to fulfill (5.6) to be accepted by the filter. If the iterate is accepted, it is added to the filter and all dominated tuples can be discarded. On the other hand, if $\chi(x^{(j)}) \leq \chi_{min}$ and condition (5.7) is true, the Armijo condition (5.8) has to hold for acceptance. Note that in case of acceptance the new iterate is not added to the filter.

5.1.3 Optimization Method `IPOPT`

Having introduced the basic idea of interior-point methods and the filter technique, this section discusses the interior-point solver `IPOPT` of Wächter and Biegler [333]. Local and global convergence properties for this solver are proven in [331, 332]. An implementation of the algorithm is available under an open source license and is used in this work to solve the nonlinear optimization problems arising in the application examples.

5.1.3.1 Tolerance for Optimality Error

The primal-dual equations (5.2) are used to define the optimality error which serves as the termination criterion of the algorithm. If the individual parts of h_θ are scaled separately, the following *optimality error* for the barrier problem results:

$$E_\theta(x, \mu, \lambda) \quad := \quad \max\left\{\frac{||\nabla\phi(x) + \nabla h(x)\mu - \lambda||_\infty}{c^s_{E_\theta}},\ ||h(x)||_\infty,\ \frac{||\mathcal{V}(x)\mathcal{V}(\lambda)\mathbb{1} - \theta\mathbb{1}||_\infty}{c^c_{E_\theta}}\right\},$$

with scaling parameters $c^c_{E_\theta} \geq 1$ for the complementarity conditions and $c^s_{E_\theta} \geq 1$ for the stationarity condition. The optimality error for the original problem is denoted by E_0 and corresponds to the definition of E_θ for the barrier parameter $\theta = 0$. If a combination (x^*, μ^*, λ^*) yielding an optimality error not larger than a user-given tolerance $\epsilon_{tol} > 0$ is determined, i.e.

$$E_0(x^*, \mu^*, \lambda^*) \leq \epsilon_{tol},$$

the method `IPOPT` terminates. Using a fixed number $c^{max}_{E_\theta} \geq 1$, the scaling factors

$$c^s_{E_\theta} = \max\left\{c^{max}_{E_\theta}, \frac{||\lambda||_1 + ||\mu||_1}{n+p}\right\} / c^{max}_{E_\theta}$$

and

$$c^c_{E_\theta} = \max\left\{c^{max}_{E_\theta}, \frac{||\lambda||_1}{n}\right\} / c^{max}_{E_\theta}$$

are chosen in [333] to avoid numerical difficulties for large multipliers μ and λ.

5.1.3.2 Outer Iteration

The basic idea of `IPOPT` is to solve a given barrier problem up to a suitable tolerance and then decrease the relaxation parameter θ towards 0. Consequently, one can distinguish between the outer iterations where the relaxation parameter is adapted and the inner iteration where the solution of a barrier problem for fixed parameter θ is computed. In [331, 333] fast local convergence is discussed for this interior-point algorithm based on the approach of [51], where superlinear convergence is shown under second-order sufficient conditions. To obtain such convergence properties, each barrier problem has to be solved up to a specific tolerance and the parameter has to be updated accordingly in the outer iteration.

Given a barrier problem with parameter θ as the current problem of the outer iteration, a solution (x^*, μ^*, λ^*) of the inner iteration has to satisfy

$$E_\theta(x^*, \mu^*, \lambda^*) \leq c_{\epsilon_{tol}}\theta$$

for a given constant $c_{\epsilon_{tol}} > 0$. A new value for the barrier parameter is then determined by

$$\max\left\{\frac{\epsilon_{tol}}{10}, \min\left\{c_\theta\theta, \theta^{c_{bp}}\right\}\right\},$$

where the constants fulfill $c_\theta \in (0,1)$ and $c_{bp} \in (1,2)$. The motivation of [51] for this update rule is on the one hand to eventually get a superlinear decrease in θ and on the other hand to avoid numerical difficulties for parameters which are by far smaller than the user-given final tolerance ϵ_{tol}.

Remark 5.1.13. The procedure of `IPOPT` that separates an outer iteration from an inner iteration can be interpreted from the perspective of an interior-point algorithm in the following way: The inner iterations, which solve the barrier problem for a fixed parameter θ, are steps in the centering direction. Solely changing the barrier parameter the outer iterations can be interpreted as affine scaling steps (cf. remark 5.1.6).

Remark 5.1.14. The strict separation of adapting θ and of optimizing x, μ and λ is not predetermined. For example, the interior-point algorithm of [313] introduces a scalar in $[0,1]$ such that directions similar to the centering and the affine scaling directions are the two extrema. By using such an approach it is possible to choose the scalar depending on the individual situations withour being forced to solve the relaxed problem to a higher accuracy than actually needed.

5.1.3.3 Inner Iteration

Mentioned already in the section 5.1.1 on general interior-point methods, a damped Newtons method is used to solve the barrier problem for a fixed barrier parameter θ up to a tolerance $E_\theta(x^*, \mu^*, \lambda^*) \leq c_{\epsilon_{tol}}\theta$.

Given a current iterate (x, μ, λ) with $x > 0$ and $\lambda > 0$, the search direction d is determined by (5.3); the vector d can be split into the individual segments corresponding to x, μ and λ and these are denoted by d_x, d_μ and d_λ, accordingly. Instead of solving the system with the non-symmetric Jacobian (5.4) a transformed system with a condensed, symmetric matrix is proposed by [333]:

$$\begin{pmatrix} \nabla^2_{xx} L(x,\mu,\lambda) + \mathcal{V}(x)^{-1}\mathcal{V}(\lambda) & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_\mu \end{pmatrix} = -\begin{pmatrix} \nabla \phi_\theta(x) + \nabla h(x)\mu \\ \mu(x) \end{pmatrix}. \quad (5.9)$$

The part d_λ corresponding to the eliminated block row can be computed by

$$d_\lambda = \theta \mathcal{V}(x)^{-1}\mathbb{1} - \lambda - \mathcal{V}(x)^{-1}\mathcal{V}(\lambda)d_x.$$

Remark 5.1.15. To guarantee certain descent properties using the filter technique (see remark 5.1.12) it has to be assured that the projection of the top left block of the matrix in (5.9) onto the null space of the Jacobian $\nabla h(x)^T$ is uniformly positive definite. A method to correct the inertia of the matrix is discussed in [333].

For a given search direction d, the goal of one inner iteration is to determine a step size $\varsigma > 0$ such that the corresponding point is acceptable by the filter (cf. section 5.1.2) and fulfills the constraints $x > 0$ and $\lambda > 0$.

The approach designed to respect the inequality constraints is based on the parameter

$$c_{fb} \in [c_{fb}^{min}, 1) \text{ defined by } c_{fb} = \max\{c_{fb}^{min}, 1 - \theta\},$$

where $c_{fb}^{min} \in (0, 1)$ is its minimal value; consequently, if θ is reduced towards zero, the parameter c_{fb} goes to one. This property is used in the definition of the maximal step size $\varsigma_{max} \in (0, 1]$ and $\varsigma_\lambda \in (0, 1]$:

$$\begin{aligned} \varsigma_{max} &:= \max\{\varsigma \in (0,1] \mid x + \varsigma d_x \geq (1 - c_{fb})x\}, \\ \varsigma_\lambda &:= \max\{\varsigma \in (0,1] \mid \lambda + \varsigma d_\lambda \geq (1 - c_{fb})\lambda\}. \end{aligned}$$

Wächter and Biegler [333] propose to split the step size in the direction of d_x and d_μ from the direction d_λ and use the above defined quantity ς_λ as the step size for the Lagrangian multipliers of the inequality constraints λ. The step size ς common for the other two directions is computed in the interval $(0, \varsigma_{max}]$ by a backtracking line-search procedure.

In some cases the backtracking procedure cannot yield a point that is acceptable to the filter, thus a procedure called the *feasibility restoration phase* is introduced, which tries to find a new iterate acceptable to the current filter.

5.1.3.4 Feasibility Restoration Phase

A minimum step size $\varsigma_{min} > 0$ is introduced to cope with cases where the backtracking line-search has problems finding an acceptable iterate.

$$\varsigma_{min} := \begin{cases} c_\varsigma \min\left\{c_\chi, \frac{c_\phi \chi(x)}{-\nabla\phi_\theta(x)^T d_x}, \frac{c_{\mathbb{F}}(\chi(x))^{\bar{c}_\chi}}{\left(-\nabla\phi_\theta(x^{(j)})^T d\right)^{\bar{c}_\phi}}\right\} \\ \qquad\qquad \text{if} \quad -\nabla\phi_\theta(x)^T d_x < 0 \quad \text{and} \quad \chi(x^{(j)}) \leq \chi_{min}, \\ c_\varsigma \min\left\{c_\chi, \frac{c_\phi \chi(x)}{-\nabla\phi_\theta(x)^T d_x}\right\} \\ \qquad\qquad \text{if} \quad -\nabla\phi_\theta(x)^T d_x < 0 \quad \text{and} \quad \chi(x^{(j)}) > \chi_{min}, \\ c_\varsigma c_\chi \qquad\qquad \text{otherwise,} \end{cases}$$

where $c_\varsigma \in (0, 1]$ is a *safety factor*. Note that this minimum step length is defined by using the linear functions introduced in the filter definitions (compare equations (5.6) and (5.7)).

If the line-search can not find a step length $\varsigma \in (\varsigma_{min}, \varsigma_{max}]$ such that the iterate is acceptable to the filter, the algorithm switches to the *feasibility restoration phase*. In this case a new iterate is sought that is acceptable to the current filter by iteratively reducing the constraint violation. Note that the original cost function ϕ is only indirectly considered via the filter. If no acceptable iterate is found by the restoration phase, which can, for example, happen if the problem is locally infeasible, [333] proposes that a local minimizer of the constraint violation is returned.

Consider the following optimization problem where the constraint violation is minimized with respect to the (primary) inequality constraints while penalizing deviations from the current iterate x:

$$\min_{\varrho \in \mathbb{R}^n} ||h(\varrho)||_1 + \frac{\varpi}{2}||\mathcal{V}(c^{sc})(\varrho - x)||_2^2 \quad \text{subject to} \quad \varrho \geq 0, \tag{5.10}$$

where $\varpi >$ is the penalty factor and the scaling vector c^{sc} is defined by

$$c_k^{sc} := \min\{1, \frac{1}{|x_k|}\}.$$

Remark 5.1.16. The idea of minimizing the violation of the constraints and adding a regularization term measuring the distance of the current iterate can be found in [313], where for another interior-point method a related restoration algorithm is introduced.

In [333] it is proposed to reformulate the optimization problem (5.10) as a smooth optimization problem that has the structure of problem type 5.1.1 and then to use the *normal* interior-point algorithm to solve it. The smooth reformulation reads:

$$\min_{\varrho \in \mathbb{R}^n, (r)_+ \in \mathbb{R}^p, (r)_- \in \mathbb{R}^p} ||(r)_+ + (r)_-||_1 + \frac{\varpi}{2}||\mathcal{V}(c^{sc})(\varrho - x)||_2^2$$

$$\begin{aligned} \text{subject to} \quad & h(\varrho) - (r)_+ + (r)_- = 0, \\ & \varrho \geq 0, \quad (r)_+ \geq 0, \quad (r)_- \geq 0, \end{aligned}$$

where $(r)_+ \in \mathbb{R}^p_{\geq 0}$ and $(r)_- \in \mathbb{R}^p_{\geq 0}$ are the positive and the negative part of the equality constraints. This problem is solved (again) by a sequence of barrier problems. Note that the consequence of the penalty term is usually that a strict local minimum exists.

Naturally, it has to discussed what happens if feasibility restoration is needed inside another restoration phase. This problem is solved in [333] by fixation of ϱ in the respective barrier problem, because $(r)_+$ and $(r)_-$ can then be computed by a quadratic equation (see [333] for details).

5.1.3.5 Convergence Results

If global convergence is ensured for each barrier problem, the global convergence of the overall interior-point method follows. The assumption of a global convergence proof is naturally that the algorithm generates a non-ending sequence of iterates. Consequently, it has to be assumed that the feasibility restoration phase always terminates successfully and that the algorithm does not stop at a point fulfilling the KKT-conditions.

It is shown in [332] that under appropriate assumptions all limit points are feasible. Furthermore, if sequence of iterates $(x^{(k)})$ is bounded, a limit point x^* exists that fulfills the KKT-conditions.

Remark 5.1.17. The approach used in [332] to show global convergence is closely related to the one of the SQP method of [107], which introduces the idea of the filter technique (see section 5.1.2).

Remark 5.1.18. To assure that the assumptions for global convergence are met, it is necessary to introduce a safeguard for the Lagrangian multipliers of the inequality constraints. This means that the values of these multipliers are changed if they become too extreme. See [332, 333] for details.

Summing up, the discussed line-search variant of the filter technique (with a suitable feasibility restoration phase) in combination with the strategy for reducing the barrier parameter in the outer iterations defines the interior-point method of Wächter and Biegler [333]. They show (under reasonable assumptions) that their algorithm guarantees global convergence and fast local convergence (compare section 5.1.3.2).

Remark 5.1.19. Further details of the algorithm `IPOPT` like second-order corrections and accelerating heuristics can be found in [331, 332, 333], but they are not mandatory for understanding the general approach.

5.2 Optimization Method `coreIOC` for Inverse Optimal Control

The inverse optimal control approach introduced in the previous chapters is realized in our program `coreIOC` - the first part of the name (core) is is the combination of the first two letters of the two main principles of the optimization strategy: collocation and relaxation; the second part (IOC) is an abbreviation of inverse optimal control. In this chapter several issues of the numerical computations are addressed, e.g., the scaling of the variables and the functions of the bilevel problem (cf. section 5.2.2), and a framework to evaluate the inversion process using simulated data are presented (see section 5.2.4).

5.2.1 Adaptive Time Discretization

The discretization of the time interval has to be sufficiently fine to assure that the solution of the discretized optimal control problem comes close to the solution of the original optimal control problem. On the other hand using a fine uniform discretization increases the problem

size significantly and consequently, a strategy is needed that assures a reasonable trade-off between the two (cf. section 3.2).

We use a static adaptation strategy which updates the time discretization after having solved the discretized problem for the current discretization; this approach is suitable to be combined with the relaxation strategy discussed in section 2.5.3. Other discretization strategies are known where the time discretization is updated during the optimization of an optimal control problem (e.g., [330]), but this introduces further nonlinearities and additional conditions have to be added to the discretized optimal control problem.

Following the line of [27], a local discretization error is used to introduce an adaptation strategy for the time discretization. Assuming that the equality $x_{(i)} = \overline{x}(t_i)$ holds, the following discretization error results for the interval $[t_i, t_{i+1}]$:

$$\begin{aligned} x_{(i+1)} - \overline{x}(t_{i+1}) &= x_{(i+1)} - \left(x_{(i)} + \int_{t_i}^{t_{i+1}} \varphi\left(\overline{x}(t), \overline{u}(t)\right) \, \mathrm{d}t \right) \\ &= \int_{t_i}^{t_{i+1}} \widetilde{x}'(t) - \varphi\left(\overline{x}(t), \overline{u}(t)\right) \, \mathrm{d}t. \end{aligned}$$

Since the values for $\overline{x}$ and $\overline{u}$ are unknown, they are replaced by the corresponding values of the approximations $\widetilde{x}$ and $\widetilde{u}$. Considering the absolute value of the difference, the following inequality yields a suitable measure for the *(absolute) local discretization error* $\overline{\varepsilon}^{(i)} \in \mathbb{R}^{\overline{n}}_{\geq 0}$, $i = 1, \ldots, \nu$:

$$\left| \int_{t_i}^{t_{i+1}} \widetilde{x}'(t) - \varphi\left(\widetilde{x}(t), \widetilde{u}(t)\right) \, \mathrm{d}t \right| \leq \int_{t_i}^{t_{i+1}} \left| \widetilde{x}'(t) - \varphi\left(\widetilde{x}(t), \widetilde{u}(t)\right) \right| \, \mathrm{d}t =: \overline{\varepsilon}^{(i)}.$$

To compare the local discretization errors of the individual intervals, a reduction to a scalar value simplifies the problem. Due to the different scales for components of $\overline{\varepsilon}^{(i)}$, a scaling of the components is advantageous. In consequence, [27] propose to use the *relative local discretization error* defined by

$$\widehat{\varepsilon}^{(i)} := \max_{j \in \{1, \ldots, \overline{n}\}} \frac{\overline{\varepsilon}_j^{(i)}}{1 + \widehat{\varsigma}_j}, \quad i = 1, \ldots, \nu - 1,$$

where the scaling weights are given by

$$\widehat{\varsigma}_j := \max_{i \in \{1, \ldots, \nu\}} \left\{ \max \left\{ \left| \widetilde{x}_j(t_i) \right|, \left| \widetilde{x}_j{}'(t_i) \right| \right\} \right\}.$$

The goal of the adaptation strategy is to refine a given discretization by subdividing the intervals with a large relative local discretization error. The most simple approach is to bisect these intervals, but, if one wants to add more than one intermediate discretization point, the error reduction has to be predicted. Consequently, the type of discretization strategy influences the choice of the time discretization.

We assume that the underlying Runga-Kutta method assures a consistency of order $\kappa > 0$, which means that the discretization error of one integration step is $\mathcal{O}\left(\delta_i (\Delta)^{\kappa+1}\right)$. Note that the consistency of order κ results in a convergence of order κ (cf. [78, 137]). If constraints have to be fulfilled in addition to the ODE, a problem with a differential-algebraic equation (DAE) results and the standard theory on initial value problems for ordinary differential equations has to be adapted [41]. It has to be noted that DAEs lead to order reductions and the value of the reduction differs between different variables and varies in time if optimal control problems

are considered [27, 41]. In consequence, the reduction values $r_i \in \mathbb{N}$, $i = 1, \ldots, \nu-1$, fulfilling $0 \leq r_i \leq \kappa$ are used to model the behavior of the discretization error as proposed in [27]:

$$\overline{\varepsilon}^{(i)} = \mathcal{O}\left(\delta_i (\Delta)^{\kappa+1-r_i}\right).$$

If the correct reduction value was known, this model could be used to predict how the discretization error is reduced if the interval $[t_i, t_{i+1}]$ is divided into $S_i \in \mathbb{N}$ smaller ones of equal size.

Since we do not have a-priori information on the reduction, the values have to be approximated by analyzing previous adaptations of the time discretization. Therefore assume that the i-th interval with discretization error $\widehat{\varepsilon}^{(i)}$ is divided into S_i subintervals and that the discretization errors $\underline{\varepsilon}_j^{(i)} \in \mathbb{R}^{S_i}$ are obtained for the finer time discretization. Considering the mean discretization error $\underline{\varepsilon}_m^{(i)}$ defined by

$$\underline{\varepsilon}_m^{(i)} := \frac{1}{S_i} \sum_{j=1}^{S_i} \underline{\varepsilon}_j^{(i)},$$

the following approximation for the reduction value results:

$$r_i \approx \kappa + 1 - \frac{\log\left(\overline{\varepsilon}^{(i)} / \underline{\varepsilon}_m^{(i)}\right)}{\log\left(1 + S_i\right)}.$$

Note that r_i has to be an non-negative integer value smaller than or equal to κ and consequently, the nearest integer to the right-hand side is chosen within the bounds.

The goal of the adaptation strategy of [27] is to determine the optimal values for S_i, $i = 1, \ldots, \nu-1$, such that the maximal discretization error is minimized. This integer optimization problem has to be solved under the constraints that maximal $\widehat{N}_{max}$ new points are added to the overall discretization and that each individual interval is divided into maximal $\underline{N}_{max}$ subintervals for given constants $\widehat{N}_{max}$ and $\underline{N}_{max} \in \mathbb{N}$.

This problem can be solved iteratively: Start with the current time discretization and initialize the predicted discretization errors for the new time discretization with the computed ones. In each iteration determine the interval with the largest predicted discretization error; then increase the corresponding S_i by 1 and update the predicted discretization errors for this interval. Terminate if the maximal values $\widehat{N}_{max}$ and $\underline{N}_{max}$ are reached, otherwise proceed to the next iteration.

5.2.2 Scaling

A central aspect when solving nonlinear optimization problems is to determine a suitable formulation of the problem which means that scaling factors for the optimization variables, the constraints and the cost function have to be introduced. Such scaling factors can increase the performance of the numerical solvers considerably and are in most cases as important as the choice of the technique to determine the derivative information.

Naturally, three different strategies to use the scaling approach can be distinguished: The first one is the a-priori strategy, where the scaling factors are computed before the computation starts. This approach allows the determination of the scales without a hard time-constraint and therefore advanced techniques can be used, but due to mostly insufficient information

about the problem at hand, i.e., the local behavior of the solution, the applicability is limited. The second strategy updates the scaling factors during the optimization in dependence on the characteristics of the current iterate. For such an online approach efficient routines are needed to avoid a decrease in the solver performance. A third strategy can be used if a problem has to be solved repeatedly with minor changes in the problem parameters: The a-posteriori approach uses the solution of one optimization run to determine scaling factors fitting to solution characteristics which results in a good solver performance and precise solutions.

Note that the MPEC structure of the discretized inverse optimal control problems results in a sequence of nonlinear optimization problems with slightly varying relaxation parameters if the solution technique of chapter 2 is used. In consequence, the a-posteriori approach seems suitable to use the information of one relaxation to increase the solver performance for the next relaxation.

We start the introduction of the scaling approach for the general constrained nonlinear optimization problem A.0.1 and in a second step detailed scaling strategies are discussed that utilize the characteristic structure of the discretized optimal control problem or the transformed problem of inverse optimal control (cf. chapter 4). All scaling factors are assumed to be strictly positive values and the resulting scaling matrices are considered to be of a diagonal structure. Scaled quantities or functions are denoted by an index s.

Let $S_x \in \mathbb{R}^{n\times n}$, $S_h \in \mathbb{R}^{p\times p}$ and $S_g \in \mathbb{R}^q$ be positive definite matrices and K_x be an element of $\mathbb{R}^n$, then the scaled variable x_s and the scaled constraints h_s and g_s are given by

$$\begin{aligned} x_s &:= S_x x + K_x, \\ h_s(x_s) &:= S_h h({S_x}^{-1}(x_s - K_x)), \\ g_s(x_s) &:= S_g g({S_x}^{-1}(x_s - K_x)). \end{aligned}$$

For the objective function ϕ a scaling factor $S_\phi > 0$ is introduced:

$$\phi_s(x_s) := S_\phi \phi({S_x}^{-1}(x_s - K_x)).$$

The scaling factors for the adjoint variables μ and λ are chosen such that the Lagrangian of the scaled problem is a scalar multiple of the Lagrangian of the original problem:

$$\begin{aligned} \mu_s &:= S_\phi {S_h}^{-1}\mu, \\ \lambda_s &:= S_\phi {S_g}^{-1}\lambda, \end{aligned}$$

which results in

$$\begin{aligned} L_s(x_s, \lambda_s, \mu_s) &:= \phi_s(x_s) + \lambda_s^T g_s(x_s) + \mu_s^T h_s(x_s) \\ &= S_\phi \phi({S_x}^{-1}(x_s - K_x)) + \lambda_s^T S_g g({S_x}^{-1}(x_s - K_x)) + \mu_s^T S_h h({S_x}^{-1}(x_s - K_x)) \\ &= S_\phi \phi(x) + \left(S_\phi {S_g}^{-1}\lambda\right)^T S_g g(x) + \left(S_\phi {S_h}^{-1}\mu\right)^T S_h h(x) \\ &= S_\phi(\phi(x) + \lambda^T g(x) + \mu^T h(x)) = S_\phi L(x, \lambda, \mu). \end{aligned}$$

Note that this scaling approach results in additional factors for derivatives of scaled functions, for example:

$$\begin{aligned} Dh_s(x_s) &= S_h Dh(x) {S_x}^{-1}, \\ \nabla^2_{x_s x_s} L_s(x_s, \lambda_s, \mu_s) &= S_\phi {S_x}^{-1} \nabla^2_{xx} L(x, \lambda, \mu) {S_x}^{-1}. \end{aligned}$$

Since the variables and the constraints of the discretized optimal control problem and the transformed inverse optimal control problem exhibit the structure resulting from the discretization strategy and the transformation technique, it is reasonable to determine scaling factors that are consistent with the underlying structure. The goal of our scaling approach is to provide a scaled problem to the optimization method `IPOPT` where the state variables are approximately within the interval $[-1, 1]$, the value of the cost function within $[0, 100]$ and constraints in the range of $[-1, 1]$.

The lower level state $\underline{x}$ (see section 4.4) combines the states $x_{(i)}$ and the controls $u_{(i)}$ of the discretized optimal control problem (cf. chapter 3). We are interested in a appropriate scaling of the optimal control problem rather than a scaling of the individual states $x_{(i)}$ and controls $u_{(i)}$. Therefore, we define the maximal and minimal values of the states and controls by

$$\begin{aligned} x_{max} &= \max\{x_{(i)} \mid i = 1, \ldots, \nu\}, \\ x_{min} &= \min\{x_{(i)} \mid i = 1, \ldots, \nu\}, \\ u_{min} &= \max\{u_{(i)} \mid i = 1, \ldots, \nu\}, \\ u_{min} &= \max\{u_{(i)} \mid i = 1, \ldots, \nu\}, \end{aligned}$$

where the max- and min-operations are executed componentwise, and obtain the following scaling matrices $S_x \in \mathbb{R}^{\overline{n} \times \overline{n}}$ and $S_u \in \mathbb{R}^{\overline{m} \times \overline{m}}$ and the shifting vectors $K_x \in \mathbb{R}^{\overline{n}}$ and $K_u \in \mathbb{R}^{\overline{m}}$:

$$\begin{aligned} (S_x)_{ij} &:= \begin{cases} \frac{2}{(x_{max})_i - (x_{min})_i} & \text{if } \; i = j, \\ 0 & \text{else} \end{cases}, \\ (K_x)_i &:= 1 - \frac{2\,(x_{max})_i}{(x_{max})_i - (x_{min})_i}, \end{aligned}$$

for the indices $i, j = 1, \ldots, \overline{n}$ and

$$\begin{aligned} (S_u)_{ij} &:= \begin{cases} \frac{2}{(u_{max})_i - (u_{min})_i} & \text{if } \; i = j, \\ 0 & \text{else} \end{cases}, \\ (K_u)_i &:= 1 - \frac{2\,(u_{max})_i}{(u_{max})_i - (u_{min})_i}, \end{aligned}$$

for the indices $i, j = 1, \ldots, \overline{m}$. Consequently, the scaling matrix $S_{\underline{x}} \in \mathbb{R}^{n \times n}$ is given by a block diagonal matrix and the shifting vector $K_{\underline{x}} \in \mathbb{R}^n$ by the corresponding concatenation of shift vectors:

$$S_{\underline{x}} := \begin{pmatrix} S_x & & & & \\ & S_u & & & \\ & & \ddots & & \\ & & & S_x & \\ & & & & S_u \end{pmatrix} \quad \text{and} \quad K_{\underline{x}} := \begin{pmatrix} K_x \\ K_u \\ \vdots \\ K_x \\ K_u \end{pmatrix}.$$

The equality constraints for the discretized optimal control problem h are a combination of the discretized ordinary differential equation for the state variable and the boundary conditions

(cf. section 4.4). Consequently, the following block-diagonal scaling matrix is proposed:

$$S_h := \begin{pmatrix} S_x & & & & & \\ & \ddots & & & & \\ & & S_x & & & \\ & & & D_1 b(x_{(1)}, x_{(\nu)}, y) S_x & & \\ & & & & D_2 b(x_{(1)}, x_{(\nu)}, y) S_x \end{pmatrix} \in \mathbb{R}^{q \times q},$$

where D_i denotes the Jacobian with respect to the i-th argument. Note that the scaling factors of the boundary conditions are defined such that a boundary condition prescribing a given value for a given state is scaled according to the variable scaling of this state.

The approach to determine suitable scaling factors $S_g \in \mathbb{R}^{p \times p}$ for inequality constraints g (if existent) is similar to the one of the equality constraints discussed in detail above. Since the inequality constraints have to be considered at each time instance of the discretization, a maximal absolute value over all these instances can be used to (approximately) scale each element of g to the interval $[-1, 0]$.

Note that the scaling matrices for the transformed inverse optimal control problem 4.4.3 are combinations of the matrices introduced above for the lower level problem. The scaling factors for the LLP state and the adjoint variables are given above and the ULP variables are assumed to be bounded such that a transformation to the interval $[-1, 1]$ is trivial. Therefore, we consider the equality constraints H and the inequality constraints G to be suitably scaled. A consistent scaling of the ULP equality constraints $\widetilde{h}$ and the ULP inequality constraints $\widetilde{g}$ is then given by

$$S_{\widetilde{h}} = \begin{pmatrix} S_{\underline{x}}^{-1} & & \\ & S_h & \\ & & \mathcal{U} \end{pmatrix} \quad \text{and} \quad S_{\widetilde{g}} = \begin{pmatrix} S_g & & \\ & S_\phi S_g^{-1} & \\ & & \mathcal{U} \end{pmatrix},$$

where the identity matrices $\mathcal{U}$ in both cases are of appropriate dimensions.

5.2.3 Goal Attainment

The number of iterations needed to compute a solution of the transformed inverse optimal control problem 4.4.3 depends critically on the choice of the initial values. Furthermore, the optimization method IPOPT computes only stationary points and in consequence an obtained minimum is rather a local one than a global one. Thus the choice of the initial values influences whether the result is close to the global minimum or not. In [5] we introduce a strategy to considerably speed up the computations and to avoid local minima with a large distance to the global one. For a numerical example see section 6.6.3 and the publication [5].

Observing that the upper level cost function $\overline{\Phi}$ is non-negative and ideally close to zero at the global minimizer, a further constraint can be added to problem 4.4.3:

$$\overline{\Phi}(\overline{x}^*, \overline{u}^*, \Lambda^d) = 0.$$

Note that a solution determined by IPOPT would fulfill this additional constraint only up to the user-given solution accuracy $\epsilon_{tol} > 0$. However, if the condition $\overline{\Phi} \leq \epsilon_{tol}$ does not hold true at the global minimum, the constraint can be relaxed by requiring an upper level cost value to be smaller than a suitable number approximating the optimal cost value.

In the following a brief analysis of the effects resulting from this modification on the operation of the algorithm `IPOPT` are stated. Since the additional constraint introduces a linear dependence in the KKT-conditions of the reformulated problem, the constraint is only added at the beginning of the optimization and dropped later on.

The equations determining the Newton direction in the course of the interior-point method are given by equations (5.3) and (5.4). If the additional constraint is added to the problem, the following systems results:

$$\begin{pmatrix} \nabla^2_{xx} L(x,\mu,\lambda) + \rho \nabla^2_{xx} \Phi(x) & \nabla h(x) & -\mathcal{U} & \nabla \Phi(x) \\ \nabla h(x)^T & 0 & 0 & 0 \\ \mathcal{V}(\lambda) & 0 & \mathcal{V}(x) & 0 \\ \nabla \Phi(x)^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_\mu \\ d_\lambda \\ d_\rho \end{pmatrix}$$

$$= - \begin{pmatrix} \nabla \phi(x) + \nabla h(x)\mu - \lambda + \rho \nabla \phi(x) \\ h(x) \\ \mathcal{V}(x)\mathcal{V}(\lambda)\mathbb{1} - \theta \mathbb{1} \\ \phi(x) \end{pmatrix}, \tag{5.11}$$

where $\rho \in \mathbb{R}$ is the Lagrange multiplier corresponding to the additional constraint. The last block row gives

$$\nabla \Phi(x)^T d_x = -\phi(x),$$

which assures that for $\phi(x) > 0$ the search direction d_x is a descent direction of ϕ at x. The system of equations (5.11) can be interpreted as a modification of the original system (5.3) in the following two ways: On the one hand, the first block row can be written as

$$\left(\nabla^2_{xx} L(x,\mu,\lambda) + \rho \nabla^2_{xx} \Phi(x)\right) d_x + \nabla h(x) d_\mu - d_\lambda = -\left((1 + \rho + d_\rho)\nabla \phi(x) + \nabla h(x)\mu - \lambda\right).$$

This can be viewed as the primal-dual equations of the objective function

$$(1+\rho)\phi(x) + d_\rho \nabla \phi(x)^T \left(x - x^{(k)}\right)$$

instead of the original $\phi(x)$. In this case, the value of d_ρ would have to guarantee that d_x is a descent direction. On the other hand, one can also interpret (5.11) as a modification of the original system (5.3) in the top-left block only. To cancel the additional term $\nabla \phi(x) d_\rho$ on the left and $-\rho \nabla \phi(x)$ on the right of the first block row of (5.11), the last row multiplied by $c_\rho \nabla \phi(x)$ is added to the first block row, where $c_\rho \in \mathbb{R}$ is defined by $c_\rho = -\frac{\rho + d_\rho}{\phi(x)}$. In consequence, the only modification compared to the original system is in the top-left block where the Hessian of the Lagrangian is replaced by

$$\nabla^2_{xx} L(x,\mu,\lambda) + \rho \nabla^2_{xx} \Phi(x) + c_\rho \nabla \Phi(x) \nabla \Phi(x)^T.$$

Here, the factor c_ρ, which implicitly depends on d_ρ, has to be chosen such that d_x satisfies $\nabla \Phi(x)^T d_x = -\phi(x)$. Summing up, the two transformations show that the system with the added constraint can be interpreted as a modification of the original system, preserving the general structure of it. See section 6.6.3 for a numerical example of the goal attainment approach; a more detailed study can be found in [5].

5.2.4 Reconstruction Tests

The following framework is used in the subsequent numerical examples to evaluate whether the proposed inverse optimal control approach is able to find a solution close to the global optimal one. Since the cost function optimized in human motions is seldom known in case of real human data, the inversion is tested on simulated values. This means that a vector of upper level parameters y is chosen and the corresponding optimal control problem is solved. Given this solution, data values are generated by adding a suitable amount of noise to the computed values; this noise should assure that no artificial behavior of the optimization method happens due to a perfect matching of data and model. Using a different vector of upper level parameters, starting values for the inverse optimal control approach can be computed. Starting from these values the performance of the inverse optimal control approach can be evaluated by comparing the optimization result to the parameter vector used to generate the data. Since the goal is to identify the original parameters given a different starting value, we term this framework a reconstruction process.

Numerical results of applying the reconstruction framework to different inverse optimal control problems can for example be found in the sections 6.6.1, 6.6.3 and 8.5.1.

Human Arm Movements

Chapter 6

Central aspects of the presented inverse optimal control approach are developed alongside different problems arising in the context of human arm movements. Consequently, this application scenario being the most complex of the three examples of this work is discussed in detail. At the beginning of this chapter a short overview of the state of the art on human arm motions is presented. Since this research field has many facets and the different disciplines interested in this topic have different perspectives, a more detailed introduction to the state of the art is given in the appendix B.

The introduction in section 6.1 is followed by the family of cost functions (section 6.4) and the discussion of the dynamical models of the human arm that are used in this work. As a starting point a planar arm model (section 6.5.1) is derived by combining dynamical models of the bones (section 6.2) and several lumped muscles (section 6.3) and later in section 6.5.2 a generalization to three dimensions is given. The description of the models is accompanied by the discussion of realized human experiments and the presentation of the respective optimization results.

6.1 Introduction to Human Arm Motions

Research on human arm motions of various disciplines differs considerably, since there are certain differences in the perspectives and goals. For example, psychologist and biologists try to determine the underlying principles of human movements, in contrast mathematicians and physicists try to describe the observed movements rather than explaining them [89]. Hence, the minimum principles we obtain by the bilevel optimization approach have to be understood as a description of the human motion and there might be various arguments why the optimal cost function describing the human movements is not biologically or psychologically plausible.

The diverse approaches in the field of human motions can be divided into two classes: model-free theories and model-based theories. In many cases models are successfully used, but, as pointed out in [235], these models are theoretical constructs and should not be identified with the phenomenon they describe, because in several cases model-free approaches can reproduce the same characteristics, e.g., [101, 211]. Consequently, we have to keep in mind that our model-based approach might not be the only way to obtain good approximations of human motion and as Hogan and Flash summarized: “Theories are not immutable truths, but mental constructions that must evolve to accommodate new data” [159].

The first characteristics of human arm motions observed in literature are the shapes of the trajectories and the velocity profiles [2, 218] for planar tasks. In consequence, dynamical

models for the arm and suitable cost functions are introduced capturing the main characteristics by using an optimal control approach [104, 314]. This leads to the discussion whether humans plan their motions in internal coordinates (e.g., the joint angles of the arm joints) or in external coordinates (e.g., the hand position), see section 6.4 for details. Several experiments are discussed in literature to address the question how humans adapt their motions to changes in the environment and what kind of feedback is used. However, it is observed in all human experiments that a characteristic variance between different trials exists. This variation is attributed to impedance characteristics of the human arm and the noise characteristics of the human muscles. Several statistical relations describing the effects of noise are known, e.g., Fitts' law [96] or the two-thirds power law [328]. Refer to section B.2 for a more elaborated discussion of motion characteristics.

A central question regarding human motion control is whether humans use internal models capturing the input-output characteristics of the human motor plant to control their movements or not; which resulted in a long-standing discussion (see section B.3 details). Even if one assumes that internal models are used, usage of forward models predicting the arm motion for a specified control has to be distinguished from inverse models where a desired state is mapped to the corresponding control. Experimental results supporting both types of models are discussed in literature, cf., [76, 175], and consequently, frameworks with multiple internal models of both types have been introduced, e.g., [347]. If one interprets different cost functions in an optimal control problem with given dynamics and boundary conditions as different models, the combination of these cost functions would correspond to the multiple internal model idea and depending on the motion task the combinations could be adapted accordingly.

The problem of planning a motion on the basis of internal models is closely related to the question of human motion generation. Since many tasks of human arm motion can be achieved in different ways, e.g., there might exist several arm configurations yielding the same hand position, the motion problems are redundant problems [25] and therefore suitable principles have to be discussed which describe the motions of these additional degrees of freedom. The two main principles discussed in literature for controlling human arm motions are the following ones: The approach being close to the engineering perspective is that muscle forces are generated in order to control the arm motion [161]. We will use this perspective to build a dynamical model of the human arm in our optimal control framework. The second approach is the equilibrium point hypothesis which unifies posture and motion by using the stability properties of the human arm [158]. For further details see section B.4.2.

To capture human adaptation and learning strategies, different kinds of feedback have to be considered in the models depending on the motion task. If the feedback is incorporated during the motion, a closed-loop framework has to be considered. However, the derivation of optimal control strategies in such a framework is by far more complex than in the open-loop case [307]. A discussion of open-loop and closed-loop ideas can be found in section B.5 and the implication on adaptation behavior in section B.6. In most of our application scenarios we assume that the human is adapted to the current task and that "the sensorimotor control is best described as being near optimal" [308]. Consequently, we use an open-loop approach to model the human motion problem and extend this to the model predictive control approach if certain modification of the task happen during execution of the motion.

As sketched above (more details can be found in the appendix B) several models exist that describe human motions as the product of an optimization process. Therefore, we use our inverse optimal control approach as a tool to find the cost function out of a given family of

cost functions that minimizes the distance between the recorded human data and the respective outputs of the optimal control framework. In [89] it is emphasized from the biological perspective that the minimization criterion is a purely descriptive tool capturing recorded data. In consequence, a cost function is only valuable from the perspective of biology if it can predict human motions for different tasks.

6.2 Rigid Body Models

In this section the dynamics of rigid bodies are introduced with the goal of obtaining an ODE-model of the dynamics of the human bones. Rigid bodies seem to be a good approximation of human bones, because only minor non-rigid effects are observed in human arm motions of daily life. Before stating strategies to derive the differential equations of a systems of rigid bodies, a notation to model a chain of rigid bodies has to be introduced.

6.2.1 Denavit-Hartenberg notation

Several conventions have been introduced in literature to describe the structure of a kinematic chain of rigid bodies. A common one is the Denavit-Hartenberg notation which will be used here. Even under this name two variants exist: the original notation of Denavit and Hartenberg [143] (in the following referred to as the *classical DH notation*) and the modified notation introduced by Craig [65] (called here the *modified DH notation*). Since both approaches have advantages for different structures of chains of rigid bodies, both will be presented in the following. The main point is that for a given set of parameters one has to know which notation is applied, but if the correct definition of the transformation between coordinate systems is identified, both notations can be used to derive the dynamical equations.

The chain of rigid bodies has the following structure: A unique sequence of N links starting with a base element (link 0) and ending with a link which has no successor, link N. All links in-between are connected to the two adjoining links by joints. Without loss of generality, assume that each joint is rotational and has one degree of freedom. Note that a joint having $\widehat{m}$ degrees of freedom can be modeled as $\widehat{m}$ joints of one degree of freedom which are connected by links of length zero. The goal of the DH conventions is to define for each joint a frame, i.e., a local coordinate system, and to describe the transformation between two successive coordinate systems by a stereotypic sequence of rotations and translations. The parameters of the transformations are called the *DH parameters* and the kinematic chain is uniquely characterized by its DH parameters.

Homogeneous coordinates, a standard tool in computer graphics, are used here to simplify notation; they allow to write rotation and translation of a point $\mathcal{P} \in \mathbb{R}^3$ as matrix operations by using the following representation: $\widehat{\mathcal{P}} = \left(\mathcal{P}^T, 1\right)^T$. Let $\mathcal{X}$ be the first vector of the basis of $\mathbb{R}^3$, then for example a translation along $\mathcal{X}$ of length l_0 and a rotation about $\mathcal{X}$ of value ρ_0 are given by

$$\widehat{\mathcal{T}}(\mathcal{X}, l_0) = \begin{pmatrix} 1 & 0 & 0 & l_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \widehat{\mathcal{R}}(\mathcal{X}, \rho_0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\rho_0) & -\sin(\rho_0) & 0 \\ 0 & \sin(\rho_0) & \cos(\rho_0) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The basic idea of the DH conventions is to use the structure given by the rotation axes and the common normals of two successive rotation axes. Hence, let the $\mathcal{Z}$-axis $\mathcal{Z}_{<i>}$ of frame i be coincident with the rotation axis of joint i for all $i \in \{0, \ldots, N\}$. If the two axes $\mathcal{Z}_{<i>}$ and $\mathcal{Z}_{<i-1>}$ are not located in the same plane, a common normal is uniquely defined. The classical DH notation differs from the modified version with regard to which $\mathcal{X}$-axis should be aligned to this normal: The classical choice is $\mathcal{X}_{<i>}$ whereas the modified variant uses $\mathcal{X}_{<i-1>}$ (see the following sections). Note that the respective axis intersects both $\mathcal{Z}$-axes. The third axis $\mathcal{Y}_{<i>}$ follows from $\mathcal{X}_{<i>}$ and $\mathcal{Z}_{<i>}$ by choosing the frame i to be a right-handed coordinate system whose origin is at the intersection of $\mathcal{X}_{<i>}$ and $\mathcal{Z}_{<i>}$.

A few special cases have to be mentioned in order to have a well-defined system. If the two axes $\mathcal{Z}$-axes intersect, the origin of frame i is placed at the intersection and the corresponding $\mathcal{X}$-axis is chosen to be normal on the plane. In case of parallel axes the remaining freedom in the choice of the origin is normally solved by setting the corresponding free DH parameter to zero. The origin of the base frame can be chosen arbitrarily on the $\mathcal{Z}$-axis of frame 0 and also the $\mathcal{X}$- and $\mathcal{Y}$-axes of this right-hand frame can be appointed conveniently; note that for many scenarios there exists a natural choice for this world coordinate system. Finally, the coordinate system of the free end of the last link, termed hand or end effector depending on the kinematic chain, can be chosen arbitrarily.

In the following two sections the Denavit-Hartenberg parameters for both notations are introduced and the resulting transformation matrix is given.

6.2.1.1 Classical DH notation

This classical DH notation is the representation originally introduced by Denavit and Hartenberg [143]. They align the $\mathcal{X}$-axis of frame i with the common normal of $\mathcal{Z}_{<i>}$ and $\mathcal{Z}_{<i-1>}$. With this choice the frames, i.e., the coordinate systems at each joint, are defined and the kinematic properties of the chain of rigid bodies can be described by sets of the following four DH parameters:

β_{i-1}: distance from $\mathcal{X}_{<i-1>}$ to $\mathcal{X}_{<i>}$ measured about $\mathcal{Z}_{<i-1>}$,

ϑ_{i-1}: angle between $\mathcal{X}_{<i-1>}$ and $\mathcal{X}_{<i>}$ measured about $\mathcal{Z}_{<i-1>}$ in the right-hand sense,

l_{i-1}: distance from $\mathcal{Z}_{<i-1>}$ to $\mathcal{Z}_{<i>}$ measured about $\mathcal{X}_{<i>}$,

ρ_{i-1}: angle between $\mathcal{Z}_{<i-1>}$ and $\mathcal{Z}_{<i>}$ measured about $\mathcal{X}_{<i>}$ in the right-hand sense.

This leads to the following definition of the homogeneous transformation matrix ${}^{<i+1>}\mathcal{T}_{<i>}$ describing the transformation of a vector written relative to frame i to being relative to frame $i+1$. Note that by transforming one frame into the other each of the operations is applied with the opposite sign to vectors represented in these frames.

$$
\begin{aligned}
{}^{<i+1>}\mathcal{T}_{<i>} &:= \widehat{\mathcal{R}}(\mathcal{X}_{<i+1>}, -\rho_i)\, \widehat{\mathcal{T}}(\mathcal{X}_{<i+1>}, -l_i)\, \widehat{\mathcal{R}}(\mathcal{Z}_{<i>}, -\vartheta_i)\, \widehat{\mathcal{T}}(\mathcal{Z}_{<i>}, -\beta_i) \\
&= \begin{pmatrix}
\cos\vartheta_i & \sin\vartheta_i & 0 & -l_i \\
-\sin\vartheta_i \cos\rho_i & \cos\vartheta_i \cos\rho_i & \sin\rho_i & -\beta_i \sin\rho_i \\
\sin\vartheta_i \sin\rho_i & -\cos\vartheta_i \sin\rho_i & \cos\rho_i & -\beta_i \cos\rho_i \\
0 & 0 & 0 & 1
\end{pmatrix} \\
&=: \begin{pmatrix}
{}^{<i+1>}\mathcal{R}_{<i>} & {}^{<i+1>}\mathcal{P}_{<i>} \\
0 & 1
\end{pmatrix}.
\end{aligned}
$$

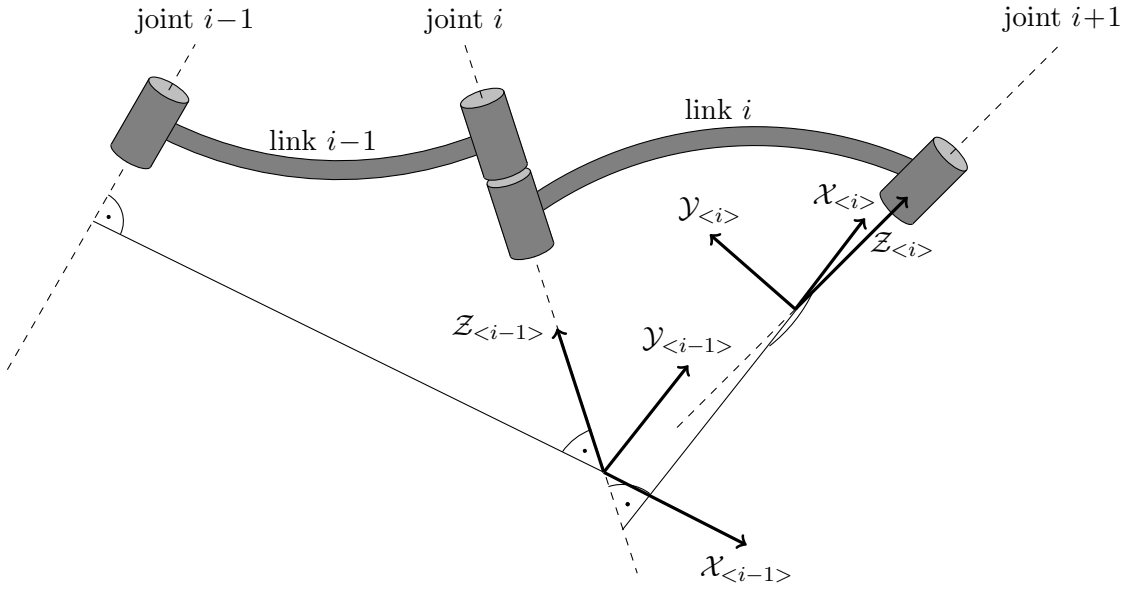


Figure 6.1: Frames of classical DH notation.

The leading superscript denotes according to which frame the values are stated. The rotation matrix ${}^{<i+1>}\mathcal{R}_{<i>} \in \mathbb{R}^{3\times3}$ and the translation vector ${}^{<i+1>}\mathcal{P}_{<i>} \in \mathbb{R}^3$ can be used to transform a vector ${}^{<i>}\mathcal{P} \in \mathbb{R}^3$ from frame i into one corresponding to frame $i+1$:

$$ {}^{<i+1>}\mathcal{P} = {}^{<i+1>}\mathcal{R}_{<i>}\, {}^{<i>}\mathcal{P} + {}^{<i+1>}\mathcal{P}_{<i>}, $$

which corresponds to ${}^{<i+1>}\widehat{\mathcal{P}} = {}^{<i+1>}\mathcal{T}_{<i>}\, {}^{<i>}\widehat{\mathcal{P}}$ in homogeneous coordinates.

6.2.1.2 Modified DH notation

The DH notation to be introduced now is the version of Craig [65], which is used in literature dealing with manipulator dynamics. This version can be interpreted as the inverse approach of the classical one, since the according parameter choices result if the classical approach is started at the end effector and then proceeded towards the basis. Here the $\mathcal{X}$-axis of frame $i-1$ corresponds to the common normal of the two rotation axes $\mathcal{Z}_{<i>}$ and $\mathcal{Z}_{<i-1>}$. Again, the frames are determined by using this choice and the following four DH parameters can be used to describe the transformation between the frames:

l_i: the distance from $\mathcal{Z}_{<i-1>}$ to $\mathcal{Z}_{<i>}$ measured along $\mathcal{X}_{<i-1>}$,

ρ_i: angle between $\mathcal{Z}_{<i-1>}$ and $\mathcal{Z}_{<i>}$ measured about $\mathcal{X}_{<i-1>}$ in the right-hand sense,

β_i: distance from $\mathcal{X}_{<i-1>}$ to $\mathcal{X}_{<i>}$ measured about $\mathcal{Z}_{<i>}$,

ϑ_i: angle between $\mathcal{X}_{<i-1>}$ and $\mathcal{X}_{<i>}$ measured about $\mathcal{Z}_{<i>}$ in the right-hand sense.

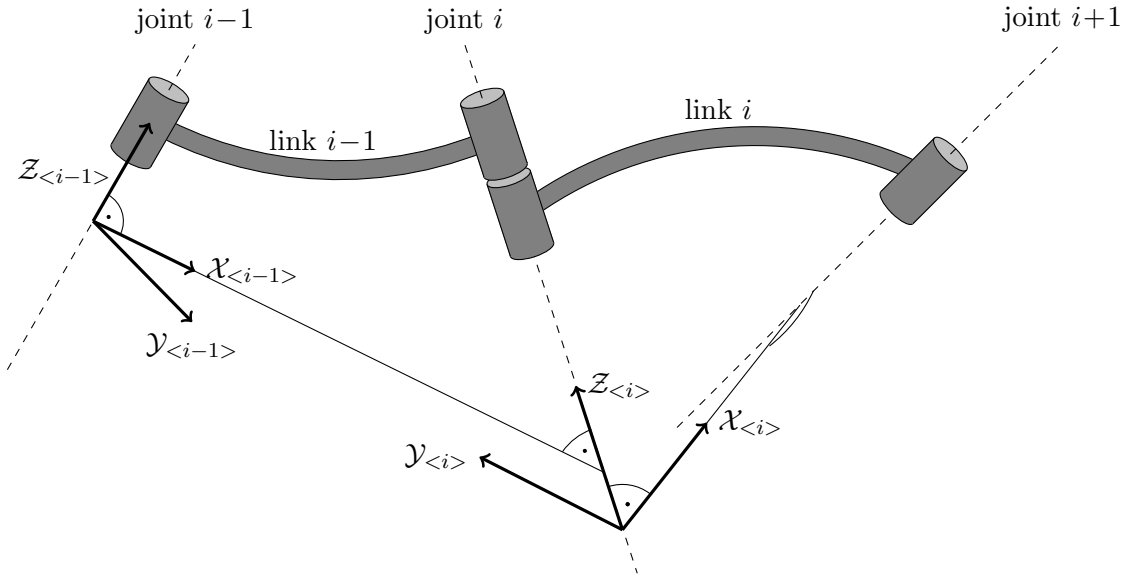


Figure 6.2: Frames of modified DH notation.

Consequently, the homogeneous transformation matrix from frame $i-1$ to frame i is given by

$$
\begin{aligned}
{}^{<i>}\mathcal{T}_{<i-1>} &:= \widehat{\mathcal{R}}(\mathcal{Z}_{<i>}, -\vartheta_i)\ \widehat{\mathcal{T}}(\mathcal{Z}_{<i>}, -\beta_i)\ \widehat{\mathcal{R}}(\mathcal{X}_{<i-1>}, -\rho_i)\ \widehat{\mathcal{T}}(\mathcal{X}_{<i-1>}, -l_i) \\
&= \begin{pmatrix} \cos\vartheta_i & \sin\vartheta_i\cos\rho_i & \sin\vartheta_i\sin\rho_i & -l_i\cos\vartheta_i \\ -\sin\vartheta_i & \cos\vartheta_i\cos\rho_i & \cos\vartheta_i\sin\rho_i & l_i\sin\vartheta_i \\ 0 & -\sin\rho_i & \cos\rho_i & -\beta_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&=: \begin{pmatrix} {}^{<i>}\mathcal{R}_{<i-1>} & {}^{<i>}\mathcal{P}_{<i-1>} \\ 0 & 1 \end{pmatrix}.
\end{aligned}
$$

We also state the inverse here, because it will be useful in the context of deriving the dynamic equations of the chain of rigid bodies:

$$
\begin{aligned}
{}^{<i-1>}\mathcal{T}_{<i>} &:= \widehat{\mathcal{T}}(\mathcal{X}_{<i-1>}, l_i)\ \widehat{\mathcal{R}}(\mathcal{X}_{<i-1>}, \rho_i)\ \widehat{\mathcal{T}}(\mathcal{Z}_{<i>}, \beta_i)\ \widehat{\mathcal{R}}(\mathcal{Z}_{<i>}, \vartheta_i) \\
&= \begin{pmatrix} \cos\vartheta_i & -\sin\vartheta_i & 0 & l_i \\ \sin\vartheta_i\cos\rho_i & \cos\vartheta_i\cos\rho_i & -\sin\rho_i & -\beta_i\sin\rho_i \\ \sin\vartheta_i\sin\rho_i & \cos\vartheta_i\sin\rho_i & \cos\rho_i & \beta_i\cos\rho_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&=: \begin{pmatrix} {}^{<i-1>}\mathcal{R}_{<i>} & {}^{<i-1>}\mathcal{P}_{<i>} \\ 0 & 1 \end{pmatrix}.
\end{aligned}
$$

Note that the inverse ${}^{<i-1>}\mathcal{T}_{<i>}$ using this modified convention has the same structure as the transformation matrix ${}^{<i>}\mathcal{T}_{<i-1>}$ of the original notation. In particular, ${}^{<i-1>}\mathcal{P}_{<i>}$ is independent of ϑ_i and thus a constant vector, because, if joints are rotational joints, the only variable DH parameters are ϑ_i, $i = 0, \ldots, N$. This observation is valuable for the following derivation of the dynamical equations by Newton-Euler recursion.

6.2.2 Equations of Motion

The dynamic equations of a chain of rigid bodies are a system of ordinary differential equations describing the motion caused by torques and forces acting upon the chain. Several methods to derive these equations of motion are presented in literature. They use the structure of the problem to different extends and differ in the computational costs. Here we state a basic approach, the Newton-Euler algorithm, which allows to break the dynamics down into the basic physical principles of Newtonian dynamics. The more advanced techniques are summarized at the end of this section.

6.2.2.1 Newton-Euler Recursion

In addition to the kinematic properties describing the shapes and relative positions of the objects (see previous section), each link i has characteristic dynamical properties like a mass $m_i \in \mathbb{R}$ and an inertia matrix $\mathcal{I}^{(i)} \in \mathbb{R}^{3\times 3}$; note that the inertia value is stated with respect to the corresponding frame i and the rotation axis of joint i. The Newtonian theory of dynamics shows that the distributed mass of an object can be idealized into a point mass without changing the dynamics of the object. Therefore, each link is assumed to have a center of mass (com) where this idealized mass point would be located; the positions are given with respect to the corresponding frame i by the constant vectors ${}^{<i>}o^{(i)} \in \mathbb{R}^3$. The forces and torques acting on the center of mass of link i are denoted by ${}^{<i>}F^{(i)}_{com} \in \mathbb{R}^3$ and ${}^{<i>}T^{(i)}_{com} \in \mathbb{R}^3$, respectively. Note that these two quantities are functions of the time t, but the argument is omitted in the following to shorten the notation; the same holds true for all quantities defined below. The vector $\left({}^{<i>}v^{(i)}_{com}\right)' \in \mathbb{R}^3$ states the (linear) acceleration of the center of mass of link i. Additionally, one has to define the corresponding quantities acting upon the joint i: force ${}^{<i>}F^{(i)}_{joint} \in \mathbb{R}^3$ and torque ${}^{<i>}T^{(i)}_{joint} \in \mathbb{R}^3$. Furthermore, the velocities and accelerations of the frame i with respect of to the base frame 0 have to be introduced to derive the dynamical equations: linear velocity ${}^{<i>}v^{(i)} \in \mathbb{R}^3$, linear acceleration $\left({}^{<i>}v^{(i)}\right)' \in \mathbb{R}^3$, angular velocity ${}^{<i>}\omega^{(i)} \in \mathbb{R}^3$ and angular acceleration $\left({}^{<i>}\omega^{(i)}\right)' \in \mathbb{R}^3$.

Having defined all appearing quantities, the Newton-Euler algorithm can be written in form of two recursions: first, an outward recursion from the base to the last link and after that an inward recursion from the end back to the base. The input to these recursions are the current joint angles ϑ_i and their time-derivatives ϑ_i' and ϑ_i'' for all i.

Outward Recursion: The following initial values for the outward recursion describe the state of the basis and have to be given: ${}^{<i>}v^{(i)}$, $\left({}^{<i>}v^{(i)}\right)'$, ${}^{<i>}\omega^{(i)}$ and $\left({}^{<i>}\omega^{(i)}\right)'$ for $i = 0$; in most cases all these values are equal to zero.

The recursion step from $i-1$ to i:

First, the equations relating the velocities and accelerations of frame i to those of frame $i-1$:

$$
\begin{aligned}
{}^{<i>}v^{(i)} &= {}^{<i>}\mathcal{R}_{<i-1>}\left({}^{<i-1>}v^{(i-1)} + {}^{<i-1>}\omega^{(i-1)} \times {}^{<i-1>}\mathcal{P}_{<i>}\right),\\
{}^{<i>}\omega^{(i)} &= {}^{<i>}\mathcal{R}_{<i-1>}\,{}^{<i-1>}\omega^{(i-1)} + \vartheta_i'\,{}^{<i>}e_z,\\
\left({}^{<i>}v^{(i)}\right)' &= {}^{<i>}\mathcal{R}_{<i-1>}\Big(\left({}^{<i-1>}v^{(i-1)}\right)' + \left({}^{<i-1>}\omega^{(i-1)}\right)' \times {}^{<i-1>}\mathcal{P}_{<i>}\\
&\qquad + {}^{<i-1>}\omega^{(i-1)} \times \left({}^{<i-1>}\omega^{(i-1)} \times {}^{<i-1>}\mathcal{P}_{<i>}\right)\Big),\\
\left({}^{<i>}\omega^{(i)}\right)' &= {}^{<i>}\mathcal{R}_{<i-1>}\left({}^{<i-1>}\omega^{(i-1)}\right)' + \left({}^{<i>}\mathcal{R}_{<i-1>}\,{}^{<i-1>}\omega^{(i-1)}\right) \times \left(\vartheta_i'\,{}^{<i>}e_z\right) + \vartheta_i''\,{}^{<i>}e_z,
\end{aligned}
$$

where ${}^{<i>}e_z \in \mathbb{R}^3$ is the unit vector in $\mathcal{Z}_{<i>}$-direction. Note that these four equations are the standard equations for two systems moving around each other at a constant distance. Consequently, the acceleration of the center of mass of link i is given by

$$\left({}^{<i>}v_{com}^{(i)}\right)' = \left({}^{<i>}v^{(i)}\right)' + \left({}^{<i>}\omega^{(i)}\right)' \times {}^{<i>}o^{(i)} + {}^{<i>}\omega^{(i)} \times \left({}^{<i>}\omega^{(i)} \times {}^{<i>}o^{(i)}\right).$$

This can be used to state the forces and torques for the center of mass by the standard equation of Newtonian dynamics based on mass and inertia of the link:

$$\begin{aligned} {}^{<i>}F_{com}^{(i)} &= m_i \left({}^{<i>}v_{com}^{(i)}\right)', \\ {}^{<i>}T_{com}^{(i)} &= \mathcal{I}^{(i)} \left({}^{<i>}\omega^{(i)}\right)' + {}^{<i>}\omega^{(i)} \times \left(\mathcal{I}^{(i)<i>}\omega^{(i)}\right). \end{aligned}$$

Inward Recursion: The following initial values for the inward recursion describe the forces and torques acting on the last link of the chain of rigid bodies and have to be given: ${}^{<i>}F_{joint}^{(i)}$ and ${}^{<i>}T_{joint}^{(i)}$ for $i = N + 1$; these quantities are non-zero if the last link is in contact with the environment.

The recursion step from $i + 1$ to i is given by the following two equations:

$$\begin{aligned} {}^{<i>}F_{joint}^{(i)} &= {}^{<i>}\mathcal{R}_{<i+1>}\,{}^{<i+1>}F_{joint}^{(i+1)} + {}^{<i>}F_{com}^{(i)}, \\ {}^{<i>}T_{joint}^{(i)} &= {}^{<i>}\mathcal{R}_{<i+1>}\,{}^{<i+1>}T_{joint}^{(i+1)} + {}^{<i>}T_{com}^{(i)} + {}^{<i>}o^{(i)} \times {}^{<i>}F_{com}^{(i)} \\ &\quad + {}^{<i>}\mathcal{P}_{<i+1>} \times {}^{<i>}F_{joint}^{(i)}, \end{aligned}$$

which assume that for forces linear superposition holds and that forces generate further torques depending on the moment arms. Note that the actual torque acting upon the joint i is the $\mathcal{Z}$-component of ${}^{<i>}T_{joint}^{(i)}$, since the other two components are absorbed by the rigid properties of the joints.

The here presented form of the Newton-Euler recursion assumes that only rotational joints connect the links, but a generalization to prismatic joints is straightforward. Furthermore, external forces and external torques can easily be included in the inward recursion (see for example [54, 55]).

6.2.2.2 State Space Equations

The Newton-Euler equations can be rewritten in a compact form, the state space equations:

$$T(t) = M(\vartheta(t))\vartheta''(t) + \Theta(\vartheta(t), \vartheta'(t)),$$

where $M(\vartheta(t)) \in \mathbb{R}^{N \times N}$ is called the mass matrix and combines all terms that are multiplied by $\vartheta''(t)$ in the course of the recursion. Note that the mass matrix is a function only of $\vartheta(t)$; it can be shown that the mass matrix is always symmetric positive semidefinite. Configurations where the mass matrix is not positive definite are called singular; consequently, such configurations have to be avoided in order to uniquely solve the equations of motions. All other terms caused by centrifugal forces, Coriolis forces and gravitation are collected in $\Theta(\vartheta'(t), \vartheta''(t)) \in \mathbb{R}^N$.

The dynamics given by the state space equation can be written in form of the first-order ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \vartheta(t) \\ \vartheta'(t) \end{pmatrix} = \begin{pmatrix} \vartheta'(t) \\ M(\vartheta(t))^{-1}\left(T(t) - \Theta(\vartheta(t), \vartheta'(t))\right) \end{pmatrix},$$

which has the form

$$\overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}(t)) \tag{6.1}$$

defining the control $\overline{u}(t) := T(t)$ and the state by

$$\overline{x}(t) := \begin{pmatrix} \vartheta(t) \\ \vartheta'(t) \end{pmatrix}.$$

Note that from the numerical perspective the inverse $M(\vartheta(t))^{-1}$ is never actually formed, rather the value $\vartheta''(t)$ of the second block row is the solution of the system of linear equations given by the state space equations.

In addition to the standard Newton-Euler recursion computing the joint torques for given joint angles and their time-derivatives, derivative information is needed for the optimization. One possibility to get this derivative information is to a analytically derive each equation of the recursions. To increase efficiency several quantities can be reused in the recursions of the derived Newton-Euler algorithm, therefore the combination is called the extended Newton-Euler recursion [55]. Other approaches to compute the derivatives are mentioned in the following section.

The extended Newton-Euler recursion computes for the inputs $\vartheta(t)$, $\vartheta'(t)$ and $\vartheta''(t)$ the values of $T(t)$, $\frac{\partial T}{\partial \vartheta_j}(t)$ and $\frac{\partial T}{\partial \vartheta'_j}(t)$ for $j = 1, \ldots, N$, consequently one can denote the joint torques as a function of the three input vectors: $T(t) = T(\vartheta(t), \vartheta'(t), \vartheta''(t))$. Using this notation the equations of the extended Newton-Euler recursion (including second derivatives) can be written as

$$\begin{aligned}
T(\vartheta(t), \vartheta'(t), \vartheta''(t)) &= M(\vartheta(t))\vartheta''(t) + \Theta(\vartheta(t), \vartheta'(t)), \\
\frac{\partial T}{\partial \vartheta_j}(\vartheta(t), \vartheta'(t), \vartheta''(t)) &= \frac{\partial M(\vartheta(t))}{\partial \vartheta_j}\vartheta''(t) + \frac{\partial \Theta(\vartheta(t), \vartheta'(t))}{\partial \vartheta_j}, \\
\frac{\partial T}{\partial \vartheta'_j}(\vartheta(t), \vartheta'(t), \vartheta''(t)) &= \frac{\partial \Theta(\vartheta(t), \vartheta'(t))}{\partial \vartheta'_j}, \\
\frac{\partial^2 T}{\partial \vartheta_j \partial \vartheta_k}(\vartheta(t), \vartheta'(t), \vartheta''(t)) &= \frac{\partial^2 M(\vartheta(t))}{\partial \vartheta_j \partial \vartheta_k}\vartheta''(t) + \frac{\partial^2 \Theta(\vartheta(t), \vartheta'(t))}{\partial \vartheta_j \partial \vartheta_k}, \\
\frac{\partial^2 T}{\partial \vartheta'_j \partial \vartheta_k}(\vartheta(t), \vartheta'(t), \vartheta''(t)) &= \frac{\partial \Theta(\vartheta(t), \vartheta'(t))}{\partial \vartheta'_j \partial \vartheta_k}, \\
\frac{\partial^2 T}{\partial \vartheta'_j \partial \vartheta'_k}(\vartheta(t), \vartheta'(t), \vartheta''(t)) &= \frac{\partial \Theta(\vartheta(t), \vartheta'(t))}{\partial \vartheta'_j \partial \vartheta'_k}, \\
& \qquad j = 1, \ldots, N, \quad k = 1, \ldots, N.
\end{aligned}$$

The structure of the ODE (6.1) shows that the values of $M(\vartheta(t))$, $\Theta(\vartheta(t), \vartheta'(t))$ and their derivatives are needed in explicit form. Thus a strategy is needed to extract these from the equations for $T(t)$, $\frac{\partial T}{\partial \vartheta_j}(t)$ and $\frac{\partial^2 T}{\partial \vartheta_j \partial \vartheta_k}(t)$. First, the summands corresponding to Θ are

singularized by setting the input values $\vartheta''(t)$ to zero:

$$\begin{aligned}
\Theta(\vartheta(t),\vartheta'(t)) &= T(\vartheta(t),\vartheta'(t),0),\\
\frac{\partial\Theta(\vartheta(t),\vartheta'(t))}{\partial\vartheta_j} &= \frac{\partial T}{\partial\vartheta_j}(\vartheta(t),\vartheta'(t),0),\\
\frac{\partial^2\Theta(\vartheta(t),\vartheta'(t))}{\partial\vartheta_j\partial\vartheta_k} &= \frac{\partial^2 T}{\partial\vartheta_j\partial\vartheta_k}(\vartheta(t),\vartheta'(t),0),\\
& \quad j=1,\ldots,N,\quad k=1,\ldots,N.
\end{aligned}$$

The other parts corresponding to M can than be obtained by changing the input value $\vartheta''(t)$ to the i-th unit vector $e^{(i)} \in \mathbb{R}^N$ for $i = 1,\ldots,N$:

$$\begin{aligned}
M(\vartheta(t))e^{(i)} &= T(\vartheta(t),\vartheta'(t),e^{(i)}) - T(\vartheta(t),\vartheta'(t),0),\\
\frac{\partial M(\vartheta(t))}{\partial\vartheta_j}e^{(i)} &= \frac{\partial T}{\partial\vartheta_j}(\vartheta(t),\vartheta'(t),e^{(i)}) - \frac{\partial T}{\partial\vartheta_j}(\vartheta(t),\vartheta'(t),0),\\
\frac{\partial^2 M(\vartheta(t))}{\partial\vartheta_j\partial\vartheta_k}e^{(i)} &= \frac{\partial^2 T}{\partial\vartheta_j\partial\vartheta_k}(\vartheta(t),\vartheta'(t),e^{(i)}) - \frac{\partial^2 T}{\partial\vartheta_j\partial\vartheta_k}(\vartheta(t),\vartheta'(t),0),\\
& \quad j=1,\ldots,N,\quad k=1,\ldots,N.
\end{aligned}$$

which means that the mass matrix and its derivatives are assembled columnwise by N additional runs of the extended Newton-Euler recursion. Note that this strategy does not consider the symmetry of the mass matrix. For more details on the extended Newton-Euler recursion refer to [55, 335].

Having $M(t)$, $\Theta(t)$ and their derivatives, a last point to address is the relation between derivatives of the mass matrix and derivatives of the inverse of the mass matrix. Starting point is the identity

$$\mathcal{U} = (M(\vartheta(t)))^{-1}M(\vartheta(t)),$$

where $\mathcal{U} \in \mathbb{R}^{N\times N}$ is the identity matrix. The first derivative with respect to ϑ_j, $j = 1,\ldots,N$, yields:

$$\frac{\partial(M(\vartheta(t)))^{-1}}{\partial\vartheta_j} = -(M(\vartheta(t)))^{-1}\frac{\partial M(\vartheta(t))}{\partial\vartheta_j}(M(\vartheta(t)))^{-1}.$$

In consequence the second derivative reads:

$$\begin{aligned}
\frac{\partial^2(M(\vartheta(t)))^{-1}}{\partial\vartheta_j\partial\vartheta_k} &= -(M(\vartheta(t)))^{-1}\frac{\partial^2 M(\vartheta(t))}{\partial\vartheta_j\partial\vartheta_k}(M(\vartheta(t)))^{-1}\\
&+ (M(\vartheta(t)))^{-1}\frac{\partial M(\vartheta(t))}{\partial\vartheta_k}(M(\vartheta(t)))^{-1}\frac{\partial M(\vartheta(t))}{\partial\vartheta_j}(M(\vartheta(t)))^{-1}\\
&+ (M(\vartheta(t)))^{-1}\frac{\partial M(\vartheta(t))}{\partial\vartheta_j}(M(\vartheta(t)))^{-1}\frac{\partial M(\vartheta(t))}{\partial\vartheta_k}(M(\vartheta(t)))^{-1},\\
& j=1,\ldots,N,\quad k=1,\ldots,N.
\end{aligned}$$

Summing up, multiple runs of the extended Newton-Euler algorithm can be used to compute all quantities needed to state the ODEs of the rigid body dynamics and their derivatives [55, 335].

6.2.2.3 Methods of Forward and Inverse Dynamics

The problem the Newton-Euler algorithm in its basic form addresses is the computation of the torques for given angles, velocities and accelerations of the joints. This problem is commonly referred to as the *problem of inverse dynamics*. Two efficient approaches to solve this problem are the Lagrangian method and the Newton-Euler recursion [17], which are from the theoretical perspective equivalent [282], but they differ in computational complexity. The basis of the Lagrangian method are energy considerations, whereupon the Newton-Euler algorithm is based on the equilibrium of torques and forces [65].

On the other hand, if the torques, angles and velocities of the joints are known, the problem of determining $\vartheta''(t)$ by the equations of motions, the so-called *problem of forward dynamics*, has to be solved. According to [167] three main solution strategies exist for this problem: The strategy of the first group is to solve the equations of motions via decomposition of the explicitly computed mass matrix. The second group of algorithms generate a modified version of the dynamic equations to simplify solving the linear system. Solution strategies that tackle the problem in a direct manner without using the structure given by the equations of motion form the third group.

As presented in the previous section, the Newton-Euler recursion can be used to compute the mass matrix and therefore might be used to solve the problem of forward dynamics by decomposition of this mass matrix, a solution strategy from group one. A more efficient version of the Newton-Euler algorithm can be realized by taking into account that in each recursion step i the links 1 to $i-1$ are at a static equilibrium and only the links i to N are accelerated. Consequently, one can reduce the recursion to a rigid body chain of shorter length and make use of the symmetry of the mass matrix. Another solution strategy representing group one is the Composite Rigid Body method [91, 335]. The basic idea of this method is to view the links accelerated in a recursion step of the Newton-Euler algorithm as one composite rigid body. The Composite Rigid Body method is an efficient algorithm which uses the symmetry of the mass matrix and allows for pre-computation of several quantities.

Algorithms which generate a decomposition version of the equations of motions, i.e., solutions strategies of group two, are for example presented by [167, 264]. The method of [264] yields a triangularization of the system using a suitable algorithmic form of the mass matrix given by Kane's method, whereas the method of [167] uses a special decomposition of the mass matrix. The most important algorithm of the third group of solution strategies for the problem of forward dynamics is the Articulated Body method [91, 325]. The idea is the reduction of the problem of N rigid bodies on N simpler problems with one joint only. This reduction is achieved by considering groups of links and using three recursions to compute the joint accelerations directly.

The derivative information required by both the direct and indirect methods to optimally control motions of a chain of rigid bodies can be obtained by various strategies like numerical, symbolical or automatic differentiation. The strategy of differentiating the recursive algorithms for the equations of motion has been pursued by a few research groups (e.g. [168, 286]). The presentation of the extended Newton-Euler recursion in the previous section is based on publications of the group of Callies [54, 55] and extension of the differentiation technique to the Composite Rigid Body method generates a very efficient technique to compute the derivative information [248]. For more details on the algorithms of the problem of forward dynamics and the derivative computation, especially for comparisons of numerical complexities see [92, 248].

6.3 Muscle Models

A large research area is the biomechanical study of the human muscles which covers the whole range from individual motor units to the total muscle itself [338]. A motor unit is the smallest part of the muscle that can be controlled individually and the number of included muscle fibers differs considerable from less than ten to more than thousand for differently precise muscles. The individual muscle fibers themselves are a combination of different microstructures including the contractile element which generates the tension and the passive connective tissue which encloses the contractile elements and connects it to the tendons at either end. To analyze the behavior of the total muscle the mechanical characteristics of both, the active and the passive elements, have to be modeled [338].

A central characteristic of human muscles is the force-length relation describing which force can be generated at a given muscle length. The general relation can be split into three subcharacteristics: The force-length curve of the contractile elements, the nonlinear behavior of parallel elastic component, which is the connective tissue enclosing the contractile elements, and the characteristics of the series elastic element, which combines the tissue that connects to the tendons and the tendons themselves. In addition to the static force-length relation one has to account for the dynamical changes which result in the force-velocity relation. A first curve fit using an exponential function is presented by [94], followed by the work of Hill [154] where a hyperbolic form is used and the result is put into context with the internal thermodynamics. For more details on muscle characteristics see [229, 338].

Various muscle models have been proposed (e.g. [42, 149, 340]); they range from simple mass-damper systems over models including several nonlinear properties of the muscles [339] to rather complex models which try to describe every miniature effect [148, 355]. In the following, the two examples of muscle models used later in the computations are introduced. First, a second-order ODE is presented which captures that the muscles' characteristics are similar to those of a low-pass filter. Second, the nonlinear model of Stroeve is discussed which includes the force-length and the force-velocity relation. From the mathematical perspective we assume that the muscle dynamics can be written in form of a first-order ODE

$$\overline{x}'(t) = \varphi(\overline{x}(t), \overline{u}(t)),$$

where $\overline{x}(t)$ is the state and $\overline{u}(t)$ the control. The output of the model is the force $F(t)$ generated by the muscle:

$$F(t) = \mathcal{M}(\overline{x}(t), \overline{u}(t)).$$

Consequently, other muscle models fitting into this general structure can be used if needed.

6.3.1 A Linear Muscle Model

The following model taken from Winter [338] is a mass-damper system that is critically damped. Due to similarities to waveform characteristics of muscle twitches and the responses of the mass-damper system to impulses, the basic idea is to use the characteristic twitching time $\overline{\tau}$ of a muscle to determine the coefficient of the second-order ODE:

$$\overline{\tau}^2 F''(t) + 2\overline{\tau} F'(t) + F(t) = \overline{u}(t),$$

where $F(t)$ denotes the generated muscle force and $\overline{u}(t)$ the scalar control. This differential equation can be rewritten in the form of a linear first-order ODE:

$$\overline{x}'(t) = \begin{pmatrix} 0 & 1 \\ -\overline{\tau}^{-2} & -2\overline{\tau}^{-1} \end{pmatrix} \overline{x}(t) + \begin{pmatrix} 0 \\ \overline{\tau}^{-2} \end{pmatrix} \overline{u}(t)$$

with the state $\overline{x}(t)$ defined by

$$\overline{x}(t) := \begin{pmatrix} F(t) \\ F'(t) \end{pmatrix}.$$

According to [338] twitching times vary between different human muscles. While fast-twitch motor units are reported to have twitching times of about 10 – 50 [ms], the slow twitches of other motor units lie in a range of 60 – 100 [ms]. In consequence, the twitching times for an overall muscle vary depending on which type of motor units is used to which amount and also the mean twitching time differs between various human muscles. Here (in accordance to [303]) we use $\overline{\tau} = 0.04$ seconds for all muscles, which is a simplification, but actual measurements of the human doing the considered movements would be needed to reduce this modeling error.

6.3.2 Muscle Model of Stroeve

Based on the classical work of Hill [154] for an isolated muscle, Winters and Stark [339] propose a nonlinear model for single joint motions which includes the four most basic properties of human muscles: a torque-velocity relation for shortening and for lengthening the muscle, the series and parallel viscoelastic properties and the static moment-angle relation. It is shown that the model is consistent with previous publications on muscle characteristics and that it can reproduce several sets of available data, while having a clear structure of the processes involved in the dynamics.

For the models used in this work a slightly simplified model, the muscle model of Stroeve [291, 292] is used. It simplifies the model of [339] with respect to combined excitation and activation dynamics and a infinitely stiff series-elastic element. The modifications are introduced to ease the numerical simulation of the muscle model, but the main characteristics of human muscles are still reproducible. The focus of the publications of Stroeve is on impedance characteristics of human muscles considering both static postures and movements of a human arm. The motor control system is modeled as a nonlinear system including feedback and the actual neural input to this control system is given by a neural network; a learning process is applied to train the network for human arm motions. Results show that the presented approach can explain impedance in human arm motions and that the muscle model does properly represent the intrinsic characteristics [291, 292]. Therefore, this model seems to be a reasonable trade-off between model complexity and model accuracy and several details of the model are presented in the following.

The input values for the muscle model of Stroeve are the muscle length $l(t)$, the muscle velocity $l'(t)$ and the neural input $\upsilon(t)$. The actual state of the model is the activation $a(t)$ and its time-derivative is given by a simple first-order ODE using the neural input $\upsilon(t)$ and activation $a(t)$:

$$a'(t) = \frac{\upsilon(t) - a(t)}{\widehat{t}(t)},$$

where $\widehat{t}(t)$ is the current time-constant. This constant switches between the activation time constant t_{ac} and the de-activation time constant t_{da} in dependence on the relation of the

neural input $\upsilon(t)$ and the activation $a(t)$:

$$\widehat{t}(t) = \begin{cases} t_{ac} & \text{for} \quad \upsilon(t) \geq a(t) \\ t_{da} & \text{for} \quad \upsilon(t) < a(t) \end{cases}$$

If necessary from the optimization perspective, one could easily introduce a smoothed version of this switching structure.

The output of this model is the muscle force $F(t)$, which is returned by function $\mathcal{M}$:

$$F(t) = \mathcal{M}\left(a(t), l(t), l'(t)\right)$$

This function makes use of the length of the contractile element $l_c(t)$, which is the difference between muscle length $l(t)$ and the (constant) length of the tendon l_t:

$$l_c(t) = l(t) - l_t.$$

And consequently for the time-derivatives the following identity holds:

$$l'_c(t) = l'(t).$$

The output function $\mathcal{M}$ is the product of the maximal muscle force F_{max}, the current activation $a(t)$, the force-length relation $F_l\left(l_c(t)\right)$ and the force-velocity relation $F_v(l'_c(t))$:

$$\mathcal{M}\left(a(t), l(t), l'(t)\right) = a(t) F_l\left(l_c(t)\right) F_v(l'_c(t)) F_{max}.$$

The force-length relation F_l depending on the length of the contractile element $l_c(t)$ is modeled by a Gaussian curve whose width and center are determined by the constants l_{csh} and l_{co}, respectively.

$$F_l(l_c(t)) = \exp\left(-\left(\frac{l_c(t) - l_{co}}{l_{csh}}\right)^2\right).$$

The constant l_{co} for the center of the Gaussian curve is given by the following relation of the constant for the minimal muscle length l_{min}, the maximal muscle length l_{max} and the length of the tendon l_t:

$$l_{co} = l_{min} + l_{opt}(l_{max} - l_{min}) - l_t,$$

where optimal muscle length ratio l_{opt} is the given constant of the model. The width of the Gaussian curve follows by

$$l_{csh} = l_{sh}(l_{max} - l_{min}),$$

where l_{sh} is another constant stating the relative width. The definition of the force-velocity relation $F_v(t)$ is divided into three cases depending on the maximum contraction velocity $v_{max}(t)$.

First, if $l'_c(t) \leq -v_{max}(a(t), l_c(t))$ the value of the force-velocity relation is zero:

$$F_v\left(l'_c(t)\right) = 0.$$

Second, for $-v_{max}(a(t), l_c(t)) \leq l'_c(t) < 0$ the value of the relation is given by:

$$F_v\left(l'_c(t)\right) = \frac{V_{sh}\left(v_{max}(a(t), l_c(t)) + l'_c(t)\right)}{V_{sh} v_{max}(a(t), l_c(t)) - l'_c(t)}.$$

Third, in case of $l_c'(t) \geq 0$ the following value is assigned:

$$F_v\left(l_c'(t)\right) = \frac{V_{sh}V_{shl}v_{max}(a(t), l_c(t)) + V_{ml}l_c'(t)}{V_{sh}V_{shl}v_{max}(a(t), l_c(t)) + l_c'(t)}.$$

This model of the Hill curve uses the constants V_{sh} and V_{shl} to determine the concavity of the curve during shortening and lengthening, accordingly. Furthermore, the maximum velocity during concentric contraction is given by the constant V_{er}. Finally, the maximum contraction velocity $v_{max}(t)$ reads:

$$v_{max}(a(t), l_c(t)) = V_{vm}\left(1 - V_{er}\left(1 - a(t)F_l(l_c(t))\right)\right),$$

where the constants V_{vm} and V_{er} are used to model the characteristic muscle velocity and the effect of the activation on the maximum velocity, respectively.

Note that all functions used to model the muscle dynamics except the characteristic time-constant $\widehat{t}(t)$ are at least two-times continuously differentiable, which is necessary for smooth optimization techniques using the Hessian of problem.

A part that still needs to be discussed is the computation of the current muscle length $l(t)$ and its time-derivative $l'(t)$, the muscle velocity. Both quantities are input values of the actual muscle model of Stroeve, since they depend on the arm configuration. Considering the bones as rigid objects, the notation of section 6.2 is used here to model the muscle lengths in dependence on the arm positions.

The basic assumption is that the muscle has a constant moment arm $R_j \in \mathbb{R}$ for each link $j = 1, \ldots, N$. Human experiments show that the moment arm is in most cases not constant, but the choice of a more realistic model for the moment arms is out of the scope of this work. Note that the here presented framework allows to include better models without changing the general problem structure.

Furthermore, a resting angle $\vartheta_{r,j}$ for each joint $j = 0, \ldots, N-1$ is introduced which is defined by the arm configuration assuring that the muscle is at its resting length l_r. The current muscle length is then approximated by the following equation:

$$l(t) = l_r - \sum_{j=0}^{N-1} R_j\left(\vartheta_j(t) - \vartheta_{r,j}\right).$$

In consequence, the time-derivative reads:

$$l'(t) = -\sum_{j=0}^{N-1} R_j\vartheta_j'(t).$$

Using these equations the muscle model of Stroeve depending so far on the input υ, l and l' can be extended to use υ, ϑ and ϑ' as inputs, which later on will allow to combine the muscle dynamics with the rigid body dynamics:

$$F(t) = \underline{\mathcal{M}}\left(\vartheta(t), \vartheta'(t), a(t)\right).$$

Defining the extended state $\overline{x}(t)$ and the extended control by

$$\overline{x}(t) := \begin{pmatrix} l(t) \\ l'(t) \\ a(t) \end{pmatrix} \quad \text{and} \quad \overline{u} := \begin{pmatrix} \vartheta(t) \\ \vartheta'(t) \\ \vartheta''(t) \\ \upsilon(t) \end{pmatrix},$$

the differential equation of the state reads:

$$\overline{x}'(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\widehat{t}(t)^{-1} \end{pmatrix} \overline{x}(t) + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -R^T & 0 \\ 0 & 0 & 0 & \widehat{t}(t)^{-1} \end{pmatrix} \overline{u}(t).$$

Note that $\vartheta(t)$ and $\vartheta'(t)$ are only added to the control vector to unify the notation, i.e., the force output is given by a function of elements of the state $\overline{x}(t)$ and the control $\overline{u}(t)$.

6.4 Cost Functions

In this section details on cost functions of human arm motions are discussed with a focus on the open-loop optimization criteria used later on in the application examples. Naturally, the first criteria presented in literature describe single joint motions; they are followed by those for planar motions and, finally, for three-dimensional movements. The complexities of the considered arm models differ considerably. To find the right combination of model and optimization framework is a non-trivial task.

Having decided on an optimization framework and a model of the plant, the next question is which cost function describes human behavior for a given task best and it can only be answered if the results of the various costs are compared to recorded human motions [244]. In this context it has to be noted that a cost yielding results corresponding to observed human motions in one setup might not be suitable to describe human arm movements in another setup with more degrees of freedom [3, 152, 158, 174]. The consequence drawn by [152] is that future tests of cost hypotheses should include several models and analyze not only planar, but rather three-dimensional movements in such a comparison. Due to the fact that some simple cost functions describe human arm movements better in one task than in another, Todorov [302] noted that "the true performance criterion in most cases is likely to involve a mix of cost terms."

6.4.1 Smoothness

The development of smoothness-related cost functions is closely connected to progression of known characteristics in human arm motions (cf. section B.2). A common principle assuring the smoothness of the solutions is that higher-order time derivatives of joint angles or the hand position are minimized. The discussion of such cost functions starts with criteria based solely on the kinematics of the arm and then dynamics-based cost function are introduced. Naturally, each of the cost function comes with a coordinate system to represent the task in, which can cause discussions and originate experimental studies (cf. section 6.1). The authors of [223] summarize the discussion of opposing planning spaces as follows: "This controversial problem will continue, because the results depend on the setting of delicate task conditions, the number of learning trails, and the instructions given in the transformation experiments."

6.4.1.1 Minimum Jerk

The first reported characteristics of planar human arm motions are the generally straight hand paths [104, 218] and the bell-shaped and approximately symmetric tangential velocity profiles [2, 218]. As a result of these observations Flash and Hogan [104, 157] propose the

minimum jerk theory. Being based purely on kinematics, the criterion postulates that the trajectory of the human hand can be determined by minimizing jerk, which is the third time-derivative of the hand position.

$$f_{HJ} := \int_{t_0}^{t_f} \left|\left| \frac{\mathrm{d}^3}{\mathrm{d}t^3} \mathcal{P}_{hand}(t) \right|\right|^2 \mathrm{d}t,$$

where $\mathcal{P}_{hand}(t)$ is the position of the hand at time instance $t \in [t_0, t_f]$.

The solution of the minimum jerk problem is given by a polynomial of fifth grade whose coefficients are determined by the boundary conditions (cf. section 3.3). Properties of the minimum jerk trajectories are straight-line hand paths for point-to-point motions, a linear amplitude scaling, an endpoint translation invariance, a movement duration invariance and a velocity profile symmetry [89]. The simplicity and the excellent agreement with the so far known experimental observations makes this criterion "one of the most influential motor control theories" [89]. Starting from elbow motions of monkeys, the minimum jerk cost is initially used to describe horizontal, planar two-link movements. The error between the model's predictions and the observed data is reported to be up to experimental error, but some motor variability is noted. Perception and motor noise are named as possible sources for this variability. Already, the authors of [104] themselves question whether the minimum jerk principle capturing several characteristics of human motions might be the single criterion underlying all human arm movements.

In several later experiments it is noted that trajectories starting or ending near the workspace boundaries are noticeably and systematically curved [16, 98, 314]. Such curved trajectories could be generated by the minimum jerk criterion if via-points are introduced. A further experimental observation inconsistent with the minimum jerk theory is that the tangential velocity profiles of the hand are not perfectly symmetric, but tend to be right-skewed for slow movements and left-skewed for fast movements [89]. Since these observations cannot be explained by theory of minimizing hand jerk, further cost functions are presented in literature.

6.4.1.2 Minimum Joint Jerk

A second kinematic cost function that uses the joint coordinate frame is minimizing the jerk of the joint angles instead of the jerk of the hand position.

$$f_{JJ} := \int_{t_0}^{t_f} \left|\left| \frac{\mathrm{d}^3}{\mathrm{d}t^3} \vartheta(t) \right|\right|^2 \mathrm{d}t,$$

where $\vartheta(t)$ are the joint angles at time instance $t \in [t_0, t_f]$.

This criterion discussed in [262] yields always straight trajectories in joint space, which naturally results in curved hand paths, but they are considered to be too curved in literature [239]. However, the authors of [98, 223] argue that the predictions are quantitatively better than those of the minimum jerk model. Subsequently, approaches are introduced that use a compromise between jerk on the hand and on the joint level [66, 153, 233]. Such combinations are reported to avoid singularities and joint limits [153], but neuro-physiological evidence supporting this idea have not been presented [247].

6.4.1.3 Minimum Torque Change

Due to the fact that the minimum jerk criterion, being based solely on the kinematic properties of the arm, cannot explain the observed details of the curvature of the hand paths and the skewness of the velocity profile, a straight-forward step is to analyze whether dynamic properties of the human arm can explain these observations. Such a dynamic cost function guaranteeing smooth motions is proposed by [223, 314]; it minimizes the time-derivative of joint torques.

$$f_{TC} := \int_{t_0}^{t_f} \left|\left| \frac{\mathrm{d}}{\mathrm{d}t} T(t) \right|\right|^2 \mathrm{d}t,$$

where $T(t)$ are the joint torques at time instance $t \in [t_0, t_f]$.

The resulting hand trajectories are reported to be roughly straight and slightly curved in agreement with the experimental data [314]. The presented model of [314] is criticized due to the fact that the used rotatory inertia value is too big. If a correct combination of viscosity and inertia values is utilized, the trajectories are no longer similar to the observed trajectories [223].

6.4.1.4 Minimum Commanded Torque Change

Several refined versions of the idea of minimizing torque change exist; the first to mention is restricting the minimization to commanded torques [223].

$$f_{CTC} := \int_{t_0}^{t_f} \left|\left| \frac{\mathrm{d}}{\mathrm{d}t} T_{com}(t) \right|\right|^2 \mathrm{d}t,$$

where $T_{com}(t)$ are the commanded joint torques at time instance $t \in [t_0, t_f]$.

The minimum torque change cost function [314] uses only a dynamical model of the links, but neglects other dynamical characteristics of the human arm. In consequence, this related cost function is obtained if additionally the dynamics of the human muscles are taken into account to differentiate between actively generated torques and those resulting from the current state of the arm. For movements in the horizontal and sagittal plane it is reported that the magnitudes and directions of curvatures are better reproduced by this cost function than by the minimum hand jerk criterion or the minimum joint jerk criterion [223]. Comparing several smoothness-related cost functions for planar motions, it is noted in [334] that the criterion using commanded torque change comes closest to the data of two-dimensional experiments. For three-dimensional movements it is noted in [29] that the predicted hand paths and the observed ones consistently deviate from each other. Additionally, the predicted speed profiles show too small peak amplitudes and double peaks, in disagreement with the observed data.

6.4.1.5 Minimum Muscle Tension Change

A second variant of the minimum torque change idea is to minimize the change of muscle tensions, since muscles generate the torques relevant for the arm motions [315].

$$f_{MTC} := \int_{t_0}^{t_f} \left|\left| \frac{\mathrm{d}}{\mathrm{d}t} F(t) \right|\right|^2 \mathrm{d}t,$$

where $F(t)$ are the muscle tensions (or forces) at time instance $t \in [t_0, t_f]$. A consequence of modeling details of muscles is that a larger amount of model parameters is needed. The authors of [3] discuss that choosing the values for these parameters in combination with the variability in anatomy between participants might cause large variability in the results which complicates the quantitative comparison with experimental data.

6.4.1.6 Minimum Motor Command Change

One final step further, motor commands of the muscles are directly utilized instead of the tension generated by them.

$$f_{MCC} := \int_{t_0}^{t_f} \left|\left| \frac{\mathrm{d}}{\mathrm{d}t} \overline{u}(t) \right|\right|^2 \, \mathrm{d}t,$$

where $\overline{u}(t)$ are the motor commands (or activations) used to control the motions by the muscles at time instance $t \in [t_0, t_f]$.

This minimum motor command change model is proposed by [175] and the main problem using this hypothesis is to obtain a reasonable model for motor commands at the muscle level. In [182] an attempt to model the characteristics is made, but it too proves to be an extremely difficult process. Consequently, the authors themselves note in [223], that "a quantitative model, not a conceptual model, is needed to actually compute an optimal trajectory." For details on muscle models being the crucial part of this cost function see section 6.3.

6.4.2 Accuracy

Recording human arm motions between two given points, one easily notices that each subject has a characteristic variation of selected trajectories. The following statement of Todorov [302] describes the resulting problem for the so far discussed criteria: "One can be perfectly accurate on average and yet make substantial variable errors on individual trials."

A criterion describing human arm motions based on the observed variation is proposed by [141] within the open-loop framework. The hypothesis of the minimum variance theory is that humans try to minimize the variance of the hand at the final position. Naturally the following question results: how does the variance at the end depend on the choice of the arm motion to the final point? Harris and Wolpert base their idea on the characteristics of human muscles, due to the fact that human motor noise is known to be control-dependent [52, 170, 318, 316], i.e., the observed noise depends linearly on used controls. Consequently, the choice of the controls at each time instance influence the endpoint variance. The minimum variance hypothesis is that humans choose the trajectory that minimizes this variance. In the initial paper [141] saccadic eye movements and goal-directed arm motions are studied and the simulation results seem to explain the observed trajectories. Furthermore, it is noted that the speed-accuracy trade-off predicted by Fitts' law can also be explained by this framework. The idea of minimum variance at the end position is extended in [138] to tasks with obstacles; the additional constraint is to keep the collision probability below a fixed limit. It is reported that the optimal paths accurately predict the empirical trajectories. The minimum variance approach in the context of estimation and learning is discussed in [343] from the neuroscience perspective. A realization of the minimum variance approach in robotics can be found in [283, 284].

This minimum variance hypothesis is loosely related to the smoothness-related criteria discussed before, since large control inputs are needed to generate non-smooth hand paths which in consequence increase the signal-dependent noise. Furthermore, minimizing endpoint variance is related to the minimization of the sum of squared motor commands, which is a term associated with effort [303]. Consequently, the minimum variability criterion and the effort cost function often share a common minimum as [240] discuss. For a simple push-button experiment differences are observed and a combination of the two cost functions seems to be optimal to describe the data. Utilizing computer simulations the conclusion of [269] is that noise cannot be the only reason causing the variability of human reaching movements. The hypothesis is introduced that the variability of movements is caused by the errors in the perception of the final position. Psychophysical experiments of [269] show similarities in endpoint variance and variance of target perception. Contrarily, [318] show that noise in the movement execution rather than sensory or planning noise explains the variability of the final hand position. Similar to the discussion of the utilized planning spaces for the smoothness-related costs, diverse statements contradict each other, which could be a result of different experimental setups.

In addition to the question whether motor noise is big enough to cause the endpoint variance observed, the model used by [141] describing the signal to noise characteristics is questioned. In [238] the problem of co-contraction within the signal dependent noise framework is discussed. Co-contraction, which is the activation of two opposing muscles at an equal level, increases the impedance of the limb. Consequently, according to the signal-dependent noise framework an increase in co-contraction should cause more noise and add variation, but actually it reduces end point variation. The discussion of accuracy is closely related to closed-loop optimal control (cf. section B.5) and adaptation (cf. section B.6).

6.4.3 Energy, Time and Others

The subjective feeling of discomfort corresponding to the given arm configurations is analyzed in [67] from a psychophysical perspective. A two-dimensional arm model is utilized in this study and U-shaped cost functions depending on the joint angle are obtained. Several other studies come to similar results where humans prefer the comfort of one arm configuration compared to others. As [89] point out, "a problem with these studies is that the concept of discomfort is not well defined."

The idea of minimum muscle effort is proposed in [231] and a quantitative formula to measure effort is presented. The direct relation between muscular activity and effort is noted. The authors hypothesize that the minimum effort idea could be a central aspect in posture control and motion generation. Energy minimization is mostly used in human full-body posture and locomotion where the muscle activity over the gait circle is analyzed, e.g., [60, 8, 183, 213]. The main problem of the minimum energy hypothesis is that it needs a precise muscle model, because the metabolic energy consumption has to reflect the details of muscle physiology [302]. Such models cannot easily be developed and verified due to the difficulties with directly measuring the in-vivo loads of the muscles [63]; for further details on muscle models see section 6.3.

Starting with single joint arm motions, an effort-related criterion, the product of stiffness and muscle changes, is proposed by [146]. The results compared to experimental data seem acceptable. Planar arm movements based on metabolic costs are studied by [9]. The angular velocity is approximated by second-order Fourier series and motions are reported to agree well

with observed trajectories for fast movements when bi-articular muscles are considered. The minimum approach based on muscle tension change is extended by [171, 172] to the metabolic level, reflecting the physiological characteristics of the human arm. A numerical comparison to minimum jerk and minimum torque change is presented in [171] and shows quantitatively, that the proposed cost might be a good choice. The developed model relating neural input and isometric force [172] is, however, not compared to measured muscular activities. In [319] recorded muscle activation data is compared to the predictions of various cost functions related to muscle quantities and it is observed that no single cost function explaining all the effects in the data can be found, but criteria using squared muscle quantities seem to perform better. Although energy minimization alone does not reproduce the behavior observed in human arm movements [225], the idea that somehow energetics have to be involved is still up to date [302].

Another cost function we want to mention is the minimization of overall motion time t_f.

$$f_T := t_f.$$

If only this strategy is used, a control of bang-bang type results and due to properties of the human arm contradicting this control strategy this cost is considered implausible [225]. But note that for most optimization criteria the duration of the movement has to be prescribed and is that way an input to the model rather than an output. In consequence, minimization of motion time might be one of several factors describing in combination the characteristics of human movements; e.g., the experiments of [209] show that the overall energy cost seems to influence the motion time. Other task-dependent motion time characteristics are captured by Fitts' law (cf. section B.2.1). But still a description is missing of how the duration depends upon the circumstances of the movement [155]. In addition to the systematical problem of determining the movement time, the neural representation of time is an open field of research. The cerebellum seems to be involved, but distinctive models are still missing; a review can be found in [163].

6.4.4 Cost Combinations

A large variety of tasks exist where human use their arm, where the motions of one person normally show certain characteristic properties they might differ considerably from motions of other persons. The causes for these deviations can vary from physical properties of the persons to cultural and social norms [176]. In consequence one can hypothesize that humans might be aware of more than one cost function and that the relative weighting can differ from person to person. Since different cost functions are able to explain different aspects of motor behavior, this could indicate that the weighting factors of the considered costs are adapted for diverse tasks [23].

Human arm motions while interacting with a crank are analyzed in [201, 232], in which no muscle dynamics are modeled but rather muscle forces are the input to the system. It is shown that the minimum muscle force change criterion alone cannot reproduce movement or interaction forces. On the other hand, the minimum hand force change criterion, a task specific cost, can reproduce the movement, but not the observed forces at the crank. Only the combination of the two criteria agrees with the experiments.

A special combination of standard cost functions is presented by [233]. Considering movements in the horizontal plane it is proposed that a weighted combination of the minimum jerk trajectory and the minimum torque change model describes human motions much better than the single criterion alone. Here, again, one has to note that "expanding the workspace to 3D space might lead to a different weighting and maybe to other constraints and more optimization parameters" [3].

6.4.4.1 Convex Combination

A standard and simple way to combine given basic cost function is to build convex combinations. The advantage of convex combinations over linear combinations is to avoid ambiguity, because each scalar multiple of a cost function has the same optimum as the unscaled version. Naturally, other approaches to guarantee a unique representation of a combination of cost functions exist, for example, the weighting factor of a given basic cost function could be normalized [217].

In this work we assume that each basic cost function f_i is a function using the current state $\overline{x}(t)$ and the current control $\overline{u}(t)$ at time instance t with

$$f_i(\overline{x}(t), \overline{u}(t) \mid \pi_i) \in \mathbb{R}, \quad \forall i = 1, \ldots, k,$$

where $k \in \mathbb{N}$ is the number of considered basic cost functions and $\pi_i \in \mathbb{R}^{\overline{s}_i}$, $\overline{s}_i \in \mathbb{N}$, is a vector of parameters for the respective basic cost function. Using the weights $\overline{w}_i \in [0, 1]$, $i = 1, \ldots, k$, the convex combination reads:

$$f(\overline{x}(t), \overline{u}(t) \mid \pi) := \sum_{i=1}^{k} \overline{w}_i f_i(\overline{x}(t), \overline{u}(t) \mid \pi_i), \quad \text{with} \quad \sum_{i=1}^{k} \overline{w}_i = 1,$$

where the parameter vector π is the concatenation of the weighting factors and the parameter vectors of the basic cost functions. For later use we introduce the weight distribution $\overline{w}$ as the vector of the weights

$$\overline{w} := (\overline{w}_1, \ldots, \overline{w}_k)^T,$$

and the parameter vector of the convex combination can in consequence be written as

$$\pi := \left(\overline{w}^T, \pi_1^T, \ldots, \pi_k^T\right)^T \in \mathbb{R}^{\overline{s}},$$

with the number of parameters

$$\overline{s} := k + \sum_{i=1}^{k} \overline{s}_i.$$

6.5 Arm Models

Having introduced rigid body models for the bones and several muscle models, a two-dimensional and a three-dimensional arm model are presented in the following. Since the overall structure of a human arm is far too complex for our purposes, simplified models are used. Before stating the arm models used in the computations, a short introduction to the state of the art in modeling the human arm is given.

First of all, the goal of modeling the human as a dynamical system is to explain experimental data, explore hypotheses on human motion and generate ideas for future experimental work. To create such a model, several aspects have to be considered and a number of details have to be paid attention to, see for example the review [316]. All the choices of the modeler from model structure to mathematical description have to fit to the final application, so that the important features are well described and the behavior can be analyzed by suitable approaches [252]. Consequently, the dimension of the simulated model for most musculoskeletal models is smaller than that of the actual dynamical system [316].

Research of various groups is related to modeling and simulating the musculoskeletal systems of the human body, e.g., [191, 208, 252]. Due to the lack of a common framework, several individual computational programs are developed and novel approaches are fine-tuned for the specific approaches. A standard approach is to combine tools for graphical design of the musculoskeletal system (e.g. `SIMM` [71], `Any-Body`[68]) with computational packages for multibody dynamics (e.g. `Autolev`, `ADAMS`) to simplify the process of generating the equations of motions and to allow for generation of more complex models using an abstract layer. A step to a common framework is the `OpenSim`-software [70] which is freely available and open-source. In this thesis we do not make use of such a software package, because we are not only interested in the simulation of the dynamics, but also in the (second-order) derivative information of the overall arm movements.

The dynamics of the human skeleton are approximated in our arm models by rigid body dynamics (cf. section 6.2) assuming that the kinematic structure of the arm can be modeled by a chain of rigid bodies with joints of one degree of freedom only. This assumption constrains the modeling of the more complex human joints like the shoulder. The analysis of the kinematic and dynamical behavior of the human shoulder in [320, 321] shows that several mechanisms have to be modeled to capture the different characteristics of the system; especially, the joint between the thorax and the scapula has to be modeled as a gliding plane to reproduce the influence on motions and stabilization of the shoulder. Additionally, it might be necessary to model non-rigid effects and contact forces if there is a significant load on the articulating surfaces of the bones, e.g., in the human knee. A simple model of passive joint properties is included in our arm models and an extension to more complex models [191] is possible.

In section 6.3 models of human muscle dynamics are discussed and implementing more realistic models is always a critical challenge [316]. Closely related to the muscle dynamics is the problem of defining the attachment sites of the individual muscles. The number of attachment sites for human muscles differs from a single spot to long or broad areas. Most models assume that the muscle (including the tendon) is attached to the bone at single points or multiple discrete points. The routing in between the attachment site defines the length of the muscle and consequently the force that is exerted; the approaches modeling the muscle lengths range from straight lines [191] over via-point constructions [121] to cubic splines with sliding and surface constraints [293]. The problems of the muscle routing and the insertion angle of the muscle into the bone are closely related to the mechanical moment arms which consequently depend on the joint angles. According to [316], it might not be necessary to model the paths for certain applications but obtaining a mathematical expression for the moment arm itself could suffice. We follow here the line of [291, 292] and assume constant moment arms, but, again, an extension to more complex models is straight-forward.

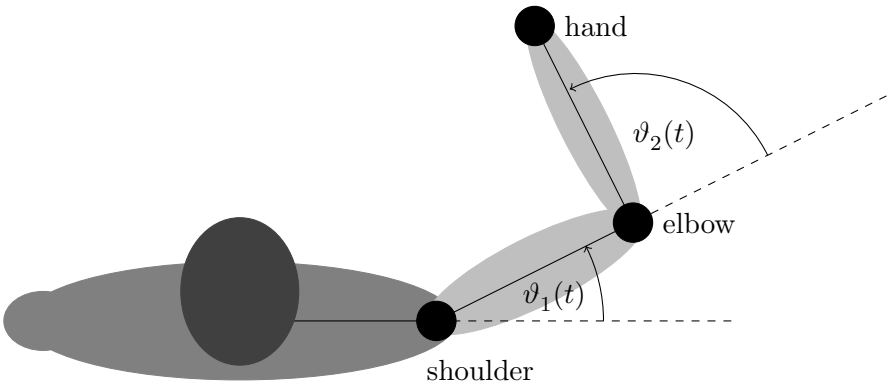


Figure 6.3: Schematic illustration of the joints of the planar arm model.

6.5.1 Planar Arm Model

In this section we state the planar arm model introduced in [291]. Most models used in literature for two-dimensional arm movements have two joints with one degree of freedom each: the shoulder and the elbow. Consequently, the shoulder angle is denoted by $\vartheta_1(t)$ and the elbow angle by $\vartheta_2(t)$ (see figure 6.3). The upper limb and the lower limb are both approximated by symmetric rigid bodies. Therefore, the lengths of limbs d_1 and d_2 are sufficient to state the three fixed parameters of the classical DH notation (cf. section 6.2.1).

	$i=1$	$i=2$
l_i	d_1	d_2
ρ_i	0	0
β_i	0	0

Table 6.1: The fixed DH parameters of the planar arm model.

Additionally, the link masses m_1, m_2, the corresponding inertias $\mathcal{I}^{(1)}$, $\mathcal{I}^{(2)}$ and the distances $d_{com,1}$, $d_{com,2}$ between the joints and the corresponding centers of mass are of interest for the derivation of the equations of motion (cf. section 6.2.2):

$$T(t) = M\left(\vartheta(t)\right)\vartheta''(t) + \Theta\left(\vartheta(t), \vartheta'(t)\right),$$

where M is the planar mass matrix, Θ the term combining Coriolis and centrifugal terms and T the joint torques. Note that assuming a symmetrical shape and a homogeneous mass distribution for each link, the center of mass of the limbs is in the center between the corresponding two joints: $d_{com,1} = 0.5d_1$, $d_{com,2} = 0.5d_2$. The mass matrix $M\left(\vartheta(t)\right)$ can be specified in an analytical form for this planar arm model (cf. section 6.2.2):

$$\begin{aligned}
M_{1,1}\left(\vartheta(t)\right) &= \mathcal{I}^{(1)} + \mathcal{I}^{(2)} + m_1\left(d_{com,1}\right)^2 \\
&\quad + m_2\left((d_1)^2 + \left(d_{com,2}\right)^2 + 2d_1 d_{com,2}\cos\left(\vartheta_2\right)\right), \\
M_{1,2}\left(\vartheta(t)\right) &= m_2 d_{com,2}\left(d_{com,2} + d_1\cos\left(\vartheta_2\right)\right) + \mathcal{I}^{(2)}, \\
M_{2,1}\left(\vartheta(t)\right) &= M_{1,2}\left(\vartheta(t)\right), \\
M_{2,2}\left(\vartheta(t)\right) &= m_2\left(d_{com,2}\right)^2 + \mathcal{I}^{(2)}.
\end{aligned}$$

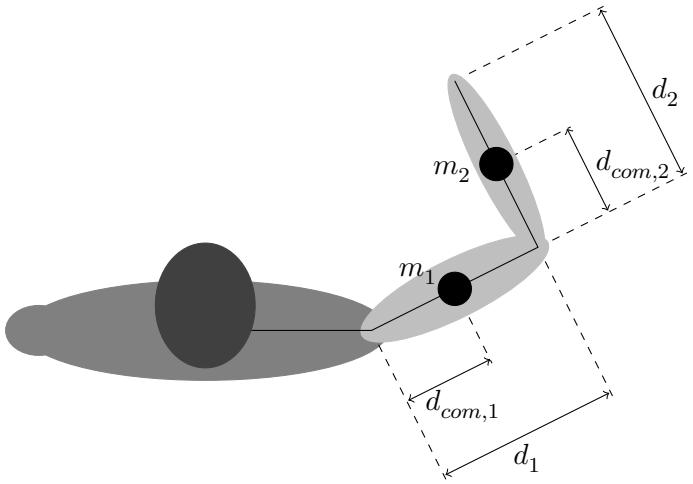


Figure 6.4: Schematic illustration of the parameters describing the arm's kinematics and dynamics.

The terms of $\Theta\left(\vartheta(t), \vartheta'(t)\right)$ are given by

$$\begin{aligned}\Theta_1\left(\vartheta(t), \vartheta'(t)\right) &= -m_2 d_1 d_{com,2} \vartheta_2{}'(t)\left(2\vartheta_1{}'(t) + \vartheta_2{}'(t)\right)\sin\left(\vartheta_2\right), \\ \Theta_2\left(\vartheta(t), \vartheta'(t)\right) &= m_2 d_1 d_{com,2}\left(\vartheta_1{}'\right)^2 \sin\left(\vartheta_2\right).\end{aligned}$$

According to the model presented in [291] three lumped muscle pairs are used to actuate the limbs; one single joint pair each for shoulder and elbow and a third muscle pair spanning both joints (see figure 6.5). Combining these lumped models with the nonlinear muscle dynamics of section 6.3.2, it is shown in [291, 292] that impedance characteristics of human arm motions can be captured by the model. Note that in this approach the simplifying assumption is made that each muscle has constant moment arms, i.e, a constant matrix R relates muscle forces to torques. In addition to the torques generate by the muscles, a simplified version of passive torques are considered. These passive torques describe the damping in the human joints counteracting motions. Consequently, a matrix B is introduced to relate joint velocities ϑ' to the passive torques; the actual torques acting on a joint are the sum of the torques generated by the muscles and the passive properties of the arm.

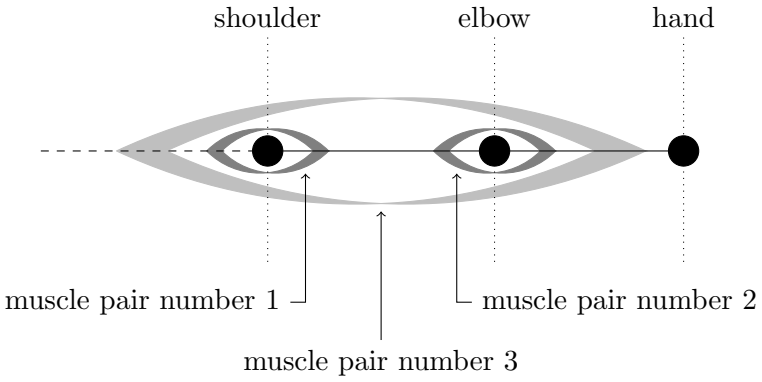


Figure 6.5: Schematic illustration of the lumped muscle pairs actuating the planar arm model.

6.5.2 Three-dimensional Arm Model

In the following a simple example of a three-dimensional arm model is introduced which can be easily extended, for example, to cases where more degrees of freedom have to be considered in order to optimize hand orientation in addition to the hand position. We assume for this model that the shoulder is a joint with three degrees of freedom, i.e., a ball in a socket joint, and the elbow has only one degree of freedom to allow for flection; for a discussion of these simplifying assumptions see the beginning of section 6.5.

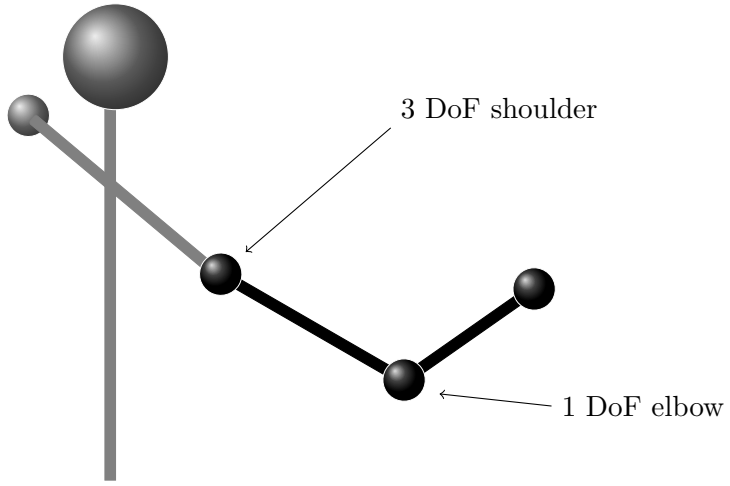


Figure 6.6: Schematic illustration of the rigid bodies in the three-dimensional arm model.

If the two rigid bodies of the upper and the lower limb are approximated by symmetrical cylinders, the following classical Denavit-Hartenberg parameters result using d_1 and d_2 to denote the limb lengths:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$
l_i	0	0	0	0	0	0	d_2
ρ_i	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0	0
β_i	0	0	0	d_1	0	0	0
ϑ_i	$\underline{\vartheta}_1$	$-\frac{\pi}{2}$	$\underline{\vartheta}_2$	$\underline{\vartheta}_3$	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$	$\underline{\vartheta}_4$

Table 6.2: The fixed DH parameters of the three-dimensional arm model.

Note that three virtual joints are used to model the arm configuration, i.e., the vector $\vartheta \in \mathbb{R}^7$ stating the Denavit-Hartenberg parameters for the joint angles has three fixed values and the free ones are denoted by $\underline{\vartheta} \in \mathbb{R}^4$. Similar to the two-dimensional case the following quantities are used to state the dynamics of the arm model: the limb masses m_1, m_2, the distances to the centers of mass $d_{com,1}$, $d_{com,2}$ and the inertia matrices $\mathcal{I}^{(1)}$, $\mathcal{I}^{(2)}$. However, the terms for the mass matrix and other parts are to lengthy to be listed here.

Note that modeling the muscles for such a three-dimensional problem is a complex task which should involve several model iterations by comparing model behavior with recorded human data; such a task is out of the scope of this work, thus we acknowledge that the number of considered lumped muscles and their moment arms might need improvement to be biologically plausible.

6.6 Numerical Results for Inverse Optimal Control

In the following section three numerical examples for inverse optimal control of arm motions are presented. First, the reconstruction of a planar arm motion is discussed, followed by inversion results for human arm motions recorded in a planar experiment. Finally, the reconstruction of a three-dimensional arm movement is addressed. For numerical inversion results of three-dimensional human arm data see section 6.7.

6.6.1 Reconstruction of Planar Arm Motions

The following first numerical example of a reconstruction for planar arm motions is kept at a minimum complexity to discuss the basic properties of the inversion task in a setting suitable for a detailed analysis. Consequently, we use the planar arm model introduced in section 6.5.1 in combination with four basic cost functions.

For numerical computations the following parameters of the planar arm model (cf. section 6.5.1) are chosen in accordance with the values used in the model [291].

Parameter	Value	
d_1	0.32	$[m]$
d_2	0.32	$[m]$
m_1	1.8	$[kg]$
m_2	1.6	$[kg]$
$\mathcal{I}^{(1)}$	0.015	$[kg\ m^2]$
$\mathcal{I}^{(2)}$	0.013	$[kg\ m^2]$

Table 6.3: Selected parameters for the planar arm model.

Additionally, the matrix $R \in \mathbb{R}^{2\times 3}$ stating the moment arms of the three lumped muscles and the matrix $B \in \mathbb{R}^{2\times 2}$ describing the joint damping properties have to be specified:

$$R = \begin{pmatrix} 0.03 & 0 & 0.025 \\ 0 & 0.03 & 0.04 \end{pmatrix}, \quad B = \begin{pmatrix} -0.3 & 0 \\ 0 & -0.2 \end{pmatrix}.$$

The task of the arm motions we are considering here is to start ($t = 0$ $[s]$) and stop ($t = 3$ $[s]$) at given positions, i.e., all state variables but the two joint angles have to be zero. For the arm configurations at these two time instances we choose the following values:

$$\vartheta(0) = \begin{pmatrix} 1.0 \\ 0.6 \end{pmatrix} \quad \text{and} \quad \vartheta(3) = \begin{pmatrix} 0.5\pi \\ 0.5\pi \end{pmatrix}.$$

No constraints on states or controls are considered here and the value of the motion time is fixed to 3 $[s]$.

Four basic cost functions are considered for this example including one minimizing the squared control values $\overline{u}$ of the linear muscles, which is here denoted by f_{MC}:

$$f_1 = f_{JJ}, \quad f_2 = f_{TC}, \quad f_3 = f_{HJ} \quad \text{and} \quad f_4 = f_{MC}.$$

The following figure 6.7 displays the optimal hand paths for these four basic cost functions. Note that each basic cost function yields an unique optimal hand path; such a property

would not result if, for example, only joint jerk of one joint would be considered instead of the combination used here. This special property, which does not hold true for some later examples, allows us to address the issue of relative scaling between the different cost functions. Naturally, one would like the numbers of the weight distribution $\overline{w}$ to be in close relation to the respective combination of the optimal hand trajectories. Since the different cost functions have considerably different scales (cf. table 6.4), such a behavior can only be obtained by using a scaling approach. One possibility is to introduce a scalar factor and a scalar shift for each cost function such that the image range of all cost functions is approximately the same. Here we use only a scalar scaling of the cost functions because these factors are visible in the KKT-conditions within the transformation approach. Therefore, the cost values corresponding to the solutions minimizing each basic cost function are compared and the scaling factor is obtained as the difference between the maximal and minimal value on the basic trajectories. In consequence, all weight distributions stated in this section refer to these scaled versions of the basic cost functions.

	$i=1$	$i=2$	$i=3$	$i=4$
$j=1$	$8.8\ 10^{0}$	$8.5\ 10^{-2}$	$6.1\ 10^{-1}$	$1.3\ 10^{3}$
$j=2$	$1.6\ 10^{2}$	$5.3\ 10^{-2}$	$6.5\ 10^{-1}$	$2.0\ 10^{3}$
$j=3$	$1.1\ 10^{2}$	$1.2\ 10^{-1}$	$2.3\ 10^{-1}$	$4.6\ 10^{3}$
$j=4$	$8.6\ 10^{1}$	$1.6\ 10^{-1}$	$1.2\ 10^{0}$	$7.0\ 10^{1}$

Table 6.4: The values of the basic cost functions f_i for the optimal values with respect to one of these costs f_j.

The following reconstruction example uses data that is obtained by solving the optimal control problem for the (randomly chosen) vector

$$\overline{w} := (0.3,\ 0.5,\ 0.2,\ 0)^T$$

and componentwise adding Gaussian white noise with a standard deviation of 0.01 times the mean difference between the successive values of the corresponding discretized components of the data. This added randomness should account for inaccuracies of models and measurements as they occur in the human experiments and thus artificial effects resulting from perfect fits are avoided in the reconstruction analysis.

The state minimizing the fourth cost function f_{MC} is chosen as a starting value for this example, because the distance of the corresponding optimal hand path to the data is the smallest. Note that in this example case this naive strategy to choose the starting value leads to a starting weight distribution considerably different from the data values. Furthermore, the time-discretization is kept constant with 20 segments of equal length and the Hermite-Simpson version is used in the collocation method. The distance measure Φ_{time} is used as the upper level cost function. The following difference is obtained in the weight distribution:

$$\left(9.74\ 10^{-5},\ 7.50\ 10^{-4},\ 8.48\ 10^{-4},\ 0\right)^T.$$

The numerical inversion result of the reconstruction data correctly states a zero weighting factor for the fourth basic cost function, which was chosen to be the starting value. This shows that the interior point approach in combination with the inverse optimal control approach does not lead to complications with respect to the bounds on the weight distribution.

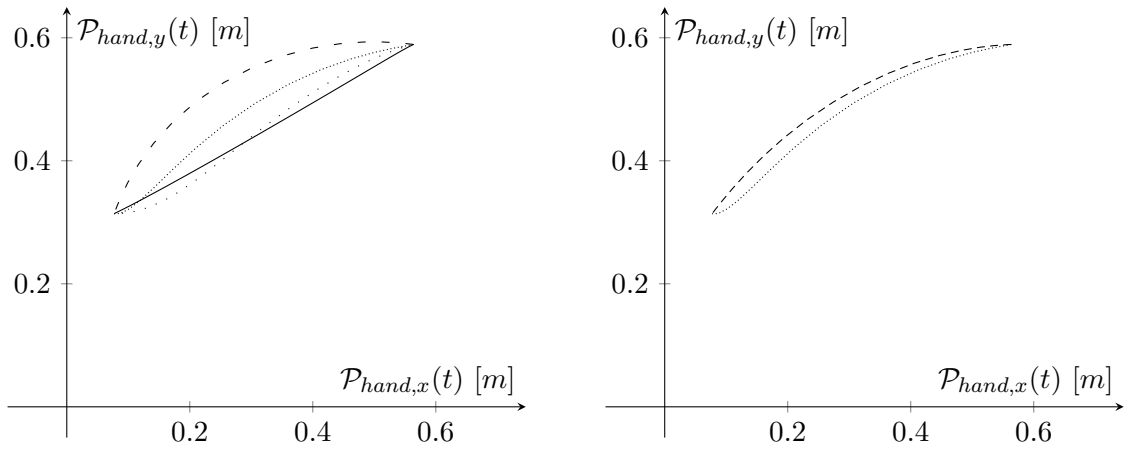


Figure 6.7: The optimal hand paths for the four basic cost functions f_i (left) and the reconstruction data with the corresponding starting value (right). The position of the shoulder is $(0.4, 0)$. [- - f_{JJ}, ····· f_{TC}, —— f_{HJ}, ····· f_{MC}, ---- data for reconstruction]

Using random weight distributions for the starting value generation, the dependence of the reconstruction results on the starting values can be analyzed. Thus the inversion problem is solved here a hundred times with random weight distributions obtained by considering uniformly distributed weights within the interval $[0, 1]$ and then normalizing the sum of them to 1. The difference of the mean values of the weight distributions corresponding to inverse optimal control solutions from the weight distribution used to generate the data in the given scenario is the following:

$$\left(1.73\ 10^{-4},\ 3.00\ 10^{-3},\ 1.89\ 10^{-3},\ 1.29\ 10^{-3}\right)^T.$$

The standard deviation of the obtained solutions is given by

$$\left(8.50\ 10^{-4},\ 9.94\ 10^{-3},\ 7.67\ 10^{-3},\ 2.34\ 10^{-3}\right)^T.$$

6.6.2 Inversion of Human Planar Arm Motions

The inversion of recorded human data is now exemplified for a planar arm motion. This example is taken from an experimental study done in a cooperation with C. Passenberg and the results considering different scenarios and participants are published in [6]. Especially, it has to be mentioned that C. Passenberg maintained the needed experimental hardware and conducted the actual recording of human motions.

The goal is to analyze human rest-to-rest movements recorded via an experimental setup that combines two linear actuators mounted at a right angle on top of each other. This setup in combination with a virtual environment reducing the visual feedback allows to study the influences of simulated masses and simulated damping on the human reaching motions. The example discussed here considers the baseline case (approximately) without external influences, for the interesting cases of different mass and damper combinations see [6]. The general motion task of the experiments is to move between three specified points; to decrease effects described by Fitts' law [96] (cf. section B.2.1), the target is specified as a square of

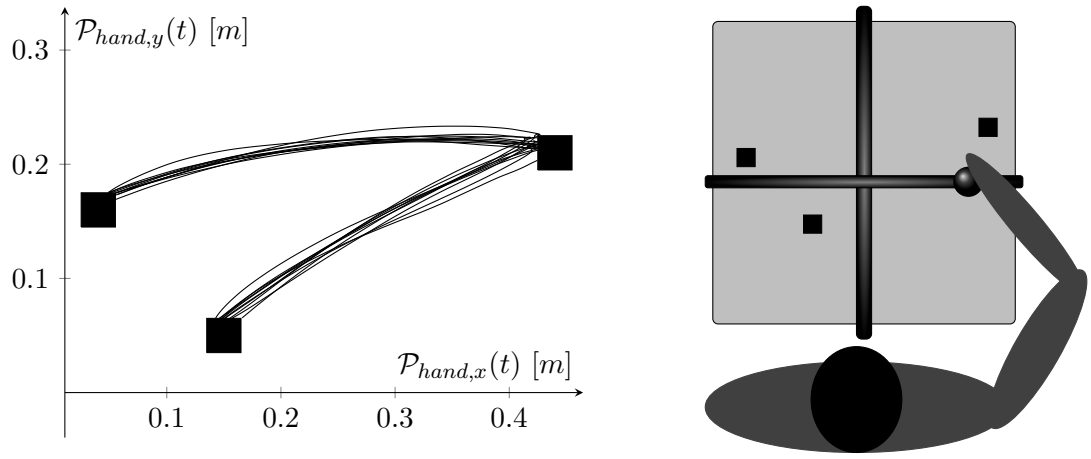


Figure 6.8: Characteristic recorded human data (left) with the shoulder being fixed at $(0.5, -0.25)$ and a schematic illustration of the experimental setup (right).

size 3 $[cm]$ by 3 $[cm]$. Figure 6.8 shows a schematic sketch of the experimental setup and additionally displays the hand paths of one participant showing that the trajectories have common characteristics.

One of the motion selected for a more detailed discussion is depicted together with the optimal trajectories for four basic cost functions in figure 6.10. The arm model and the family of cost functions is identical to the previous reconstruction example; however, the motion time and the boundary conditions are adapted according to the given data. Since only positional information of the recorded human motion is available, no consistent starting values for the muscle forces and their time-derivatives can be prescribed; consequently, only the joint angles and joint velocities are given as boundary conditions. Additionally, the hand paths are compared via the path length-based ULP cost function.

The inverse optimal control approach yields the following weight distribution for this example:

$$\overline{w} = (0.31, 0.44, 0, 0.25)^T .$$

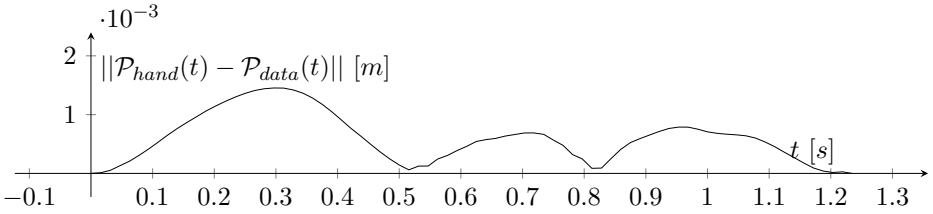


Figure 6.9: The distance of the solution from the recorded human data over time t.

The distance of the recorded hand path and the hand path corresponding to the solution of the inversion task is shown in figure 6.9. It shows the rather close fit between data and numerical result which could be expected considering the characteristics of the basic trajectories (cf. figure 6.10). The results of the whole experimental study [6] show that the common characteristics of the hand paths lead to similar weight distributions for the different trials. However, the recorded hand paths differ considerably between different participants

and consequently, the weight distributions are only similar for a specific participant. A possible application scenario for the information is a telepresence task where the operating human and the robotic hardware are spatially separated, resulting in a considerable delay in the feedback loop. To reduce the delay effects on the robot control, the previously computed human cost function could be used to predict the future motion of the operator. The predicted information can then be employed to control the robotic hardware until the human feedback is available.

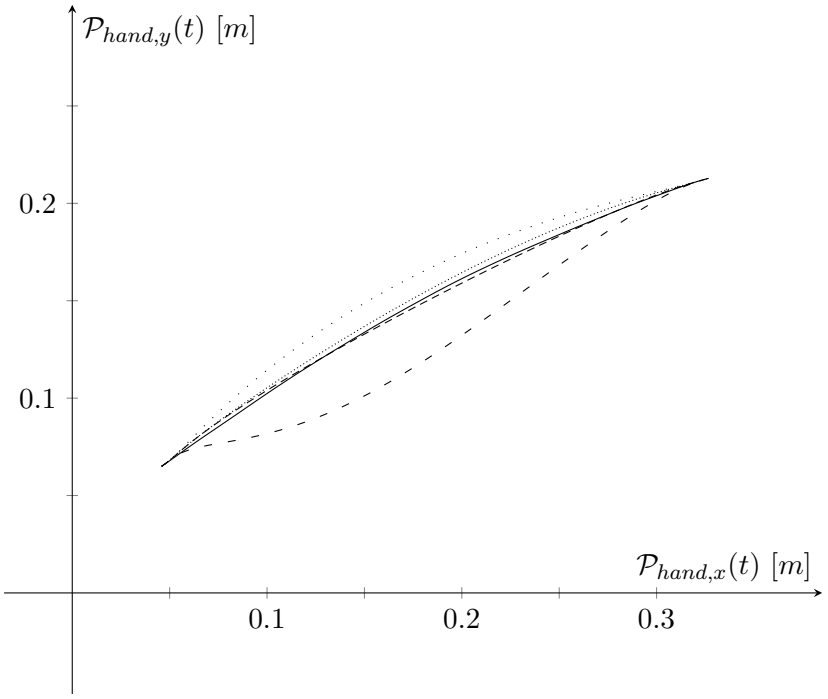


Figure 6.10: The optimal hand paths for the four basic cost functions f_i and the recorded human data. The position of the shoulder is $(0.4, -0.25)$. [f_{JJ}, f_{TC}, f_{HJ}, f_{MC}, —— data for reconstruction]

6.6.3 Reconstruction of Three-Dimensional Arm Motions

The three-dimensional arm model as a generalization of the two-dimensional one is discussed in section 6.5.2. Here we use five muscle pairs to actuate the two limbs with its four degrees of freedom. As discussed before, this is a rather large simplification, but it is not claimed that the model is biologically plausible with respect to this aspect.

The following matrices $R \in \mathbb{R}^{4\times5}$ and $B \in \mathbb{R}^{4\times4}$ state the moment arms of the lumped muscle pairs and the joint damping properties in a straightforward generalization of the values used in the planar model:

$$R = \begin{pmatrix} 0.03 & 0 & 0 & 0 & 0.025 \\ 0 & 0.03 & 0 & 0 & 0 \\ 0 & 0 & 0.03 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & 0.04 \end{pmatrix}, \quad B = \begin{pmatrix} -0.3 & 0 & 0 & 0 \\ 0 & -0.1 & 0 & 0 \\ 0 & 0 & -0.01 & 0 \\ 0 & 0 & 0 & -0.2 \end{pmatrix}.$$

In a similar manner the following values are used in the computations:

d_1	0.32	$[m]$
d_2	0.32	$[m]$
m_1	1.8	$[kg]$
m_2	1.6	$[kg]$

Table 6.5: Selected parameters for the three-dimensional arm model.

Finally, the inertia matrices of the upper and lower arm have to be stated; since the original values of the planar model correspond to inertias of sticks with negligible radii, small values are used for the inertia about the axes of symmetry. The quantities are stated in $kg\ m^2$:

$$\mathcal{I}^{(1)} = \begin{pmatrix} 0.015 & 0 & 0 \\ 0 & 0.015 & 0 \\ 0 & 0 & 0.001 \end{pmatrix}, \quad \mathcal{I}^{(2)} = \begin{pmatrix} 0.001 & 0 & 0 \\ 0 & 0.013 & 0 \\ 0 & 0 & 0.013 \end{pmatrix}.$$

Note that these inertia matrices are stated within the corresponding link frame of the DH notation. Consequently, the the first limb extends along the $\mathcal{Z}$-axis, whereas the second limb is positioned along the $\mathcal{X}$-axis of the corresponding frame.

In order to only allow for arm configurations corresponding to possible human poses (e.g., the rotation of the lower limb *through* the upper limb is theoretically possible from the DH notation, but definitely not realistic), the following upper and lower limits on the joint values are demanded in the numerical computation. However, in the given reconstruction example these bounds do not become active at any time.

$$\frac{\pi}{3} \geq \underline{\vartheta}_1 \geq -\frac{\pi}{6}, \quad \frac{\pi}{3} \geq \underline{\vartheta}_2 \geq -\frac{\pi}{6}, \quad \frac{\pi}{6} \geq \underline{\vartheta}_3 \geq -\frac{\pi}{3}, \quad \frac{3\pi}{4} \geq \underline{\vartheta}_4 \geq 0.$$

The motion task considered in this reconstruction is a rest-to-rest movement in the time period of 3 seconds, where the boundary values for the joint angles are the following:

$$\underline{\vartheta}(0) = \left(0,\ 0,\ 0,\ \frac{\pi}{2}\right)^T, \quad \underline{\vartheta}(3) = \left(\frac{\pi}{3},\ -\frac{\pi}{8},\ 0,\ \frac{\pi}{2}\right)^T.$$

First, we want to compare the optimal control results for the different muscle models. One model is obtained if the rigid body model is combined with five muscle pairs of the Stroeve model (cf. section 6.3.2); the other if the five lumped muscles are modeled by a linear ODE (cf. section 6.3.1). Considering the motion task and the family of cost functions of this example, both models yield similar results if the following parameters are used:

l_t	0.02	$[m]$
l_r	0.15	$[m]$
V_{sh}	0.3	$[\cdot]$
V_{shl}	0.23	$[\cdot]$
l_{sh}	0.6	$[m]$

l_{opt}	0.7	$[\cdot]$
V_{vm}	$6l_{co}$	$[m/s]$
V_{er}	0.5	$[\cdot]$
V_{ml}	1.3	$[\cdot]$
F_{max}	2000	$[N]$

Table 6.6: Selected parameters for the muscle model of Stroeve.

The following plot shows the differences between two hand paths minimizing the torque change criterion. One hand path is computed using the muscle model of Stroeve and the other using the simpler muscle model.

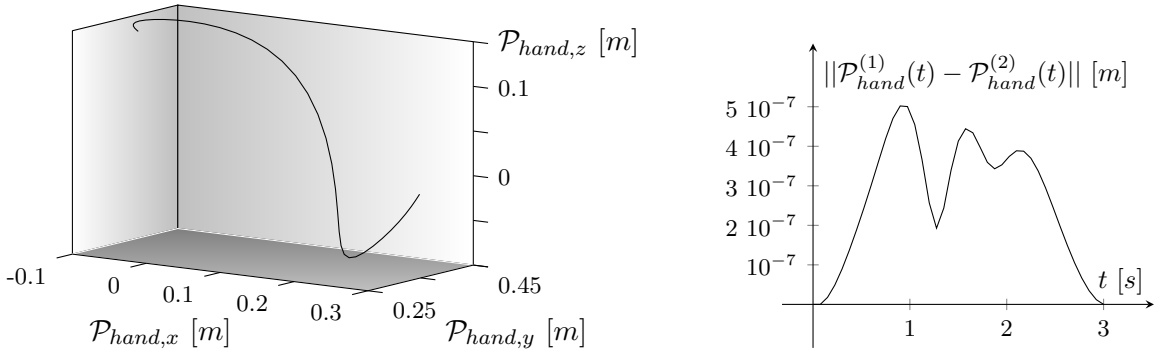


Figure 6.11: The hand path minimizing the torque change criterion where the shoulder is positioned in the origin (left) and the difference between the hand paths minimzing the torque change criterion using the two different muscle models (right).

This similarity can be interpreted in the following way: both muscle models allow for the same arm motion minimizing the integral of the squared torque derivatives, i.e., the nonlinearities of the Stroeve's model do not limit the force generation compared to the simpler linear ODE model. This observation is of course limited to the rather slow-paced arm motion without any external forces acting on the arm. However, this result suggests to use the simpler muscle model for the reconstruction task, because the resulting problem size is approximately half the size of the one based on the nonlinear muscle model.

The following basic cost functions are used in this example:

$$f_1 = f_{JJ}, \quad f_2 = f_{TC}, \quad f_3 = f_{HJ,xy} \quad \text{and} \quad f_4 = f_{MTC},$$

where the criterion $f_{HJ,xy}$ considers only the hand jerk in the horizontal plane. Consequently, it has to be noted that not all basic cost functions lead to a unique solution of the optimal control problem. If the hand jerk was minimized in three dimensions, this would lead to a unique hand path, but the elbow position is then not specified by this cost; since we consider only the jerk in horizontal directions, even one more degree of freedom is obtained. In consequence, the weight distributions have to be interpreted with care, i.e., the changes resulting from a modification of the weight combination should be analyzed by comparing the corresponding optimal states and hand paths.

The following table gives the weight distributions used to generate the data and the starting value of this reconstruction example:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
data value $\overline{w}_i$	0.1	0.2	0.4	0.3
starting value $\overline{w}_i$	1	0	0	0
difference in $\overline{w}_i$	$3.64\ 10^{-4}$	$7.60\ 10^{-4}$	$4.06\ 10^{-3}$	$5.18\ 10^{-3}$

Table 6.7: Weight distributions for data and starting value and the difference between the reconstruction result and the data values.

Both the hand path corresponding to the starting value minimizing joint jerk and the data of the reconstruction are displayed in figure 6.12. In the numerical computations a uniform time discretization with 40 segments is used and the upper level cost is computed by comparing hand positions at defined time instances. The differences in weight distribution of the reconstruction result compared to the distribution used to generate the data are considerably small, which is a result of extending the problem with an additional constraint. The idea of this goal attainment approach is discussed in section 5.2.3 and a number of optimization steps can be fixed after which this additional constraint is dropped to avoid linear dependencies in the KKT-conditions. Figure 6.12 shows the course of the upper level cost during the optimization runs where the number of iterations with the additional constraint is varied. It can be seen that without the additional constraint the optimization does not leave the neighborhood of the starting value and consequently, no solution is obtained in this case. Furthermore, the curves with 50, 100, 150 and 200 iterations using the additional constraint are shown. A forking can be observed if these maximal numbers are reached.

Note that if the additional constraint in this example is considered at least for the first 150 iterations, a solution of the problem is obtained within the smallest number of optimization method iterations. Problems with the linear dependence in the KKT-condition are not observed for the discussed example. In general, the choice of an suitable number of iterations with the additional constraint depends significantly on the dynamics of the problem and the boundary conditions of the motion task. The presented optimization of three-dimensional arm motions is a complex problem. Consequently, a larger number of iterations in the inversions are needed which results in larger values for the number of iterations with the additional constraint.

A more detailed discussion of the goal attainment approach for a similar three-dimensional arm motion problem can be found in [5].

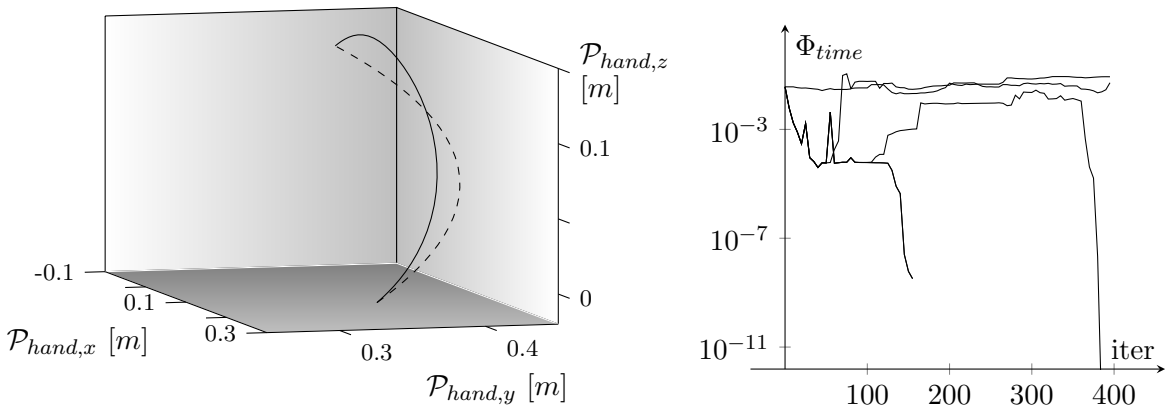


Figure 6.12: The hand paths of the data and the starting value with the shoulder at the origin (left) and the upper level cost function over the iteration number of the optimization (right). Optimization runs with different numbers of iterations using the additional constraint are displayed (Every fifth cost value is depicted). [- - -: start, ——: data hand path]

6.7 Transfer to Robotic Systems

In recent years a lot of research on human motions has been driven by the robotics community. This is due to the rise of humanoid robots which become more and more suitable to work in real world environments shared with humans. Naturally, several problems arise in order to suitably control such robotic systems. First of all, the tasks involve interaction with a priori unknown objects in a dynamically changing environment which makes replaying of beforehand defined movements nearly impossible. Second, sharing a common workspace with humans necessitates that the motions of each other can be anticipated. Third, the acceptance of a robotic system by the human depends on the hardware design and the realized movements and skills [81]. However, one has to keep in mind the effect known as the uncanny valley that the acceptance of a technical system being in many aspects very similar to the human, but not perfectly so, might cause a rejection by a human observer.

A standard way to address these issues is to build humanoid robotic systems with the goal of controlling them in a human-like fashion, but to observe the differences in the kinematics and dynamics between the robot and the human. Considering the large amount of possible tasks, the best way to control the robots is to learn from humans. A considerable amount of research in this direction is published; see section 6.7.2 for a few details on imitation learning. If selected motions can be learned from demonstration, the task remains to generate adaptively reasonable motions for new tasks. One way to do so is to use building blocks describing submotions. These motion primitives can then be combined to generate a variety of overall motions; a short introduction is given in section 6.7.1.

Even if the principles underlying human motions are known, a mapping problem has to be solved in order to use the same principles for robot control. Due to the different kinematic and dynamic properties of human and robot arms, a rather complex mapping process seems to be needed [81]. Mapping strategies presented in literature range from key posture usage in computer graphics [350] to multi-stage processes assuring feasibility and similarity [81]. The problem of animating non-human characters using human motion data is closely related to the robot control problem. On the one hand, the differences between the human properties and those of the character might be greater than those between human and humanoid robot, but, on the other hand, issues like collisions, joint limits and dynamical restrictions are not considered [250, 268, 350]. The focus of multi-stage processes used in robotics [35, 81] is on the latter aspects while trying to maintain a certain similarity between the robot posture and the given human one. Another approach to tackle the mapping problem is to design the robotic system accordingly. For example, the complex layout of the human shoulder allows for a large range of motions which can be captured by a robot with additional degrees of freedom [85].

6.7.1 Motion Primitives

The idea that humans use several separate models, called motion primitives, to accomplish subtasks is supported by two basic observations. First, going back to Bernstein [25], humans seem to follow mental templates of motion when executing motor tasks [103]. Second, a large number of different motions are used by humans for all kind of manipulation tasks in various environments [346]. Human motor coordination is known to develop gradually during postnatal life. Three concurrent steps are essential in the coordination of the sensorimotor systems [288]: a basic repertoire of spontaneous motions, the ability to sense the effects of

various movements and selection of the actual movement. Various experiments, e.g., [32, 142, 221, 222], for different kinds of motor systems suggest that voluntary actions are composed of simpler elements [103] and that the sensorimotor control is realized simultaneously on multiple levels [200]. Furthermore, experiments suggest that new and more complex primitives are obtained in a learning process by combining simpler motion primitives. Thus, the basic assumption of this section is that motion primitives combined to a sequence can accomplish a complete goal-directed movement [273] capturing human characteristics. Using a set of motion primitives reduces the dimensionality and complexity associated with the motion control problem [103]. In consequence, from the perspective of learning the primitives seem to be much more reasonable than pure trial and error learning [275]. For unsupervised learning methods such as reinforcement learning such complex motions might be computationally impossible [272].

The idea of motion primitives directly leads modular frameworks like the one in [346] where multiple forward and inverse models are used to build an enormous vocabulary of motor behaviors. A responsibility signal is proposed to select the suitable primitives based on the context. A related idea introduced by [145] uses a hierarchical structure with bidirectional information exchange and a responsibility function to select the correct primitives at each layer, which range from low level controllers to global task representations. In this context learning new motions coincides with finding the right responsibility function for this task. A functional hierarchy is also used by the models [195, 305] such that the lower level deals with the dynamics of the system and the upper level controls a simplified dynamical system. Unsupervised learning is used in [305] to build a compact model of the correlations between motor commands and sensory feedback. In [195] the applicability is shown for arm models of different complexities. Another approach to build a framework using motion primitives is to model the human anatomy. Three layers modeling the musculoskeletal plant, the spinal cord and the brain are used in [200] to approximate the hierarchical structure of the real control problem faced by the brain.

Several approaches, e.g., [69, 114, 162, 273, 274, 300], try to find a suitable description of primitives. If appropriate operations and transformations are used, these motion primitives can yield complex actions. A segmentation and classification algorithm computing the sequence of primitives that generate the movements is discussed in [69]. Similarly, in [114] a principal component analysis with a clustering technique is utilized to extract the primitives out of the complex motion. [300] use an approach based on Gaussian-like tuning functions to predict learning of hand motions. In contrast [273] use point attractors and limit cycles to describe the nonlinear behavior in rhythmic motions and point-to-point movements. Reinforcement learning is enabled by dynamic primitives based on point attractive systems [162, 274].

6.7.2 Imitation Learning

The classical manual programming of control strategies for robotic motions exceeds its possibilities if a large set of different motions has to be considered. Consequently, other methods are needed to generate the controls for the control of humanoid robots in everyday environments. The most simple approach to get such control strategies is trail-and-error learning or reinforcement learning where some reward function is maximized over various trials. Because robotic control problems are in general continuous in time, states and controls, most reinforcement learning strategies are based on a discretization of these quantities, which causes a

dimensionality problem. In case of a coarse discretization the control performance is poor and in case of a fine discretization the problem size explodes and many learning trials are needed. Thus, a number of strategies to circumvent this problem ranging from using prior knowledge to adaptive partitioning applications have been discussed, see, for example, the book [294] for reviews and details. A generalization of reinforcement learning to continuous problems is discussed in [83] where the Hamilton-Jacobi-Bellmann equation for infinite-horizon, discounted reward problems and suitable approximations are used.

In order to speed up the tedious learning process, imitation learning - also termed learning by demonstration - is proposed where the motion is demonstrated by a teacher and the student has to copy the motion. Such a learning strategy being natural to most humans is a demanding task if the student is a robotic system. With the goal to automate this learning strategy, all relevant elements from perception models to metrics for comparisons have to be known or consistently defined [273]. Additionally, it has to be assured that motions learned via imitation are generalizable to other contexts, because adaptivity and robustness are needed in dynamical environments [30, 246]. This imitation learning idea is in line with human-inspired robot design caused by the observation that "humans exhibit all the properties we want from a robot system in terms of adaptivity, learning capabilities, compliance, versatility, imitation and interaction capabilities etc." [281]. All variables defining the primitives to be learned have to be observable in imitation learning, leaving only kinematic variables as candidates [273]. To enable robotic systems to learn from humans, a framework is discussed in [80] which includes the whole loop from segmenting actions and understanding human intention, over learning and representing the task to mapping and executing the learned behavior. For further details on these methods and other learning frameworks see, for example, [14, 273, 281].

Using probabilistic representations of demonstrated motions, several approaches based on regression techniques are discussed in literature [53, 132, 327] and some successful generalizations to other situations are reported [53].

Another possibility to represent the relevant information of observed motions are motion primitives (cf. section 6.7.1). Then, in the learning phase the perceived movements are mapped onto the set of existing primitives [73, 246, 273, 341]. However, depending on the choice of the motion primitives, a correspondence problem between the different kinematic and dynamic systems might occur. Another approach to capture the learning process is to introduce a hierarchy of artificial neural networks structured in accordance with neurological structures [30, 280]. To enforce the plausibility from the biological perspective, the functionalities of the individual networks of [30] resemble those of specific brain regions and the parameters optimized by learning describe the connections between the networks. The idea of [280] is to learn models for the dynamics and the cost function without prior knowledge and then generate approximately optimal motions for the learned models.

A third major group of methods used in the context of imitation learning are strategies of inverse reinforcement learning introduced by [226]. This group assumes a discrete problem, in most cases in form of a Markov decision problem, and tries to find a cost function such that the demonstrated motion is optimal. [226] present a solution strategy and discuss the extension to a non-discrete setting using linear function approximations. If the optimal cost function is assumed to be a linear combination of known criteria, the problem is termed the apprenticeship learning problem [1]. A first solution strategy is presented by [1] and the later version of [296] is reported to improve the theoretical convergence order. In [295] the apprenticeship learning problem is given in form of a linear programming problem and

examples of learning demonstrated impedance characteristics are discussed. One has to note that the concept of inverse reinforcement learning can lead to ambiguities, since a demonstrated motion can be optimal for more than one cost function [358]. In [358] a probabilistic approach to the inverse reinforcement problem is used; utilizing approximate distributions of the paths in the MDP, the ambiguity in choosing a distribution is resolved by the principle of maximum entropy. Finally, the maximum margin planning framework [256, 257] tries to solve a problem very similar to the problem of the inverse reinforcement learning.

6.7.3 Human-like Optimal Control

In this section we want to discuss an example of how to use the inverse optimal control approach for human-like robot control. The research in this direction is done in close cooperation with the research team of M. Beetz that maintained the hardware and conducted and processed the human experiments; results exceeding the example discussed here are published in [7].

The general idea is to observe a human doing everyday manipulation tasks in a common environment - this special setup is focused on human motions in a kitchen. The recorded data suggests that human arm motions have stereotypic patterns in this environment (cf. figure 6.13). After recording and segmenting the motions of the humans, the inverse optimal control framework is used to compute the combination of the basic cost functions that comes closest to the data. Assuming that the used basic cost functions are transferable to a robotic manipulator, the combination of cost functions obtained by inversion of human data can be used to compute optimal trajectories for the robot respecting its dynamic properties.

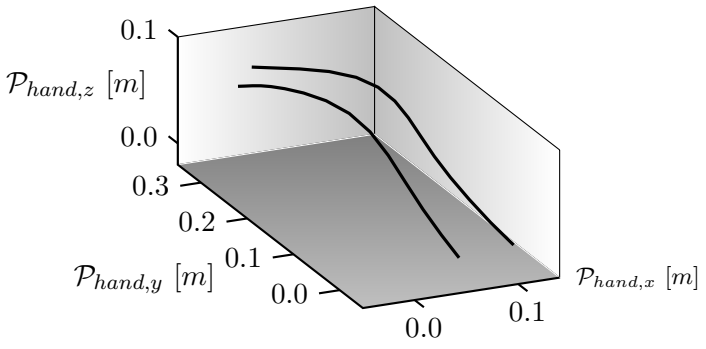


Figure 6.13: Two hand paths of recorded human reaching motions.

Since the recorded human motions are full three-dimensional arm motions, the three-dimensional arm model is used here (cf. section 6.5.2). This model combines two rigid links with several linear muscles and can only be viewed as a rather coarse approximation of the real dynamics of the human arm. The boundary conditions on the position and velocity of the hand at start and end are prescribed in accordance with the given data values. We restrict the considered basic cost functions to the rigid body level and avoid the problem of transferring cost functions based on muscle properties to the robotic system later on. Note that in literature no general cost functions for the three-dimensional arm movements are proposed different from the two-dimensional case, consequently, the planar ones like torque change and hand jerk are generalized to the three-dimensional case (cf. section 6.6.3).

The figure 6.14 shows some snapshots of a typical video sequence recorded in the kitchen while the participant sets the table. With data processing techniques the motion data is extracted from the video sequence and then the overall motion is segmented into individual motions. Given the hand path of the participant, the inversion technique yields the closest trajectory that is optimal with respect to a combination of the basic cost functions. It can be seen in figure 6.14 that the optimization result comes close to the recorded data.

Figure 6.14: Pictures of recorded human motions in the kitchen and the corresponding hand trajectories of the data (· · ·) and the inversion result (——).

The obtained optimal weight distribution is then used to generate optimal trajectories for a robotic manipulator. In this case an iCub robot (see figure 6.16) having kinematic and dynamic properties similar to a small human child is used to exemplify the control approach. The optimal control problem for the robotic system is solved with respect to the kinematic and dynamic properties of the robot; this provides the huge advantage over a simple mapping of recorded human joint values on the technical system that consistent motions are generated which make use of all the actuating properties of the robotic arm and, for example, do not violate bounds on the joint values. Futhermore, the inverse optimal control problem yields a human-like control strategy which can easily be adapted to changes in the boundary conditions by re-optimizing the corresponding optimal control problem. To visualize this adaptation to task changes, the hand paths for the iCub robot are displayed in figure 6.15 that are optimal with respect to the human-demonstrated cost function.

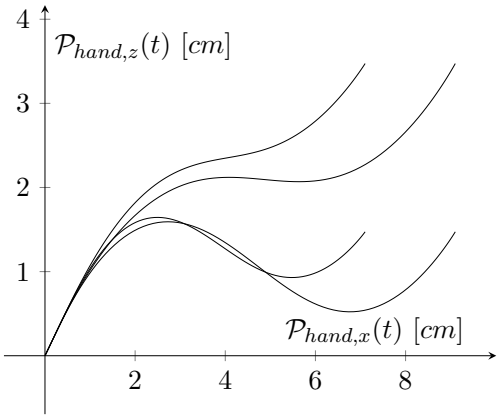


Figure 6.15: Projection of the three-dimensional hand paths to different final positions optimized for the iCub dynamics using the inversion result.

Finally, it has to be mentioned that the transfer of an optimal control problem to a real technical system is a challenging task where one has to consider the hardware specific control possibilities and the limitations resulting from actuation and model errors. However, figure 6.16 shows that the whole process from observing human motions to controlling the robot according to the inversion result has been accomplished for example motions (cf. [7]).

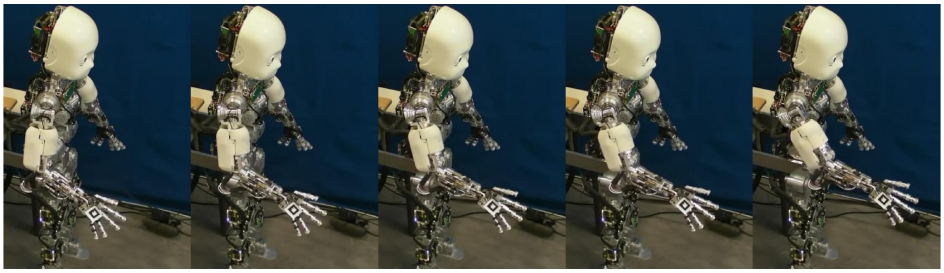


Figure 6.16: The iCub robot controlled according to a cost combination obtained by inversion of human data.

Human Car Driving

Chapter 7

The number of driver assistance systems assuring a higher level of security and improving the comfort of car driving is increasing steadily in modern cars. The goal of the systems is to support the driver, but not to limit possible maneuvers. In general, the tendency in car development is clearly towards (partly) autonomous systems, which leads to the problem of imitating human control behavior in order to increase the acceptance of such assistance systems by the human. Consequently, we frame this problem in the setup of our bilevel optimization approach and ask for the underlying optimization criteria reproducing data of human car driving.

The work presented in this chapter focuses on lane changes on a highway and is the result of a cooperation with Sven Kraus, Lehrstuhl für Fahrzeugtechnik, Technische Universität München. Selected results are already published in our paper [187], the diploma thesis of Weigl [336] and the dissertation of Kraus [186].

The structure of this chapter is as follows: First, a brief introduction of the state of the art with regard to lane change characteristics and optimal control modeling of the driver is given in section 7.1, then details on the autonomous car and the data acquisition and processing are discussed in section 7.2. Dynamical models of the car are derived in section 7.3 using the popular approach of a single-track model and finally, in section 7.4, aspects of formulating the bilevel problem are discussed and some optimization results are presented.

7.1 State of the Art

Important characteristics of a lane change are the amount of time needed for the maneuver and the lateral distance between the two lanes which is given by the structural condition of the street. Additionally, the geometrical and temporal properties of the trajectories have to be considered.

Several papers, e.g., [234, 289], analyze the time needed for the total movement, but the critical issue of how start and end of a lane change are determined is not discussed. Consequently, it is hard to compare the individual results and thus they can only be used as approximate values. In [289] values are reported to depend on the particular traffic situation and the range of possible values is from 3.5 seconds to 6.5 seconds; in situations of emergency one might even observe lane changes within a two-second period.

Approaches to model the geometry of lane changes originate mostly in the accident reconstruction research. A simple approach is to combine two segments of a circle with opposite curvatures, but the non-continuous curvature at the connection results in unrealistic controls.

More realistic results can be obtained if segments of clothoids are used instead of circle segments, because they guarantee the continuity and the piecewise linearity of the curvature. This approach is used to design the layout of streets in the construction process, but results in a mathematically complex problem without even considering the dynamical properties of a car. Consequently, other approaches are discussed in literature. For example, a straight line is combined with a sinusoidal segment in [205] or a polynomial approach is used to model the geometry of the trajectory in [258].

The goal of a driver model is to capture the complex human behavior which is highly adaptable to the task at hand and the corresponding traffic situation. A detailed discussion of approaches based on an optimal control framework can be found in [50].

7.2 Experimental Vehicle

The experimental car used to record the human steering data is developed as a part of the German trans-regional research cooperation SFB/TR-28 "Kognitive Automobile". It is a modified Audi Q7 3.0 TDI with various sensors and actuators which allow autonomous control; for details on this experimental vehicle see [301]. The sensors allow to measure all relevant dynamical quantities of the car motion.

The data of the human lane change maneuvers was recorded on the German highway A9 using segments with a totally straight layout. To reduce the noise in the measurements, the data is smoothed with a low-pass filter according to the maximal frequency realizable by a human controlling a car [151]. Furthermore, to reduce the dead time of the camera-based measurements a Kalman filter is used; see [187] for details.

A central aspect in analyzing human-steered lane changes is identification of the time instances of start and end. It is a hard task to obtain good approximation of these time instances; for example, the triggering of the direction indicator is not a suitable indicator for the start of the lane change, because the triggering only indicates the intention to change the line, but not the actual start of the maneuver. Similarly, the measured values of individual states do not reliably define the start or end of the motion. In consequence, we use thresholds for several states, but nevertheless the obtained values have to be considered as approximations.

7.3 Dynamical Car Model

A common approach to model the dynamical behavior of cars is to consider single-track models where the pairs of wheels on each axle are virtually replaced by a single one in the center of the respective axle. Such a simplification is possible if the rolling and pitching angles of the car are small throughout the motion. In the following sections we will derive a system of equations modeling the nonlinear behavior of such a single-track model and then discuss a linearizaton of these equations.

7.3.1 Nonlinear Single-Track Model

The modeling of the single-track car follows the line of [124]. The following quantities are relevant to state the dynamical equations of the system: The mass of the car m is abstracted in the virtual center of mass which lies between the front and rear axle with a distance of l_r and

l_f. The position of the center of mass at time instance t is given by $\mathcal{P}(t) = (\mathcal{P}_x(t), \mathcal{P}_y(t)) \in \mathbb{R}^2$ in Cartesian coordinates. The angle between the $\mathcal{P}_x$-axis of this coordinate system and the single-track axis is the yaw-angle $\beta(t)$. Using this yaw angle, one computes the positions of the virtual front and virtual rear wheel given by $\mathcal{P}_f(t)$ and $\mathcal{P}_r(t) \in \mathbb{R}^2$, respectively. One of the controls of the car is the steering angle $\vartheta(t)$ which is the angle between the direction of the front wheel and the single-track axis. For all three positions $\mathcal{P}(t)$, $\mathcal{P}_r(t)$ and $\mathcal{P}_f(t)$ the corresponding velocities are defined by $v(t) = (v_x(t), v_y(t))$, $v_f(t)$ and $v_r(t)$. Each of these velocities draws a (slip) angle with the single-track axis denoted by $\alpha(t)$ and $\alpha_r(t)$ or with the direction of the wheel $\alpha_f(t)$.

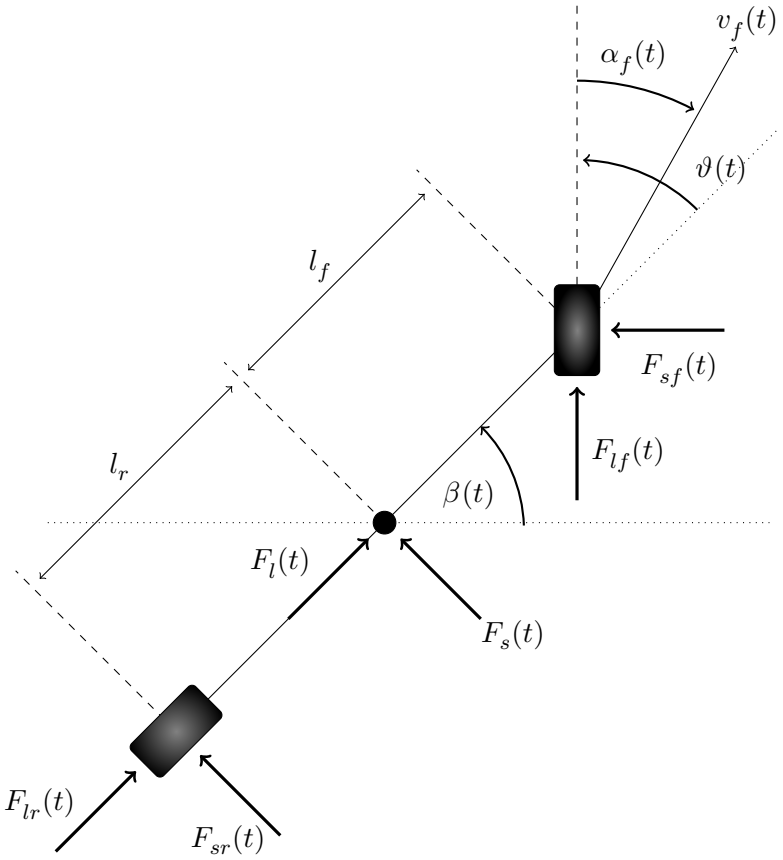


Figure 7.1: Schematic model of the single-track car.

Note that the slip angle $\alpha(t)$ is given by

$$\alpha(t) = \beta(t) - \arctan\left(\frac{v_y(t)}{v_x(t)}\right)$$

and the absolute value of the velocity of the center of mass follows from

$$|v|(t) = \sqrt{(v_x(t))^2 + (v_y(t))^2}.$$

Furthermore, the slip angles at the front axle and the rear axle are given by

$$\alpha_f(t) = \vartheta(t) - \arctan\left(\frac{l_f\beta'(t) - |v|(t)\sin(\alpha(t))}{|v|(t)\cos(\alpha(t))}\right) \tag{7.1}$$

and

$$\alpha_r(t) = \arctan\left(\frac{l_r\beta'(t) + |v|(t)\sin(\alpha(t))}{|v|(t)\cos(\alpha(t))}\right). \tag{7.2}$$

In addition to these kinematic properties of the single-track model, several forces have to be introduced. First of all, the longitudinal forces $F_{lf}(t)$ and $F_{lr}(t)$, and the side forces $F_{sf}(t)$ and $F_{sr}(t)$ on the respective wheels. Furthermore, air resistance of the car results in a drag force opposing the current motion. Here in this model we assume that such forces have only an effect along the single-track axis: denote by $F_A(t) \geq 0$ the absolute value of the force and by l_A the distance between the center of mass and the drag mount point. Finally, the actual force working on the center of mass $F(t) = (F_l(t), F_s(t)) \in \mathbb{R}^2$ in car-based coordinates is the sum of all forces working on the car:

$$\begin{aligned} F_l(t) &= F_{lr}(t) + F_{lf}(t)\cos(\vartheta(t)) - F_{sf}(t)\sin(\vartheta(t)) - F_A(t), \\ F_s(t) &= F_{sr}(t) + F_{lf}(t)\sin(\vartheta(t)) + F_{sf}(t)\cos(\vartheta(t)). \end{aligned}$$

As a consequence of Newton's law, the following differential equation describes the translational motion of the center of mass:

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{P}(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2}\begin{pmatrix} \mathcal{P}_x(t) \\ \mathcal{P}_y(t) \end{pmatrix} = \frac{1}{m}\begin{pmatrix} F_l(t)\cos(\beta(t)) - F_s(t)\sin(\beta(t)) \\ F_l(t)\sin(\beta(t)) + F_s(t)\cos(\beta(t)) \end{pmatrix}. \tag{7.3}$$

The rotational dynamics of the car are captured by the equation

$$\begin{aligned} \mathcal{I}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\beta(t) &= F_{sf}(t)l_f\cos(\vartheta(t)) - F_{sr}(t)l_r \\ &\quad + F_{lf}(t)l_f\sin(\vartheta(t)) - F_A(t)l_A, \end{aligned} \tag{7.4}$$

where $\mathcal{I} > 0$ is the inertia constant of the car for rotations about the center of mass.

In the following formulas are given for the individual forces acting on the car. A standard approach to model the drag due to air resistance is

$$F_A(t) = \frac{1}{2}c_w\rho_A A|v|^2,$$

where c_w is the air drag coefficient of the car, ρ_A the air density and A the effective surface on which the air resistance is working.

The forces acting upon the wheels in longitudinal direction are the sum of the forces generated by the rolling resistance of the wheels ($F_{Rr}(t)$ and $F_{Rf}(t)$) and the acceleration and braking forces controlled by the driver ($F_{Dr}(t)$ and $F_{Df}(t)$):

$$F_{lf}(t) = F_{Df}(t) - F_{Rf}, \quad F_{lr}(t) = F_{Dr}(t) - F_{Rr}.$$

In addition to the steering angle $\vartheta(t)$, the model has a second control $F_D(t)$ which is the total forces generated by accelerating or braking; positive values refer to acceleration of the car and negative ones to a speed reduction. Since the force distribution between front and rear axle is seldom uniform, we use the following approximations for our experimental vehicle (cf. section 7.2):

$$F_{Df}(t) = \begin{cases} \frac{1}{3}F_D(t) & \text{if} \quad F_D(t) > \varepsilon, \\ \frac{1}{2}F_D(t) - \frac{1}{4\varepsilon}F_D(t)^2 + \frac{1}{12\varepsilon^3}F_D(t)^4 & \text{if} \quad |F_D(t)| \leq \varepsilon, \\ \frac{2}{3}F_D(t) & \text{if} \quad F_D(t) < -\varepsilon, \end{cases}$$

$$F_{Dr}(t) = \begin{cases} \frac{2}{3}F_D(t) & \text{if} \quad F_D(t) > \varepsilon, \\ \frac{1}{2}F_D(t) + \frac{1}{4\varepsilon}F_D(t)^2 - \frac{1}{12\varepsilon^3}F_D(t)^4 & \text{if} \quad |F_D(t)| \leq \varepsilon, \\ \frac{1}{3}F_D(t) & \text{if} \quad F_D(t) < -\varepsilon, \end{cases}$$

where $\varepsilon > 0$ is a small number, e.g., $\varepsilon = 0.01$ $[N]$. The interval $[-\varepsilon, \varepsilon]$ is used to generate a continuously differentiable transition between braking and acceleration behavior. If necessary, the above polynomial could be replaced by a higher order one to assure that $F_{Df}(t)$ and $F_{Dr}(t)$ are twice differentiable with respect to $F_D(t)$.

For both the rolling resistance and the lateral forces on the wheels we use simpler models than the ones presented in [124] which are based on the works of [243] and [260]. First, the relations between the sidewards forces and the slip angles of the wheels are assumed to be linear and the model seems to capture the main characteristics of the here analyzed car maneuvers with reasonable small slipping:

$$F_{sf}(t) = c_f \alpha_f(t) \quad \text{and} \quad F_{sr}(t) = c_r \alpha_r(t), \tag{7.5}$$

where the constants are $c_f = 1.3 \cdot 10^5$ $[N/rad]$ and $c_r = 2.55 \cdot 10^5$ $[N/rad]$ in accordance with measurements of the behavior of the experimental vehicle. Second, it is assumed that the rolling resistance is independent of the velocity of the car. Consequently, the forces are the product of the coefficient of the rolling resistance c_R and the static loads on the wheels:

$$F_{Rf} = c_R \frac{m\underline{g} l_f}{l_f + l_r} \quad \text{and} \quad F_{Rr} = c_R \frac{m\underline{g} l_r}{l_f + l_r},$$

where $\underline{g} = 9.81$ $[m/s^2]$ is the gravitation constant and measurements of the rolling resistance coefficient result in $c_R = 2.2 \cdot 10^{-2}$.

In addition to the control $F_D(t)$ influencing the velocity of the car, the second derivative of the steering angle $\vartheta(t)$ is second control variable. Summing up, the following system of ordinary differential equations is used to model the dynamics of the car:

	Parameter	Value	
m	mass of car	$2.92 \cdot 10^3$	$[kg]$
$\mathcal{I}$	inertia of car	$5.561 \cdot 10^3$	$[kg\ m^2]$
l_f	distance between center of mass and front axle	1.533	$[m]$
l_r	distance between center of mass and rear axle	1.457	$[m]$
A	effective surface causing air drag	2.87	$[m^2]$

Table 7.1: Model parameters for the experimental vehicle

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \mathcal{P}_x(t) \\ \mathcal{P}_y(t) \\ \beta(t) \\ v_x(t) \\ v_y(t) \\ \beta'(t) \\ \vartheta(t) \\ \vartheta'(t) \end{pmatrix} = \begin{pmatrix} v_x(t) \\ v_y(t) \\ \beta'(t) \\ \left(F_l(t)\cos(\beta(t)) - F_s(t)\sin(\beta(t))\right)/m \\ \left(F_l(t)\sin(\beta(t)) + F_s(t)\cos(\beta(t))\right)/m \\ \Big(F_{sf}(t)l_f\cos(\vartheta(t)) - F_{sr}(t)l_r \\ \qquad + F_{lf}(t)l_f\sin(\vartheta(t)) - F_A(t)l_A\Big)/\mathcal{I} \\ \vartheta'(t) \\ \vartheta''(t) \end{pmatrix}.$$

Defining the state vector by

$$\overline{x}(t) := \big(\mathcal{P}_x(t), \mathcal{P}_y(t), \beta(t), v_x(t), v_y(t), \beta'(t), \vartheta(t), \vartheta'(t)\big) \in \mathbb{R}^8$$

and the control vector by

$$\overline{u}(t) := \big(F_D(t), \vartheta''(t)\big) \in \mathbb{R}^2,$$

the dynamics of the car can be written in the standard form

$$\overline{x}'(t) = \varphi(\overline{x}(t), \overline{x}'(t))$$

with $\varphi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ combining all terms discussed in this section.

7.3.2 Linear Single-Track Model

In this section we simplify the nonlinear car dynamics by linearization; the resulting linear single-track model is a common model to control the lateral dynamics of a car and differences in the bilevel optimization results between the two models are discussed in section 7.4.3. The two basic assumptions legitimating the linearization of the dynamics are that the velocity $|v|(t)$ is constant and that the angle $\beta(t) + \alpha(t)$ is small. Note that these assumptions result in a motion of the car mainly in the $\mathcal{X}$-direction.

We start the approximation process with the lateral forces on the wheels. The nonlinear formulas for the slip angels of the front wheel and the rear wheel are the equations (7.1)

and (7.2). Using the assumption that the angles are small, one can approximate the two equations by

$$\alpha_f(t) \approx \vartheta(t) - \frac{l_f\beta'(t) - |v|(t)\alpha(t)}{|v|(t)}$$

and

$$\alpha_r(t) \approx \frac{l_r\beta'(t) + |v|(t)\alpha(t)}{|v|(t)}.$$

Combining these with the equations (7.5), the following linear equations for the lateral forces are obtained:

$$\begin{aligned} F_{sf,lin}(t) &:= c_f\vartheta(t) + c_f\alpha(t) - \frac{c_f l_f}{|v|(t)}\beta'(t), \\ F_{sr,lin}(t) &:= c_r\alpha(t) + \frac{c_r l_r}{|v|(t)}\beta'(t). \end{aligned}$$

These terms lead to the linear version of the total side force working on the center of mass of the car:

$$F_{s,lin}(t) := F_{sf,lin}(t) + F_{sr,lin}(t).$$

Note that the longitudinal forces have to be zero, because otherwise they would cause a velocity change which would violate one of the basic assumptions.

If one assumes that the absolute value of velocity is given by the constant $|v|$, the velocity vector of the car is given by :

$$v(t) = \begin{pmatrix} v_x(t) \\ v_y(t) \end{pmatrix} = |v| \begin{pmatrix} \cos(\alpha(t) + \beta(t)) \\ \sin(\alpha(t) + \beta(t)) \end{pmatrix}. \tag{7.6}$$

Consequently, the acceleration of the car is given by

$$a(t) = \begin{pmatrix} a_x(t) \\ a_y(t) \end{pmatrix} = |v| \begin{pmatrix} -\sin(\alpha(t) + \beta(t)) \\ \cos(\alpha(t) + \beta(t)) \end{pmatrix} (\alpha'(t) + \beta'(t)).$$

Using the assumption that $\beta(t) + \alpha(t)$ is small, the following linear relation is obtained for the acceleration in $\mathcal{Y}$-direction:

$$a_y(t) \approx |v|(\alpha'(t) + \beta'(t)).$$

In a similar manner, the Newton equation (7.3) can be approximated by

$$a_y(t) \approx \frac{F_s(t)}{m}.$$

Combining these equations we obtain the following linear relation:

$$\alpha'(t) = \frac{c_f + c_r}{m|v|}\alpha(t) + \left(\frac{c_r l_r - c_f l_f}{m|v|^2} - 1\right)\beta'(t) + \frac{c_f}{m|v|}\vartheta(t). \tag{7.7}$$

The second dynamical equation to linearize is equation (7.4) which leads to

$$\mathcal{I}\beta''(t) \approx F_{sf,lin}(t)l_f - F_{sr,lin}(t)l_r,$$

if the forces due to air drag are neglected. In consequence, the following linear relation is determined for the rotational dynamics:

$$\beta''(t) = \frac{c_f l_f - c_r l_r}{\mathcal{I}}\alpha(t) - \frac{c_r (l_r)^2 + c_f \left(l_f\right)^2}{\mathcal{I}|v|}\beta'(t) + \frac{c_f l_f}{\mathcal{I}}\vartheta(t). \tag{7.8}$$

Combining the linearized $\mathcal{Y}$-component of equation (7.6), i.e.,

$$v_y(t) \approx |v|(\alpha(t) + \beta(t)),$$

with the linear versions of equations (7.7) and (7.8) yields the following linear ordinary differential equation:

$$\overline{x}'_{lin}(t) = \overline{A}\overline{x}_{lin}(t) + \overline{B}\overline{u}_{lin}(t),$$

where the state vector and its time-derivative are defined by

$$\overline{x}_{lin}(t) := \begin{pmatrix} \alpha(t) \\ \beta'(t) \\ \vartheta(t) \\ \beta(t) \\ \mathcal{P}_y(t) \end{pmatrix} \quad \text{and} \quad \overline{x}'_{lin}(t) := \begin{pmatrix} \alpha'(t) \\ \beta''(t) \\ \vartheta'(t) \\ \beta'(t) \\ v_y(t) \end{pmatrix},$$

and the scalar control $\overline{u}_{lin}(t) \in \mathbb{R}$ is the rotational velocity of the steering wheel $\vartheta'_w(t)$; the ratio between $\vartheta'_w(t)$ and $\vartheta'(t)$ is given by the constant $\zeta > 0$, i.e.,

$$\zeta\vartheta'(t) = \vartheta'_w(t).$$

The time-independent matrices $\overline{A} \in \mathbb{R}^{5\times5}$ and $\overline{B} \in \mathbb{R}^{5\times1}$ have the following structure:

$$\overline{A} := \begin{pmatrix} \frac{c_f + c_r}{m|v|} & \frac{c_r l_r - c_f l_f}{m|v|^2} - 1 & \frac{c_f}{m|v|} & 0 & 0 \\ \frac{c_f l_f - c_r l_r}{\mathcal{I}} & -\frac{c_r (l_r)^2 + c_f \left(l_f\right)^2}{\mathcal{I}|v|} & \frac{c_f l_f}{\mathcal{I}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ |v| & 0 & 0 & |v| & 0 \end{pmatrix},$$

$$\overline{B} := \left(0, 0, \frac{1}{\zeta}, 0, 0\right)^T.$$

Note that this linear model of the car dynamics can be extended to motions along curved paths by introducing a reference trajectory and considering its curvature [336]. For the experiments analyzed in section 7.4.3 the assumption of a straight street is sufficient.

7.4 Car-Steering Bilevel Problem

In the following the elements of the car-steering bilevel problem are discussed and numerical results for both the linear and the nonlinear version of the car dynamics are presented. The focus of section 7.4.1 is on the definition of the parameterized family of cost functions for the optimal control problem and in section 7.4.2 a suitable distance measure for the upper level problem is discussed. Inverse optimal control results can be found in section 7.4.3.

7.4.1 LLP Formulation

The problem of optimal car control being the lower level problem of the considered bilevel problem is the combination of a cost function to minimize, a system of ordinary differential equations modeling the car dynamics and several boundary conditions describing the driving task. In section 7.3 two models of the car dynamics are derived; the linear version models only the lateral dynamics of the car while assuming a constant velocity, but the nonlinear model considers both longitudinal and lateral dynamics allowing for more control of the driving maneuver. Consequently, the considered cost functions and the boundary conditions have to be chosen in accordance with the respective model.

We start with the discussion of cost functions for the linear model, because these criteria can easily be transferred to the nonlinear model, but not vice versa. Naturally, the standard cost functions are the minimization of selected states, their time-derivatives or controls. The cost functions related to an economical driving style are the minimization of the steering angle $\vartheta(t)$ or the angular steering velocity at the steering wheel $\vartheta'_w(t)$:

$$\begin{aligned} f_{\vartheta}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} \vartheta(t)^2 \, \mathrm{d}t, \\ f_{\vartheta'}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} \vartheta'_w(t)^2 \, \mathrm{d}t. \end{aligned}$$

On the other hand, the stability of the driving maneuver being a safety-related issue is captured by cost functions using either the yaw angle $\beta(t)$, the slip angle $\alpha(t)$ or their time-derivatives $\beta'(t)$ and $\alpha'(t)$:

$$\begin{aligned} f_{\beta}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} \beta(t)^2 \, \mathrm{d}t, \\ f_{\alpha}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} \alpha(t)^2 \, \mathrm{d}t, \\ f_{\beta'}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} \beta'(t)^2 \, \mathrm{d}t, \\ f_{\alpha'}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} \alpha'(t)^2 \, \mathrm{d}t. \end{aligned}$$

A more comfort-related cost function is the minimization of the lateral acceleration $a_y(t)$ which leads to the cost function

$$f_{ay}(\overline{x}_{lin}, \overline{u}_{lin}) := \int_{t_0}^{t_f} a_y(t)^2 \, \mathrm{d}t = \int_{t_0}^{t_f} |v|^2 (\alpha'(t) + \beta'(t))^2 \, \mathrm{d}t.$$

Another set of cost functions is obtained if the deviation from a reference trajectory is considered, cf. [50]. Here we consider trajectories resulting from combining two polynomials of fourth order according to [258]. The connection point of the two polynomials corresponds to the sign change of the steering angle. An asymmetrical geometry is observed in human-steered lane changes by [289] and it is reported that the length of the time-interval corresponding to the second trajectory part is twice the length of the first one. This behavior is captured by the trajectory $\tilde{r}_{0.33} : [t_0, t_f] \to \mathbb{R}$. Additionally, we introduce a trajectory $\tilde{r}_{0.70} : [t_0, t_f] \to \mathbb{R}$ where the switching point occurs at seventy percent of the total movement time. Such a more

racy driving behavior could be necessary in crowded traffic situations. The corresponding cost functions are:

$$\begin{aligned} f_{\tilde{r},0.33}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} (\mathcal{P}_y(t) - \tilde{r}_{0.33}(t))^2 \, \mathrm{d}t, \\ f_{\tilde{r},0.70}(\overline{x}_{lin}, \overline{u}_{lin}) &:= \int_{t_0}^{t_f} (\mathcal{P}_y(t) - \tilde{r}_{0.70}(t))^2 \, \mathrm{d}t. \end{aligned}$$

Note that all the quantities minimized in these cost functions are also available in the nonlinear car model and consequently, the nonlinear equivalents of these cost functions can be defined straightforwardly. However, the following cost functions make use of the velocity or positional information of the nonlinear dynamics and are therefore not suitable for the linear case.

Given the positions of the car $\mathcal{P}(t) = \big(\mathcal{P}_x(t), \mathcal{P}_y(t)\big)$ for all $t \in [t_0, t_f]$, the cost function minimizing the jerk of the car is given by

$$f_{jerk}(\overline{x}, \overline{u}) := \int_{t_0}^{t_f} \left(\frac{\mathrm{d}^3}{\mathrm{d}t^3} \mathcal{P}_x(t) \right)^2 + \left(\frac{\mathrm{d}^3}{\mathrm{d}t^3} \mathcal{P}_y(t) \right)^2 \, \mathrm{d}t.$$

Further cost functions result if the accelerating forces F_D controlled by the driver are minimized

$$f_{acc}(\overline{x}, \overline{u}) := \int_{t_0}^{t_f} (F_D(t))^2 \, \mathrm{d}t$$

or the overall control input is considered:

$$f_u(\overline{x}, \overline{u}) := \int_{t_0}^{t_f} (F_D(t))^2 + \big(\vartheta''(t)\big)^2 \, \mathrm{d}t.$$

Note that a relative scaling of the different control variables in f_u might be needed. The last cost function we want to add is the deviation of the car's absolute velocity value from a reference velocity $\underline{v} \geq 0$:

$$f_{\underline{v}}(\overline{x}, \overline{u}, \underline{v}) := \int_{t_0}^{t_f} (|v|(t) - \underline{v})^2 \, \mathrm{d}t.$$

Given these basic cost functions, the parameterized family of cost functions for the lower level problem is obtained as convex combinations of the basic ones. To fully state the optimal control problem, boundary conditions on the state are needed; we specify these conditions in the section on the corresponding numerical results.

7.4.2 ULP Formulation

The definition of a distance measure is needed for the upper level program which captures the important aspects of the deviations between the trajectory computed by optimal control and the recorded trajectory. In general, one could consider the weighted combination of the distances of all state values, but this would lead to the problem of determining a suitable weight combination. Furthermore, the scales and the precisions of the measurements differ between the states, thus finding a reasonable combination of weights is difficult.

In consequence, we here use the only the reliable positional information to compare the recorded data with the computed values. In case of the linear dynamics the comparison is

limited to the $\mathcal{Y}$-component only, while for the more general nonlinear model the Euclidean distance between the two points in the horizontal plane is used.

Note that (at least in the nonlinear problem) different goals might be accomplished in the upper level problem depending on the choice of the actual distance measure. If points of equal path length are compared, the focus is on reproducing the Cartesian path of the data, but the temporal relation is of no importance. Contrarily, if distances of positions at the same time instances are used, the temporal aspects gain more importance than the pure positional information. For a more detailed discussion of distance measures for the upper level problem see the section 4.3.

7.4.3 Numerical Results

The following numerical examples address the reconstruction and inversion of recorded lane changes for both dynamical models. We start with an reconstruction example for the linear model and present an inversion result considering only the lateral dynamics of the car. This is followed by a reconstruction of a lane change using the full nonlinear car model and finally, this model is used for inversion of recorded human data.

7.4.3.1 Reconstruction of a Lange Change for the Linear Model

The linear model is based on the assumption of a constant velocity $|v|$ and in this first reconstruction example the value for this velocity is given by $|v| = 30\ [m/s] = 108\ [km/h]$. Prescribing a fixed motion time of $t_f = 3\ [s]$ for the maneuver, a distance of approximately $90\ [m]$ results in forward direction. In this example the lane change corresponds to a difference of $3.5\ [m]$ between start and goal position in sideward direction, i.e., the following boundary conditions are considered:

$$\overline{x}_{lin}(0) = (0, 0, 0, 0, 1)^T \quad \text{and} \quad \overline{x}_{lin}(t_f) = (0, 0, 0, 0, 4.5)^T .$$

The parameters defining the dynamical behavior of the linear car model are already stated in table (7.1). Four basic cost functions f_i are considered to define the parameterized family of cost functions for this example:

i	1	2	3	4
f_i	$f_{\beta'}$	$f_{\vartheta'}$	f_α	f_y

For the data generation the weight distribution

$$\overline{w} = (0.3,\ 0.5,\ 0,\ 0.2)^T$$

is chosen and the starting value is generated by minimizing the cost function corresponding to the weight distribution

$$(0.5,\ 0,\ 0.5,\ 0)^T .$$

The curves of lateral positions $\mathcal{P}_y(t)$ corresponding to the data and the starting value are displayed in figure 7.2. Adding white noise of the magnitude $10^{-3}\ [m]$ to the generated data, the following differences between the weight distribution of the data and of the numerical result are obtained:

$$\left(1.44\ 10^{-2},\ 1.10\ 10^{-2},\ 6.85\ 10^{-6},\ 3.43\ 10^{-3}\right)^T .$$

The corresponding differences in the lateral positions which are the only quantities considered in the ULP cost function are also shown in figure 7.2. Note that the result is comparatively good, because the maximal differences are of the order of 1 millimeter which is about the noise level added to the generated data.

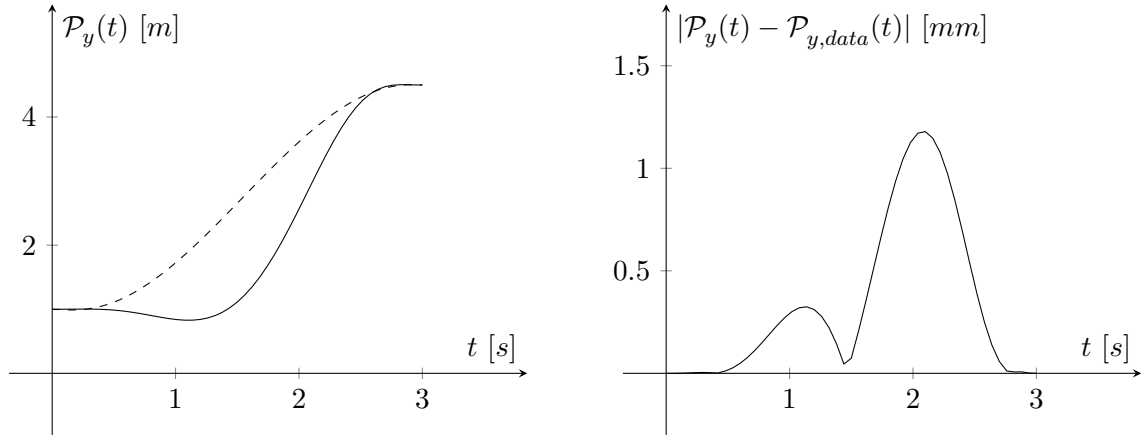


Figure 7.2: The lateral positions corresponding to the data and the start value for the reconstruction (left) and the differences in the lateral positions between the data and the reconstruction result (right). [- - -: starting value, ——: generated data]

7.4.3.2 Inversion of Lateral Car Data

The inversion of a human-steered lane change is discussed in the following. Figure 7.3 shows the recorded lateral positions for several lane changes and common characteristics can be observed.

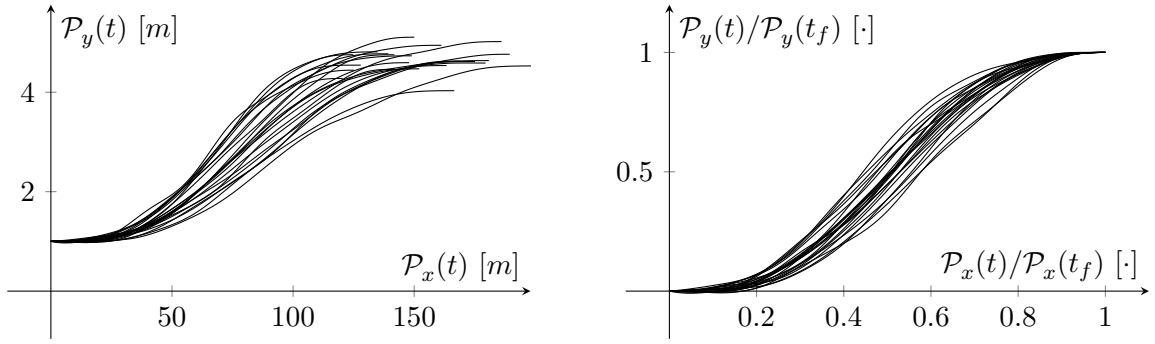


Figure 7.3: Recorded position data of human-steered lane changes (left) and normalized versions (right).

Note that the distances needed in forward and sideward direction differ slightly for the recorded lane changes. This variation can be explained by the slightly different street configurations, other traffic participants and the variance of human perception and control.

The general setting for the inversion task is identical to the previous section on the reconstruction of data simulated with the linear model, i.e., the family of cost functions, the dynamical model and the general structure of the boundary conditions stays the same. Only the final value $\mathcal{P}_y(t) = 4.5289\ [m]$ and the motion time $t_f = 5.6\ [s]$ are adapted to fit to the given data. Different weight distributions including the one used in the reconstruction example lead to the following inversion result:

$$\overline{w} = (0,\ 1,\ 0,\ 0)^T .$$

This means the basic cost function $f_{\vartheta'}$ minimizing the squared control input yields the lateral car position closet to the recorded human data; see figure 7.4 for plots of the recorded and computed lateral positions and the corresponding control inputs. Note that due to the linear dynamics some basic cost functions introduced in section 7.4.1 lead to similar lateral car positions and consequently, the number of considered basic cost function is rather limited. To analyze whether the linear model of the car dynamics is an adequate simplification, the following sections will address the full nonlinear car model.

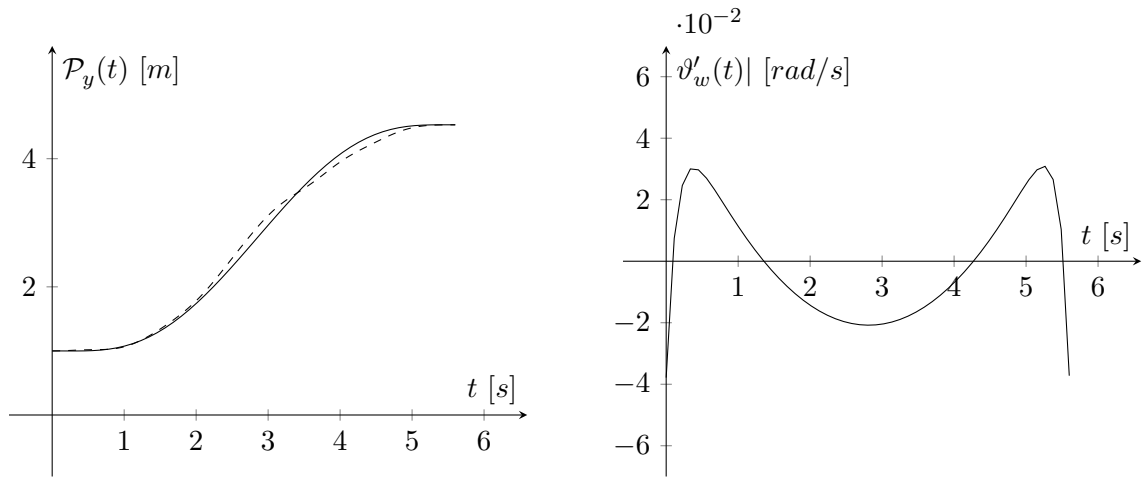


Figure 7.4: The lateral positions corresponding to the recorded data and the inversion result (left) and the corresponding optimal control (right). [——: numerical result, - - -: recorded data]

7.4.3.3 Reconstruction of a Lange Change for the Nonlinear Car Model

The main differences between the nonlinear and the linear model for the car dynamics are the time-dependent velocity of the car and the representation of the car positions. In the nonlinear case the car position is given by a point in a plane instead of a lateral position only. Consequently, the nonlinear model allows to consider more driving maneuvers by controlling the acceleration of the car in addition to the steering angle of the front wheels.

The following reconstruction results for the nonlinear car model consider a lane change in $t_f = 5\ [s]$; the corresponding distances of the maneuver are 150 $[m]$ in forward direction and 3.5 $[m]$ in sideward direction. The sideward velocity of the car is prescribed to be 0 $[m/s]$ at both start and end, while the forward velocity has to be 30 $[m/s]$ at both time instances;

note that a constant velocity is not feasible for the given task. Consequently, the following boundary conditions result:

$$\overline{x}(0) = (0, 1, 0, 30, 0, 0, 0, 0)^T \quad \text{and} \quad \overline{x}(t_f) = (150, 4.5, 0, 30, 0, 0, 0, 0)^T .$$

In this example we introduced the following bounds on the states and controls, but they do not become active for the actual reconstruction:

$$(-1, -2, -0.5\pi, 15, -10, -1, -0.5\pi, -0.5)^T \leq \overline{x}(t) \leq (151, 5, 0.5\pi, 45, 10, 1, 0.5\pi, 0.5)^T$$

and

$$(-15000, -1000)^T \leq \overline{u}(t) \leq (5000, 1000)^T .$$

Again, four basic cost functions are selected for the example:

i	1	2	3	4
f_i	f_y	f_β	f_{jerk}	f_u

The corresponding optimal trajectories of the car using the nonlinear model are displayed in figure 7.5 and similar to the planar arm motion example scaled versions of the weight distributions are considered in the following. The corresponding scales are determined by comparing the values of each cost function for the different basic trajectories (cf. section 6.6.1). The upper level cost function Φ_{time} is used that measures the distances between the data and the LLP state by comparing the corresponding tuples $\big(\mathcal{P}_x(t), \mathcal{P}_y(t)\big)^T \in \mathbb{R}^2$ of the car positions for given equidistant time instances.

The data trajectory of the reconstruction example is generated by optimizing the cost function given by the (scaled) weight distribution

$$\overline{w} = (0.05,\ 0.4,\ 0.25,\ 0.3)^T .$$

The starting value of the inversion corresponds to the weight distribution $(0,\ 0,\ 0,\ 1)^T$.

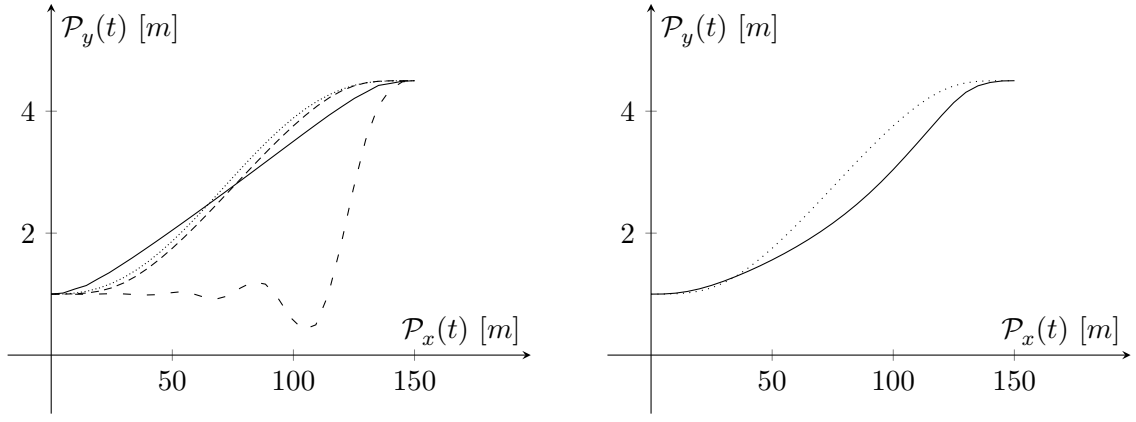


Figure 7.5: The car positions minimizing the cost functions [- - f_1, —— f_2, ····· f_3, ---- f_4] (left) and the car positions of the reconstruction example [····· starting value, —— data] (right).

The car trajectories corresponding to the starting value and the data are displayed in figure 7.5. Identically to previous reconstructions white noise of the magnitude 10^{-2} times the mean difference of the individual data component is added to the simulated data to avoid artificial effects resulting from a perfect model fit, i.e., the magnitude is about 1 $[cm]$ in $\mathcal{Y}$-direction and about 0.5 $[m]$ in $\mathcal{X}$-direction. The following differences between the weight distribution of the data and of the numerical result are obtained:

$$\left(1.10\ 10^{-3},\ 8.55\ 10^{-3},\ 3.63\ 10^{-2},\ 4.59\ 10^{-2}\right)^T.$$

This result indicates that our solution approach is able to determine the weight distribution of a simulated lane change well enough and consequently, we advance to the inversion of recorded human data.

7.4.3.4 Inversion of Planar Car Position Data

This section combines the inversion setup for the nonlinear car model as discussed in the previous section with the data used in section 7.4.3.2. The goal is to address the differences between the inversion results for the two car models. The inversion problem for the linear car model is solved by the cost function minimizing the control input (cf. section 7.4.3.2). Since in the nonlinear car model the car acceleration is as a second control variable in addition to the derivative of the steering wheel angle, the generalization f_u could be expected to yield a good starting value for the inversion. Figure 7.6 displays the car trajectories corresponding to the starting value, the data and the result of the bilevel optimization which is given by the weight distribution

$$(0,\ 0,\ 0.391,\ 0.609)^T.$$

The solution corresponds to a upper level cost value of 1.18 where a combination of the control input and the jerk of the (planar) car position is minimized. Since both cost functions yield similar trajectories and positional jerk is not available in the linear case, this result can be interpreted as a generalization of the result obtained for the simpler linear car model.

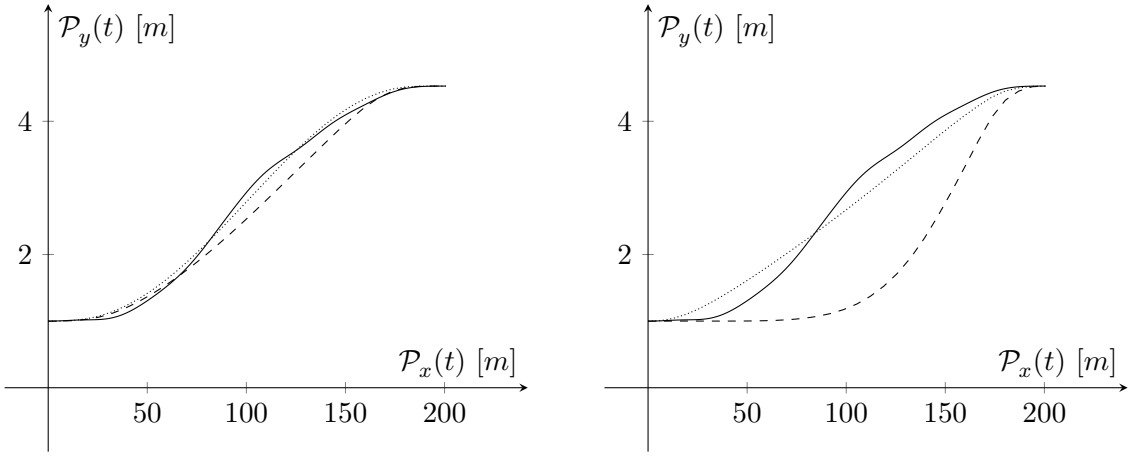


Figure 7.6: The car positions of the starting value, the inversion result and the data for the first weight distribution (left) and the second one (right). [- - - starting value, ····· inversion result, —— data]

However, random starting values show that the above mentioned starting value yields a local solution with a ULP cost value being approximately 1.5-times larger than the value obtained for other starting values. Exemplarily, we mention here a starting value corresponding to the weight distribution

$$(0.15,\ 0.3,\ 0.45,\ 0.1)^T .$$

The inversion result corresponding to this starting value has an ULP cost value of 0.73 and is depicted together with the data in figure 7.6. The weight distribution of this inversion result is the following:

$$\left(8.24\ 10^{-3},\ 7.85\ 10^{-1},\ 2.06\ 10^{-1},\ 0\right)^T .$$

A comparison of the two plots of figure 7.6 alone does not explain the differences in the upper level cost function. A further central aspect influencing Φ_{time} is the velocity profile of the car and figure 7.7 shows that the velocity profile of the second starting value is especially in the first half closer to the recorded data than the one of the first starting value. However, the differences are relatively small compared to the variances in the human data which clearly shows the critical dependence of the upper level cost function on the velocity profile.

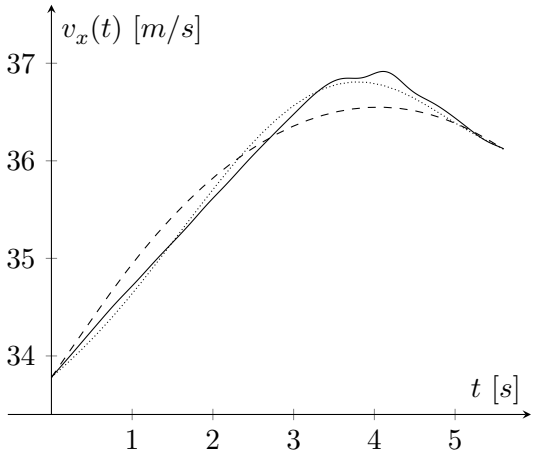


Figure 7.7: The $\mathcal{X}$-component of the car velocity for the two inversion results and the data. [- - - first starting value, ····· second starting value, —— data]

In consequence, this example illustrates on the one hand that the choice of the distance measure influences considerably the result and has therefore to be selected in accordance with the overall goal of the inversion. If one is interested only in the spatial component, the upper level cost function should use a path representation in path length (cf. section 4.3). A combination of such a cost function using only spatial information with one comparing the velocity profiles separately allows for a relative weighting and might therefore be better suited for certain applications (cf. for example the locomotion example in chapter 8). On the other hand, the observation of a local minimum resulting from the combination of temporal and spatial aspects is the reason for always testing multiple starting values in the inversion and reconstruction examples.

Human Locomotion

Chapter 8

Human locomotion considers the overall problem of walking from a start position to a goal position, without paying attention to the complex dynamical problem of taking individual steps. Consequently, we will introduce a simple model of the locomotion dynamics, the unicycle model, where the person is abstracted to a mass point with an orientation (cf. section 8.1).

The idea of determining the optimal cost function used in human locomotion via inverse optimal control is introduced in [217]. There, obstacle-free paths are considered and the family of cost functions is given as linear combinations of five basic cost functions. The bilevel problem is solved by nesting the individual solvers for the data fitting problem and the optimal control problem. It is reported that the characteristics of the human motion data are met and the results are used to control a humanoid robot (see section 4.2.4 for more details). The goal of our research presented in this section is to extend the problem class to navigation problems with moving obstacles, e.g., crossing persons. Therefore, we consider additional cost functions and treat some of the modeling parameters as further optimization variables, which introduces additional nonlinearities with respect to the parameters of the family of cost functions. Furthermore, we extend the optimal control idea to a model predictive control framework where the control strategies are updated during the motion according to newly available information on the trajectory of the interferer.

Note that parts of this section are already published in [4].

8.1 Locomotion Dynamics

Simplifying the human navigation problem to a two-dimensional problem, the configuration of the participant can be described by his/her Cartesian coordinates $\mathcal{P}_P(t) = (\mathcal{P}_{Px}(t), \mathcal{P}_{Py}(t)) \in \mathbb{R}^2$ and its orientation $\beta_P(t) \in [0, 2\pi]$ at a time instance t; in the following, the direction given by the angle $\beta_P(t)$ is referred to as the forward direction.

The considered model assumes that the rigid body can only be (linearly) accelerated in the forward direction and consequently, the following ordinary differential equations state the dynamics related to a translation of the rigid body:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{P}_P(t) = (v_P(t)\cos(\beta_P(t)), v_P(t)\sin(\beta_P(t)))^T$$

and

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}v_P(t) = \overline{u}_v(t),$$

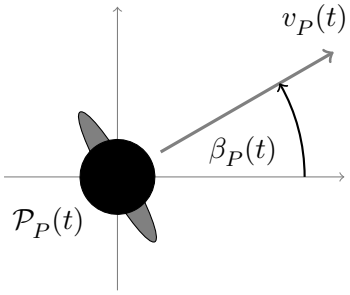


Figure 8.1: Schematic illustration of the unicycle model.

where $\overline{u}_v$, one of the two control variables, is the jerk in forward direction. The rotational dynamics are modeled by the simple differential equation

$$\frac{\mathrm{d}^3}{\mathrm{d}t^3}\beta_P(t) = \overline{u}_\beta(t),$$

where $\overline{u}_\beta$ is the second control variable. Note that these simple integrator chains can easily be extended to a model using the mass m and the inertia $\mathcal{I}$ of the moving person. In a similar manner motions in sideward directions can seamlessly be included in this model. However, the focus of this chapter is on more general properties of human locomotion and therefore more complex models are set aside.

In consequence, the following system of first-order ordinary differential equations describes the model dynamics: $\frac{\mathrm{d}}{\mathrm{d}t}\overline{x}(t) = \overline{A}(\beta_P(t))\overline{x}(t) + \overline{B}\overline{u}(t)$, where the matrices are given by

$$\overline{A}(\beta_P(t)) := \begin{pmatrix} 0 & 0 & 0 & \cos(\beta_P(t)) & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin(\beta_P(t)) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \overline{B} := \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and the state $\overline{x}(t)$ and the control $\overline{u}(t)$ are defined by

$$\overline{x}(t) := (\mathcal{P}_{Px}(t), \mathcal{P}_{Py}(t), \beta_P(t), v_P(t), {\beta_P}'(t), a_P(t), {\beta_P}''(t))^T$$

and $\overline{u}(t) := (\overline{u}_v(t), \overline{u}_\beta(t))^T$, accordingly.

8.2 Cost Functions

The following basic cost functions are used in our computations of the locomotion problem to build the parameterized family of LLP costs. In contrast to human arm motions where several cost functions explaining the overall motion are given in literature, we have to start with generic cost functions, because no single optimization criterion is proposed to explain human locomotion. The most classic cost functions are minimization of a state or a control variable:

$$f_{x,j}(\overline{x}) := \int_{t_0}^{t_f} \overline{x}_j(t)^2 \ \mathrm{d}t \quad \text{and} \quad f_{u,j}(\overline{u}) := \int_{t_0}^{t_f} \overline{u}_j(t)^2 \ \mathrm{d}t.$$

In our setting this could, for example, correspond to minimization of forces and torques. In addition, considering deviation from a reference value $\underline{r}_j \in \mathbb{R}$ leads to the definition of further cost functions. One realization of such a cost function could be motivated by the tendency to walk at a comfortable walking speed:

$$f_{ref,j}(\overline{x}) := \int_{t_0}^{t_f} \left(\overline{x}_j(t) - \underline{r}_j\right)^2 \, \mathrm{d}t.$$

Note that the reference value $\underline{r}_j$ corresponds to a nonlinear parameter being optimized in the bilevel optimization. Another considered cost function is the deviation from a straight line connecting start and goal positions.

$$f_{line}(\overline{x}) := \int_{t_0}^{t_f} ||\mathcal{P}_P(t) - \mathbb{P}_{line}(\mathcal{P}_P(t))||^2 \, \mathrm{d}t,$$

where $\mathbb{P}_{line}$ is the projection on the straight line connecting the start and goal position. Furthermore, the cost function f_{goal} introduced by [217] integrates the squared difference between the current orientation $\beta_P(t)$ and the direction towards the goal position $\mathcal{P}_G = (\mathcal{P}_{Gx}, \mathcal{P}_{Gy}) \in \mathbb{R}^2$:

$$f_{goal}(\overline{x}) := \int_{t_0}^{t_f} \left(\beta_P(t) - \arctan\left(\frac{\mathcal{P}_{Gy} - \mathcal{P}_{Py}(t)}{\mathcal{P}_{Gx} - \mathcal{P}_{Px}(t)}\right)\right)^2 \, \mathrm{d}t.$$

Since the navigation tasks are considered to have a free final time t_f, the minimization of this final time gives a further basic cost function:

$$f_{time}(\overline{x}) := t_f.$$

8.2.1 Modeling the Interferer

In addition to the introduced cost functions for the optimal control problem a further cost function is needed to model the influences of the interfering person on the path of the participant. Note that the interferer does not communicate with the participant and is walking at a (approximately) constant velocity along a straight line.

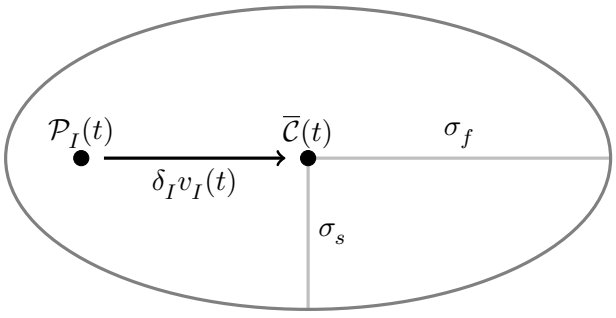


Figure 8.2: The parameters of the Gaussian model for the interferer.

The interferer cost is modeled by a normalized Gaussian centered at the point $\overline{\mathcal{C}}(t)$ which is computed from the current position of the interferer $\mathcal{P}_I(t)$, and its velocity $v_I(t)$:

$$\overline{\mathcal{C}}(t) = \mathcal{P}_I(t) + \delta_I v_I(t),$$

where the characteristic time $\delta_I \geq 0$, an optimization variable, is used to model the off-center position observed in experiments. This takes into account the observation that the participants tend to pass behind the interferer with a smaller distance than in front of it. The optimization variables σ_s and σ_f are the standard deviations of the normal distribution corresponding to the lengths of the semi-axes of the ellipsoid. Consequently, the cost function reads:

$$f_{inter}(\overline{x}|\sigma_f, \sigma_s, \delta_I) := \int_{t_0}^{t_f} \exp\left(-\frac{1}{2}\left(\left(\frac{l_f(t)}{\sigma_f}\right)^2 + \left(\frac{l_s(t)}{\sigma_s}\right)^2\right)\right) \, \mathrm{d}t,$$

where $l_f(t)$ and $l_s(t)$ are the distance between the center $\overline{\mathcal{C}}(t)$ and the position of the participant $\mathcal{P}_I(t)$ in forward and in sideward direction, accordingly.

8.3 Human Experiments

The analysis of human navigation in the presence of interfering persons is the goal of a cooperation with the research group of Glasauer, institute for clinical neurosciences, Ludwig-Maximilians-Universität München, and the research group of Hermsdörfer, faculty of sport and health sciences, Technische Universität München. The human locomotion data was recorded at the laboratory of Hermsdörfer using a high-precision tracker based on markers.

In general, the analyzed motion tasks can be stated in the following way: Walk from a start to a goal position where both positions are specified on the floor. No limitations on the overall motion time are given, therefore it can be assumed that the participants used a comfortable movement time, i.e., a comfortable walking speed. Several different scenarios of such motion tasks are part of the recorded human data, i.e., tasks leading to straight or curved motions. Especially, collision avoidance scenarios where an interfering person crosses the paths of the participants are of interest, since such problems occur regularly in daily life. The interfering person is instructed to walk along a virtual straight line at a constant speed and avoid any interaction with the participant. Consequently, the participant has to react in order to avoid a collision.

8.3.1 Distance Measures

For the data matching problem a distance measure Φ has to be introduced that is suitable for the task at hand (see section 4.3). Ideally one would like to match both the Cartesian path and the velocity profile, thus a first choice would be to compare the position of the participant $\mathcal{P}_P(t_i)$ with the position $\mathcal{P}_D(t_i)$ of the recorded human data for given time instances $t_i \in [t_0, t_f]$, $i = 1, \ldots, \overline{\nu}$ by using Φ_{time} (cf. section 4.3).

However, the velocity profiles recorded in the experiments exhibit considerable oscillations corresponding to the individual steps of the participants. Since individual steps are not modeled in the introduced plant model, a data fitting with respect to this raw velocity data is not fully appropriate. Consequently, the velocity data is smoothed to obtain the mean velocity $\tilde{v}_D(t)$ of the center of mass.

The distance measure used in our computations is a combination of two measures; the first one considers the differences between the velocity profiles by using Φ_{time}. The cost function Φ_{path} based on path-length is selected as the second one to compute the differences between the Cartesian paths. This decoupling of positional and temporal information proves to be a

suitable way to assure that the Cartesian path of the solution of the optimal control problem is compared to the originally recorded path of the human participant, and at the same time to consider only a smoothed velocity profile suitable for the simple dynamical model.

8.4 Model Predictive Control

The experimental data suggests that in the case of linearly moving obstacles humans do not optimally plan their overall motion from the start to the goal, but rather stick to a control strategy ignoring the obstacle as long as the distance is large enough. Not until the distance gets small, a maneuver of collision avoidance is started. Since it is hard to distinguish whether the distance of the strategy switching is determined by a temporal or a spatial measure, we introduce in the following a framework combining both kinds of measures.

The idea to repeatedly solve optimal control problems during a motion in order to react to changes in the environment is best captured in a model predictive control framework. The basic idea of model predictive control is to sequentially solve optimal control problems for the task to move from the current position to the goal position, but only realize the computed controls over a limited horizon and then start over with solving the next optimal control problem with the new position as the starting position. In consequence, the overall motion, which has to be compared to given data in the upper level of the inverse optimal control problem, is a combination of segments obtained from a series of optimal control problems, i.e., the inversion has to be done with respect to all submotions at once.

Note that the switching structure observed in human navigation asks for two combinations of basic cost functions f_i, thus the notation introduced in section 6.4.4 has to be extended here. Consider k^I and $k^{II} \in \mathbb{N}$ to be the numbers of basic cost functions used in the combinations:

$$\begin{aligned} f^I(\overline{x}(t), \overline{u}(t) \mid \pi^I) &:= \sum_{i=1}^{k^I} \overline{w}_i f_i(\overline{x}(t), \overline{u}(t) \mid \pi_i), \\ f^{II}(\overline{x}(t), \overline{u}(t) \mid \pi^{II}) &:= \sum_{i=k^{II}+1}^{k^{II}+k^I} \overline{w}_i f_i(\overline{x}(t), \overline{u}(t) \mid \pi_i), \end{aligned}$$

where the parameter vectors π^I and π^{II} are defined by

$$\begin{aligned} \pi^I &:= \left(\overline{w}^I_1, \ldots, \overline{w}^I_{k^I}, \left(\pi^I_1\right)^T, \ldots, \left(\pi^I_{k^I}\right)^T\right)^T \in \mathbb{R}^{\overline{s}^I}, \\ \pi^{II} &:= \left(\overline{w}^{II}_{k^I+1}, \ldots, \overline{w}^{II}_{k^I+k^{II}}, \left(\pi^{II}_{k^I+1}\right)^T, \ldots, \left(\pi^{II}_{k^I+k^{II}}\right)^T\right)^T \in \mathbb{R}^{\overline{s}^{II}}. \end{aligned}$$

Note that the dimensions $\overline{s}^I$ and $\overline{s}^{II} \in \mathbb{N}$ result from

$$\overline{s}^I := k^I + \sum_{i=1}^{k^I} \overline{s}_i \quad \text{and} \quad \overline{s}^{II} := k^{II} + \sum_{i=k^I+1}^{k^{II}+k^I} \overline{s}_i,$$

where $\overline{s}_i \in \mathbb{N}$ is the number of parameters for the respective basic cost function combined in the vector $\pi_i \in \mathbb{R}^{\overline{s}_i}$. The idea of convex combinations of basic cost functions used in the case of the classical inverse optimal control problems can be generalized to two combinations of

cost functions if a relative scaling factor $\overline{\varpi} > 0$ is introduced. This results in the following conditions for the scaling factors of the basic cost functions:

$$\sum_{i=1}^{k^I} \overline{w}_i = \overline{\varpi} \quad \text{and} \quad \sum_{i=k^I+1}^{k^I+k^{II}} \overline{w}_i = 1.$$

Consequently, the vector of upper level parameters y is given by

$$y := \left(\pi^I, \pi^{II}, \overline{\varpi}\right)^T \in \mathbb{R}^{\bar{s}^I+\bar{s}^{II}+1}.$$

Having defined the two combinations of basic cost functions, the switching process between the two has to be stated:

$$f(\overline{x}(t), \overline{u}(t) \mid y) = \begin{cases} f^I(\overline{x}(t), \overline{u}(t) \mid \pi^I) & \text{if} \quad t \leq \bar{t} \ \text{ and } \ ||\mathcal{P}_P(t) - \mathcal{P}_I(t)|| \leq d, \\ f^{II}(\overline{x}(t), \overline{u}(t) \mid \pi^{II}) & \text{else,} \end{cases}$$

where $\bar{t} > 0$ is the time horizon where the linear approximation of the interferer position has to be considered and $d > 0$ is the maximal distance where the interferer influences the locomotion path of the participant. Note that hard switching between the two cost functions will cause non-differentiability in the optimal control problem; to avoid problems that might arise in the numerical optimization, a smooth transition between the two cases is recommended.

Remark 8.4.1. The implicit assumption is that the interferer cost f_{inter} is part of f^I but not part of f^{II}. In this case the time condition $t \leq \bar{t}$ leads to a reduced cost function for the second part of the motion which can be interpreted as a terminal cost term common in model predictive control. Such a terminal cost term should assure that the final position of one motion segment does not lead to tremendous costs in the next one.

8.5 Optimization Results

The following examples focus on various aspects arising in the context of human locomotion. One aspect addressed in the first example is the problem of finding a common cost function for several motions which means that the inversion has to be done with respect to more than one lower level problem. Additionally, some of the bounds imposed on the states and controls of the unicycle model become active and the numerical results show that the resulting MPEC structure can be handled. Furthermore, we address the problem of collision avoidance and present reconstruction results for the model predictive control framework.

8.5.1 Reconstruction of Multiple Locomotions

The stereotypical locomotion tasks are combined in this example: a U-turn, a turn to the left and a switching to a path parallel to the starting direction. Note that in all cases the starting and end velocities are prescribed to have the value 0.5 and especially the bounds on the velocity

$$0.2 \leq v_P(t) \leq 0.7$$

are of interest, because they are designed to become active for some of the stereotypical motions in this example.

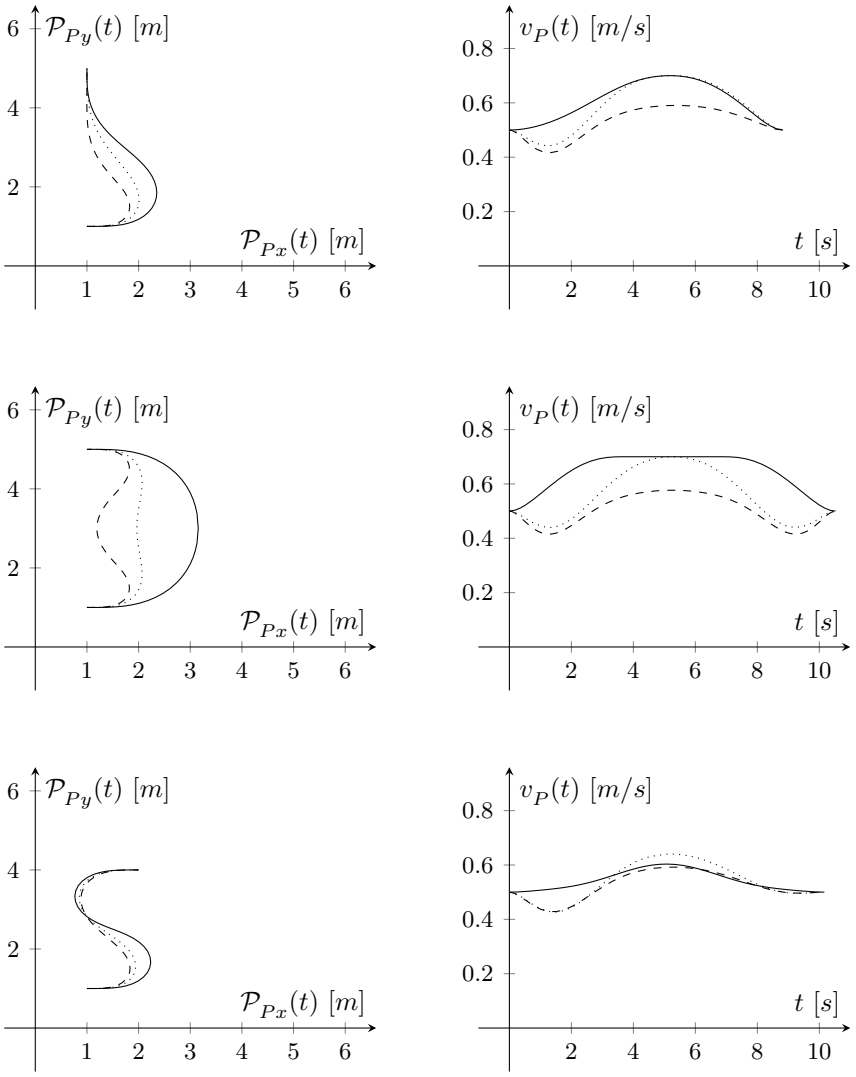


Figure 8.3: The locomotion paths and the corresponding velocity profiles for three stereotypic motion tasks analyzed in the reconstruction framework. [——: generated data, - - -: 1st starting value, ·····: 2nd starting value]

All three motions have a common starting position where the state $\overline{x}$ has to fulfill:

$$\overline{x}(t_0) = (1, 1, 0, 0.5, 0, 0, 0)^T .$$

The prescribed end configurations at the free final time t_f for the U-turn are

$$\overline{x}(t_f) = (1, 5, \pi, 0.5, 0, 0, 0)^T ,$$

for the left turn

$$\overline{x}(t_f) = (1, 5, 0.5\pi, 0.5, 0, 0, 0)^T ,$$

and for the parallel path motion

$$\overline{x}(t_f) = (2, 4, 0, 0.5, 0, 0, 0)^T .$$

Five basic cost functions f_i are selected for this example using the reference value $\underline{r}_4 = 0.8$ for the cost function $f_{ref,4}$:

i	1	2	3	4	5
f_i	$f_{u,1} + f_{u,2}$	$f_{ref,4}$	f_{time}	f_{line}	f_{jerk}

The figure (8.3) shows selected optimal states and controls for the weight vector

$$\overline{w} = (0.3, 0.4, 0.2, 0, 0.1)^T ,$$

which is chosen for the data construction within the reconstruction framework, and the two vectors

$$(0.1, 0.05, 0.3, 0.2, 0.35)^T \quad \text{and} \quad (0.2, 0.15, 0.35, 0.1, 0.2)^T$$

are used to generate suitable starting values for the inversion by solving the corresponding optimal control problems. Using the combination of the upper level cost functions comparing the Cartesian paths by path length and the velocity profiles with respect to time, where in both cases 50 points are used for the cost computation, the following reconstruction results are obtained for the case where a white noise of magnitude 10^{-2} time the range of the corresponding variable is added to the computed data. The table (8.1) states the differences in the weight distributions between the reconstruction results and the values used for the data generation.

difference in $\overline{w}_i$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
1st starting value	$4.59\ 10^{-4}$	$1.22\ 10^{-3}$	$1.08\ 10^{-3}$	0	$3.23\ 10^{-4}$
2nd starting value	$6.55\ 10^{-4}$	$1.55\ 10^{-3}$	$1.42\ 10^{-3}$	0	$5.29\ 10^{-4}$

Table 8.1: Differences in weights distributions between reconstruction results for two starting values and the values used to generate the data.

Note that for both starting values the interior point approach is able to determine that the fourth basic cost function is not part of the combined cost function. The obtained optimal values of the upper level cost functions are of the magnitude 10^{-5} which is small compared to the scales of the considered states, however, the accuracy of the weights as stated in table (8.1) are more distinct.

8.5.2 Inversion of U-Turn Motions

In this section we discuss the inversion of two U-turn motions recorded in human experiments (compare section 8.3). Both motions have certain characteristics in common, for example the choice of the maximal distance from the baseline connecting start and goal position or the non-symmetrical drop in the velocity profile. Using the same parameterized family of feasible lower level cost functions as in the previous section, the reference velocity of the second basic cost function is considered here to be an upper level variable in addition to the weights of the five basic cost functions. The following plots show the recorded velocity profiles and the Cartesian paths for the selected motions. Note that this presentation is in accordance with the selected combination of the two distance measures as discussed above.

The following values are obtained for the weights and the nonlinear parameter by using the inverse optimal control approach:

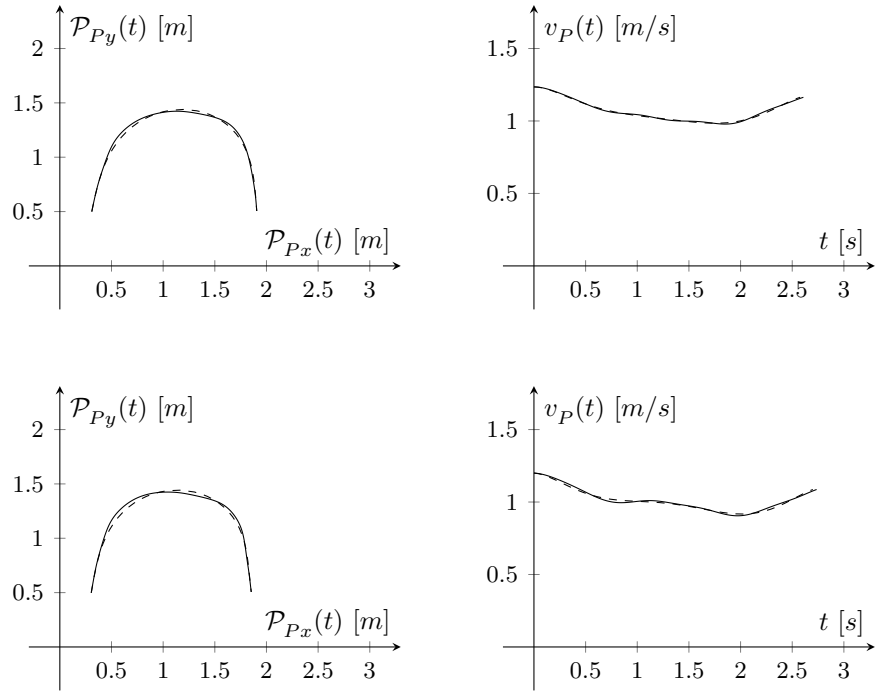


Figure 8.4: The locomotion paths and the corresponding velocity profiles for two recorded human motions and the solution of the inversion optimal control approach. [——: recorded data, - - -: solution of inversion]

$\overline{w}_i$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$\underline{r}_4$
1st U-turn	$7.09\ 10^{-4}$	$5.91\ 10^{-1}$	$1.93\ 10^{-7}$	$3.87\ 10^{-1}$	$2.20\ 10^{-2}$	$7.46\ 10^{-1}$
2nd U-turn	$7.12\ 10^{-4}$	$6.42\ 10^{-1}$	$1.25\ 10^{-7}$	$3.44\ 10^{-1}$	$1.35\ 10^{-2}$	$8.02\ 10^{-1}$

Table 8.2: Optimization results of the upper level parameters for two recorded human U-turn motions.

The common characteristics observed in the recorded data here result in similar results for the weight distributions, in both cases the cost functions considering the deviation of the velocity from a reference value and the deviation of the path from a straight line connecting start and goal position are dominant. The additional cost terms assure that the trade-off between velocity and distance does not result in jerky motions or too large control inputs. The obtained values for the nonlinear parameter $\underline{r}_4 \in [0.4, 1.6]$ are close to each other and the value of 0.8 used in section 8.5.1 seems to be a reasonable approximation.

8.5.3 Reconstruction of MPC Navigation

The model predictive control idea for human navigation problems as introduced in section 8.4 is used here for the reconstruction of a collision avoidance maneuver where a linearly moving interferer crosses the path. The overall motion task corresponds to one of the three motions considered in section 8.5.1. The following values are chosen for the constants describing the

switching structure of the MPC approach: $\bar{t} = 3$ $[s]$ and $d = 2$ $[m]$, which in this example results in three MPC segments. The constructed data trajectory (cf. figure (8.5)) exhibits certain characteristics which have been observed in the human experiments; for example, a slight deviation from the path that would have been used if no interferer had crossed, and a considerable change in the velocity profile. The deceleration assures that no collision happens and the goal position is reached after passing behind the interferer. The starting position of the interferer is $(1, 3)^T$ and the position after 8 seconds is $(2, 3)^T$, i.e., the original trajectory would result in a collision.

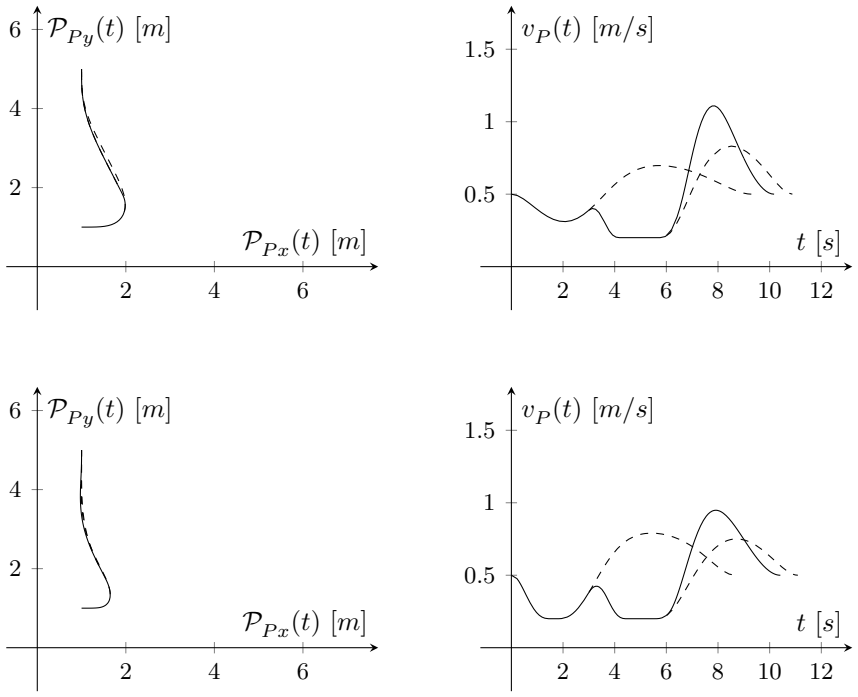


Figure 8.5: The locomotion paths and the corresponding velocity profiles for MPC structured motion analyzed in the reconstruction framework (first row) and the corresponding starting value (second row). [- - -: preplanned motion segments, ——: MPC motion]

The following table states the weight distributions used to generate the data and the starting values of the reconstruction run. Additionally, the differences between the numerical result of the inversion of the data with added noise and the original values are given. Note that $\overline{w}_i$ for $i = 6$ is the weighting factor for the interferer cost f_{inter} and that the weights with index $i = 7, \ldots, 11$ correspond to the second cost combination using the same basic cost functions in the identical order.

Note that this reconstruction example has certain characteristics in order to focus on the adaptation of the motion based on the interferer's positions. Having one cost function for the free motion and one for the collision avoidance motion would allow to use two totally different cost functions here, but this means that the trajectory is even adapted if a static interferer has a considerable distance to the planned trajectories, i.e., solely by advancing the planning horizon the next motion segment would have to be planned with the collision avoidance cost rather than the free motion cost. To avoid a mixture of these effects for presentation reasons, we here only added the interferer cost function to the free motion cost

combination. Furthermore, note the sum of the weights $\overline{w}_i$ with index $i = 1, \ldots, 6$ equals 6 in this example. Since the relative scaling of the two combinations of basic cost functions is an optimization variable, the reconstruction result shows an error of $1.82\ 10^{-2}$ and consequently, the difference of the weight distributions of the reconstruction run and the data has to be of a similar scale. However, the obtained reconstruction result for the upper level cost function being identical to the one of section 8.5.1, where similarly three motions are compared, is smaller than 10^{-4}.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
data value $\overline{w}_i$	0.15	0.7	0.15	0	0	5
starting value $\overline{w}_i$	0.15	0.3	0.15	0.4	0	5
difference in $\overline{w}_i$	$2.71\ 10^{-4}$	$3.59\ 10^{-3}$	$1.64\ 10^{-2}$	$2.62\ 10^{-3}$	0	$7.72\ 10^{-3}$

	$i = 7$	$i = 8$	$i = 9$	$i = 10$	$i = 11$
data value $\overline{w}_i$	0.15	0.7	0.15	0	0
starting value $\overline{w}_i$	0.15	0.3	0.15	0.4	0
difference in $\overline{w}_i$	$1.17\ 10^{-3}$	$4.74\ 10^{-3}$	$5.83\ 10^{-3}$	0	$1.80\ 10^{-4}$

Table 8.3: Weight distributions for data and starting value and the difference between the reconstruction result and the data values.

Summary

Inverse optimal control problems are addressed in this work, because they are a promising approach to obtain suitable optimal control models for human motions in everyday life scenarios. Using this bilevel framework, it is possible to quantify that a specific model is the best one within a given parameterized family of models, as opposed to the qualitative comparisons used in many works on human motions. In addition to the mathematical theories used in our solution strategy, a major part of this work addresses three application examples. In all three cases, details on modeling the dynamics of the according human plant are given and the current state of the art is reviewed. The central aspects are to derive a dynamical model that is a reasonable compromise between complexity and efficiency, i.e., the model has to be detailed enough to describe the main aspects of the dynamics of the plant and at the same time its outputs have to be obtainable at a relatively low computational cost, and to deduce a suitable family of lower level cost functions that allows to capture the central characteristics of the human motions but avoids over-fitting. The different scenarios are used to highlight different aspects in inverse optimal control ranging from highly nonlinear plants over nonlinear parameters in the upper level and multiple lower level problems to inequality constraints in the optimal control problem. Numerical results of the inverse optimal control problems for recorded experimental data are presented and the usage of such optimization results in order to control robotic systems is discussed.

Since inverse optimal control problems have a rather complicated structure, several mathematical theories have to be combined to solve them. The strategy used in this work is based on discretizing the optimal control problem by a collocation method and to reformulate the bilevel program to a one-level problem by replacing the discretized optimal control problem by its KKT-conditions. The resulting MPEC is then solved by applying a relaxation scheme and the resulting sequence of standard nonlinear optimization problems is solved by using an interior-point method. In consequence, the theoretical part of this work starts with the introduction of standard theory on bilevel optimization problems and the existence of a global (optimistic) solution is proven under suitable assumptions. Because the solution strategy chosen in this work to solve the bilevel problems yields a mathematical program with complementarity constraints, an introduction to MPECs with their specific stationarity concepts follows. Related relaxation schemes introduced to solve MPECs numerically are reviewed and a variant usable in the context of interior point methods is discussed. Furthermore, theory on optimal control is reviewed and, especially, collocation methods are discussed in order to discretize the LLP of the inverse optimal control problem. Suitable convergence results of the discretized problems towards the original problems are stated and a proof of the existence of an solution for the continuous optimal problem is given using certain assumptions. Then, the structures of the optimization problems resulting from our solution strategy are analyzed with respect to the existence of a global optimistic solution for the resulting bilevel prob-

lems and the guarantee of a CQ for the transformed one-level problem. Finally, numerical strategies for nonlinear optimization problems are addressed with the focus on interior point methods and some implementation details of our optimization method are presented.

Summing up, a numerical method for solving inverse optimal control problems is developed in this work and numerical experiences for several problems are discussed. Bilevel optimal control problems are a relatively new problem class which seems to have many application in engineering sciences. Consequently, further research on this problem class is needed to improve the current theoretical framework and the numerical methods for solving these problems. The following directions seem to be promising for future research: First, approaches of robust optimization might improve the applicability of inverse optimal control results in technical systems with respective measurement errors and uncertainties in the boundary conditions. Second, a test-set of reference bilevel optimal control problems would be useful to verify the results and classify the performance of new numerical procedures. Third, considering the optimality conditions for bilevel optimal control problems, an optimization approach suitable for non-smooth problems might be an efficient way of using information on the full problem structure.

Nonlinear Optimization

Appendix A

The basic problem of *nonlinear optimization* is to find an optimum for a finite dimensional optimization problem:

$$\min_x \phi(x) \quad \text{subject to} \quad x \in \mathbb{X},$$

where the non-empty *feasible set* $\mathbb{X} \subset \mathbb{R}^n$ and the continuous generally nonlinear *objective function* $\phi : \mathbb{X} \to \mathbb{R}$ are given. In this chapter we will define the specific nonlinear optimization problems we are interested in and discuss available theory to analyze them. Finally, selected numerical methods for solving such problems are discussed.

The content of this chapter can be found in greater detail in many textbooks on nonlinear optimization. For the presentation in this work we follow the line of [312] and, accordingly, [122, 230].

Throughout this work most problems of nonlinear optimization are described by at least twice continuously differentiable functions; thus the following definition of a nonlinear optimization problem is made:

Definition A.0.1. ***(Constrained Nonlinear Optimization Problem)***
In a ***constrained nonlinear optimization problem*** *a nonlinear objective function* $\phi : \mathbb{X} \to \mathbb{R}$ *is minimized over a non-empty feasible set* $\mathbb{X} \neq \mathbb{R}^n$ *which is described by the two functions* $h : \mathbb{X} \to \mathbb{R}^p$ *and* $g : \mathbb{X} \to \mathbb{R}^q$*:*

$$\min_x \phi(x) \quad \textit{subject to} \quad h(x) = 0,\ g(x) \leq 0.$$

All functions, i.e., the cost function ϕ*, the equality constraints* h *and the inequality constraints* g*, are assumed to be twice continuously differentiable.*

Remark A.0.2. The above definition leads to the following feasible set:

$$\mathbb{X} = \{x \in \mathbb{R}^n \mid g(x) \leq 0,\ h(x) = 0\}.$$

A point $x \in \mathbb{R}^n$ is called *feasible* for problem A.0.1 if $x \in \mathbb{X}$.

Definition A.0.3.

a) A point $x^* \in \mathbb{R}^n$ *is called a* ***local minimum*** *of problem A.0.1 if* x^* *is feasible and a scalar* $\varepsilon > 0$ *exists such that*

$$\phi(x) \geq \phi(x^*) \quad \forall x \in \mathbb{X} \cap \mathbb{U}_\varepsilon(x^*),$$

where the set $\mathbb{U}_\varepsilon(x^*) := \{x \in \mathbb{R}^n \mid ||x - x^*|| < \varepsilon\}$ *is the ball around* x^* *with radius* ε.

b) A point $x^* \in \mathbb{R}^n$ *is called a* ***strict local minimum*** *of problem A.0.1 if* x^* *is feasible and a* $\varepsilon > 0$ *exists such that*

$$\phi(x) > \phi(x^*) \quad \forall x \in (\mathbb{X} \cap \mathbb{U}_\varepsilon(x^*)) \setminus \{x^*\}.$$

c) A point $x^* \in \mathbb{R}^n$ *is called a* ***global minimum*** *of problem A.0.1 if* x^* *is feasible and*

$$\phi(x) \geq \phi(x^*) \quad \forall x \in \mathbb{X}.$$

c) A point $x^* \in \mathbb{R}^n$ *is called a* ***strict global minimum*** *of problem A.0.1 if* x^* *is feasible and*

$$\phi(x) > \phi(x^*) \quad \forall x \in \mathbb{X} \backslash \{x^*\}.$$

A.1 First-Order Optimality Conditions

In order to discuss the first-order optimality conditions of nonlinear optimization, known as *KKT-conditions*, two characteristic cones have to be defined and the concept of *constraint qualification* has to be used.

We start with the introduction of the *tangent cone*, which at a given point combines all directions that can be generated by sequences of feasible points:

Definition A.1.1. ***(Tangent Cone)***
For a non-empty set $\mathbb{M} \subset \mathbb{R}^n$ *the tangent cone of* $\mathbb{M}$ *at a point* $x \in \mathbb{M}$ *is given by the following set:*

$$\mathbb{T}(\mathbb{M}, x) := \left\{ d \in \mathbb{R}^n \mid \exists \eta^{(k)} > 0,\ x^{(k)} \in \mathbb{M} : \lim_{k\to\infty} x^{(k)} = x,\ \lim_{k\to\infty} \eta^{(k)} \left(x^{(k)} - x \right) = d \right\}.$$

A first necessary condition for a local minimum results directly:

Theorem A.1.2. ***(Necessary Condition)***
Let $x^* \in \mathbb{R}^n$ *be a local solution of problem A.0.1, then the following statements hold:*

a) $x^* \in \mathbb{X}$,

b) $\nabla\phi(x^*)^T d \geq 0, \quad \forall d \in \mathbb{T}(\mathbb{X}, x^*)$.

Proof. Follows directly by applying Taylor's theorem, see [122]. □

Since checking whether a vector is an element of the tangent cone is usually not realizable in a general setting, a second cone using the linearized versions of the constraints is defined:

Definition A.1.3. ***(Linearized Tangent Cone)***
Using the ***set of active inequality constraints*** $\mathbb{A}(x) := \{i \in \mathbb{N} \mid i \leq q,\ g_i(x) = 0\}$, *the* ***linearized tangent cone*** *is given by*

$$\mathbb{T}^{lin}(g, h, x) := \{d \in \mathbb{R}^n \mid \nabla h(x)^T d = 0,\ \nabla g_i(x)^T d \leq 0,\ i \in \mathbb{A}(x)\}.$$

The linearized tangent cone is convex as a direct consequence of the definition:

$$\alpha d^{(1)} + (1-\alpha) d^{(2)} \in \mathbb{T}^{lin}(g, h, x)$$

for $\alpha \in [0, 1]$ if $d^{(1)}$ and $d^{(2)} \in \mathbb{T}^{lin}(g, h, x)$.

Remark A.1.4. The tangent cone is a subset of the linearized one:

$$\mathbb{T}(\mathbb{X}, x) \subset \mathbb{T}^{lin}(g, h, x).$$

Note that the tangent cone is uniquely defined by the feasible set $\mathbb{X}$ and the evaluation point x. In contrast, the definition of the linearized tangent cone depends on the actual definitions of the constraint functions. This means that a variation of the functions h and g describing the feasible set can result in a different linearized tangent cone.

Since the other inclusion of the remark A.1.4 is not true in general, the concept of a *constraint qualification (CQ)* is introduced. The most general CQ is the *Guignard constraint qualification (GCQ)* [128, 134], but for the purposes of this work it is sufficient to start with the *Abadie constraint qualification (ACQ):*

Definition A.1.5. ***(ACQ)***
The constraint

$$\mathbb{T}^{lin}(g, h, x) = \mathbb{T}(\mathbb{X}, x)$$

is called ***Abadie Constraint Qualification (ACQ)****. Every other constraint that implies the ACQ will be named a* ***constraint qualification (CQ)****.*

The theorem A.1.2 in combination with the definition of the ACQ yields the following necessary condition for a local minimum of the nonlinear optimization problem:

Corollary A.1.6.
Let x^* be a local minimum of A.0.1 and let the ACQ hold for x^*. Then follows:

a) $x^* \in \mathbb{X}$,

b) $\nabla\phi(x^*)^T d \geq 0, \quad \forall d \in \mathbb{T}^{lin}(g, h, x^*)$.

This corollary is the basis for the *Karush-Kuhn-Tucker conditions (KKT-conditions)*, the necessary first-order optimality conditions:

Theorem A.1.7. ***(KKT-conditions)***
Let x^* be a local solution of problem A.0.1 where a CQ is fulfilled, then there exist ***Lagrange multipliers*** *$\mu^* \in \mathbb{R}^p$ and $\lambda^* \in \mathbb{R}^q$ such that*

a) $\nabla\phi(x^*) + \nabla g(x^*)\lambda^* + \nabla h(x^*)\mu^* = 0$,

b) $h(x^*) = 0$,

c) $\lambda^* \geq 0, \quad g(x^*) \leq 0, \quad (\lambda^*)^T g(x^*) = 0.$

Part (a) is called ***stationarity condition*** *or* ***multiplier condition*** *and part (c) is known as the* ***complementarity condition****.*

Proof. This theorem can be proven by combining the corollary A.1.6 with the necessary optimality conditions of linear optimization theory; see for example [122]. □

Remark A.1.8. The following two conditions are equivalent to the complementarity conditions of theorem A.1.7:

c') $\lambda_i^* \geq 0, \quad g_i(x^*) \leq 0, \quad \lambda_i^* g_i(x^*) = 0,\ i = 1, \ldots, q.$

c") $g(x^*) \leq 0, \quad \lambda_i^* \geq 0 \text{ for } i \in \mathbb{A}(x^*), \quad \lambda_i^* = 0 \text{ for } i \notin \mathbb{A}(x^*).$

A point fulfills *strict complementarity* if for all $i \in \mathbb{A}(x^*)$ follows that $\lambda_i^*(x^*) > 0$.

We now introduce the *Lagrangian* which can be used to shorten the notation of the stationarity condition and will be useful later on. The Lagrangian is a function returning a scalar value that is the sum of the cost function and the weighted constraints:

Definition A.1.9. ***(Lagrangian)***
For problem A.0.1 the Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^p \to \mathbb{R}$ *is defined by*

$$L(x, \lambda, \mu) := \phi(x) + \lambda^T g(x) + \mu^T h(x).$$

Corollary A.1.10.
The stationarity condition of theorem A.1.7 can be written as $\nabla_x L(x, \lambda, \mu) = 0$.

The following constraint qualifications are most common in literature; they are easier to check than the ACQ, but they imply the ACQ. The first one is the *Mangasarian-Fromovitz constraint qualification* ensuring the existence of a direction along which all linearized active inequality constraints strictly decrease and all equality constraints hold.

Definition A.1.11. ***(MFCQ)***
The ***Mangasarian-Fromovitz constraint qualification (MFCQ)*** *is fulfilled at a point* $x \in \mathbb{X}$ *if*

a) $\nabla h(x)$ *has full column-rank or* h *is affine linear,*

b) $\exists d \in \mathbb{R}^n : \quad \nabla g_i(x)^T d < 0,\ i \in \mathbb{A}(x), \quad \nabla h(x)^T d = 0.$

Condition (b) can be omitted if $q = 0$ *or* $\mathbb{A}(x) = 0$*; the case* $p = 0$ *can be interpreted as* $h \equiv 0$.

Remark A.1.12. To justify that the MFCQ is called a constraint qualification one has to show that the condition implies the ACQ. The proof of this implication can be found, for example, in [122].

A second CQ called the *positive linear independence constraint qualification* guarantees that the columns of the Jacobians of the equality and active inequality constraints are positive linearly independent, which means that the scalar factors of the columns corresponding to the active inequality constraints are non-negative and at least one of these factors is positive:

Definition A.1.13. ***(PLICQ)***
The ***positive linear independence constraint qualification (PLICQ)*** *is fulfilled at a point* $x \in \mathbb{X}$ *if*

a) $\nabla h(x)$ *has full column-rank or* h *is affine linear,*

b) No vectors $\lambda \in \mathbb{R}^q$ and $\mu \in \mathbb{R}^p$ exist such that:

$$\nabla g(x)\lambda + \nabla h(x)\mu = 0,$$
$$\lambda_{\mathbb{A}(x)} \geq 0,\ \lambda_{\mathbb{A}(x)} \neq 0,\ \lambda_{\mathbb{I}(x)} = 0.$$

Condition (b) can be omitted if $q = 0$ or $\mathbb{A}(x) = 0$; the case $p = 0$ can be interpreted as $h \equiv 0$.

Remark A.1.14. The PLICQ is fulfilled at a point $x \in \mathbb{X}$ if and only if the MFCQ holds; consequently, the PLICQ is a constraint qualification. For a proof of the equivalence of the PLICQ and the MFCQ we refer to [312].

In many cases a stronger constraint qualification is used that ensures that the columns of the Jacobians of the equality and the active inequality constraints are linearly independent:

Definition A.1.15. ***(LICQ)***
A point $x \in \mathbb{X}$ fulfills the ***linear independence constraint qualification (LICQ)*** *if the columns of the matrix $\big((\nabla g(x))_{\mathbb{A}(x)}, \nabla h(x)\big)$ are linearly independent.*

Remark A.1.16. The LICQ is a constraint qualification since it implies the PLICQ, but the reverse implication does not hold. Therefore, the LICQ is called *stronger* than the PLICQ.

A.2 Second-Order Optimality Conditions

The second-order optimality conditions discussed in this section are naturally based on the assumption that all functions describing the nonlinear optimization problem A.0.1 are twice continuously differentiable. In order to state second-order necessary and sufficient conditions the following cone combining the relevant directions is introduced:

Definition A.2.1.
Define the ***cone*** $\mathbb{T}^+$ *for $x \in \mathbb{X}$ and $\lambda \in [0, \infty)^q$ by*

$$\begin{aligned}\mathbb{T}^+(g,h,x,\lambda) := \big\{ d \in \mathbb{R}^n \mid \nabla g_i(x)^T d &= 0, \quad \text{if } i \in \mathbb{A}(x) \text{ and } \lambda_i > 0,\\ \nabla g_i(x)^T d &\leq 0, \quad \text{if } i \in \mathbb{A}(x) \text{ and } \lambda_i = 0,\\ \nabla h(x)^T d &= 0 \,\big\}.\end{aligned}$$

Using this cone in combination with the LICQ assumption assuring that the Lagrange multipliers in the KKT conditions are unique for a given primary variable yields the following necessary conditions of second order:

Theorem A.2.2. ***(Second-Order Necessary Conditions)***
Let ϕ, g and h be twice continuously differentiable and let x^* be a local minimum of problem A.0.1 where the LICQ is fulfilled. Then there exist (unique) Lagrange multipliers $\lambda^* \in \mathbb{R}^q$ and $\mu^* \in \mathbb{R}^p$ fulfilling parts (a) to (c) of the KKT-conditions A.1.7 and the following condition holds true:

$$d^T \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) d \geq 0 \quad \forall d \in \mathbb{T}^+(g,h,x^*,\lambda^*).$$

Proof. See [122]. □

Similar to the case of unconstrained optimization, where a sufficient condition is obtained if the Hessian of the cost function is not only positive semi-definite but positive definite,

second-order sufficient conditions are obtained in the case of constraints if the Hessian of the Lagrangian is positive definite for all non-zero vectors in $\mathbb{T}^+$:

Theorem A.2.3. ***(Second-Order Sufficient Conditions (SOSC))***
Let $x^* \in \mathbb{R}^n$ in combination with the Lagrange multipliers $\lambda^* \in \mathbb{R}^q$ and $\mu^* \in \mathbb{R}^p$ fulfill the parts (a) to (c) of the KKT-conditions A.1.7. Additionally, assume that

$$d^T \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) d > 0 \quad \forall d \in \mathbb{T}^+(g, h, x^*, \lambda^*) \setminus \{0\}.$$

Then x^* is a strict local minimum of problem A.0.1.

Proof. This is also shown in [122]. □

State of the Art on Human Arm Motions

Appendix B

In literature the research on human motions or human-like motions has various facets and the amount of publications in this field is fast-growing. Disciplines dealing with analyzing and describing these movements range from psychology and biology over computer and engineering sciences to mathematics. Therefore this section can only give an overview of main aspects and state the connection to the research of this work. If the reader is interested in details of specific aspects, various references are named as starting points for further research.

We will start the discussion of the state of the art with describing the goals of analyzing human motions (section B.1) and state characteristics of human arm motions for the different disciplines (section B.2). This is followed by the discussion of the basic principles in human motor control (section B.3), e.g., introduction of forward and inverse models, and the different hypotheses on the underlying principles of human motion generation (section B.4). Finally, in sections B.5 and B.6, the open-loop and closed-loop frameworks and human adaptation characteristics are introduced and a brief outlook to learning frameworks is given. Details on open-loop cost functions and on transfer of motions to robotic systems, both relevant aspects for this work, will be discussed in more detail in the sections 6.4 and 6.7.

B.1 Research Goals of Disciplines

For different disciplines the basic goals of analyzing human motion differ and therefore the results have to be viewed in the corresponding setting. In most cases psychologists and biologists try to deduce the underlying structure and principles of human motions with their experiments. The final goal is to understand the mechanisms determining the human actions. On the other hand mathematicians and physicists attempt to build a model describing the observed movements. The main purpose of the model is to describe behavior rather than to explain it [89]. Hence, the minimum principles we obtain by the bilevel optimization approach have to be understood as a description of the human motion and there might be various arguments why the optimal cost function describing the human movements is not biologically or psychologically plausible.

Another general distinction between diverse approaches in the field of human motions is the usage of a model. In many cases discussed in the following models are derived, but "these theoretical constructs should not be identified with the phenomenon they attempt to explain" [235]. In some cases the same phenomena can be described by model-free theories, e.g., [101, 211]. Consequently, we have to keep in mind that our model-based approach might not be the only way to obtain good approximations of human motion and as Hogan and Flash summarized: "Theories are not immutable truths, but mental constructions that must evolve

to accommodate new data" [159]. Most models describing human motions are based on the idea that human arm motions minimize an unknown cost function. Naturally, the question arises which one of the proposed cost functions describes human motions best? "Ideally, the cost assumed in an optimal control model should correspond to what the sensorimotor system is trying to achieve" [302]. Bilevel optimal control is a tool to find the cost function out of a family of cost functions that minimizes the distance between the respective simulated results and recorded human data. From a biological perspective, as [89] pointed out, this approach might lead into circular reasoning if the obtained cost function is viewed as an indicator that human motions are in some way optimal. Consequently, an obtained minimization criterion should be seen as "a purely descriptive tool that concisely summarizes a set of experimental data" [89]. Furthermore, a minimization principle obtained by the bilevel approach "is by itself not a significant achievement" [89], since simple functional relations describing human arm motions exist, e.g., see [249]. Therefore a minimization criterion has only a real value in the view of biology if it can predict different human behavior in different settings.

B.2 Motion Characteristics

We will start the discussion of the motion characteristics in human arm motions by stating general observations on the shapes of trajectories and velocity profiles. This is followed by the summary of results on variance in human motor generation and the resulting fundamental relations of human motions, for example Fitts' law.

First observations of stereotypical characteristics of human arm movements are presented by [2] and [218]. Two-dimensional arm motions recorded by a planar handle show common features like a single peaked shape of the tangential velocity profile and the shape of the hand trajectory. As a consequence Morasso [218] hypothesizes that "the central commands which underlie the observed movements are more likely to specify the trajectory of the hand than the motion of the joints" and guesses that it might be possible to describe the paths by optimizing a suitable criterion. In [2] basic pointing tasks are considered and roughly straight lines with bell-shaped velocity profiles are observed. Additionally, the task of following a curved path is analyzed and segmentation noticed.

Similar characteristics are obtained for three-dimensional movements in [219] where it is noted that point-to-point movements at natural speed produce approximately straight trajectories with bell shaped velocity profiles. Furthermore, it is observed that curved motions are essentially planar. Alternatively to these experimental results, a simulation-based approach is used by [161] to study the influences of interaction torques on the human arm motions. The results show that the forces generated by Coriolis, centripetal and inertia properties are essential for planar human arm motions. Additionally, the significance of the velocity torques relative to the inertial torques does not change with movement speed.

The observation that planar arm motions are mostly straight is limited by the discovery of movement regions within the workspace where the hand paths are noticeably and systematically curved [16], which is affirmed by [66, 314] for planar motions if the start and end points are in an uncomfortable position, i.e., near the workspace boundaries. In addition to the standard horizontal movements, upward and downward motions are analyzed by [16, 98] and it is observed that the upwards arm movements are more curved than the downward ones.

Based on the observation of these characteristics of the hand paths, Flash and Hogan [104] and many others argue that a planning scheme based on the Cartesian coordinate frame is

generally used in human motor control, even for tasks that do not require the hand trajectory to be explicitly controlled. In opposition, the idea of motion control on joint level is favored by others, e.g., [314]. The choice of the coordinate system underlying the human motion generation is closely related to the question whether the trajectories are formed on the basis of the kinematics or the dynamics of the human, which leads to the proposition of different cost functions for open-loop control, see section 6.4. An experiment to solve the question is discussed in [344], where the visual feedback of the task is disturbed. Since humans do react to this false feedback, the arm movements are not controlled by dynamics only. After perturbation humans straighten their paths, but the original ones are not obtained again.

A basic observation in all experiments of human arm motions is that a considerable variance between different trials of the identical motion task exists, e.g., [129]. This variation seems to be a fundamental characteristic and can be attributed to the impedance characteristics of the human arm and the noise characteristics of the human muscles (cf. section 6.3). Furthermore, it is observed that characteristics vary for different tasks, especially, if free motion is compared to constrained motion [77]. Consequently, one might assume that humans control their movements in dependence on the task.

In the following, we will not only discuss Fitts' law relating motion time and accuracy, but also Donders' law and the two-thirds power law; all these relations are based on empirical observations and do not take the dynamics into account.

B.2.1 Fitts' Law

Fitts [96] presents a relation between movement time and task-dependent accuracy to reach the goal; consequently, the time needed by the human to do certain arm motions is neither random nor a direct consequence of the kinematics or dynamics of the arm, but it is itself a variable planned according to the accuracy of the task.

$$\delta = c_1 + c_2 \log_2 \left(1 + \frac{l_1}{l_2}\right),$$

where δ is the motion time, l_1 the distance from the starting position to the goal and l_2 the width of the target which can be interpreted as the allowed error tolerance. The constants c_1 and c_2 are problem-dependent and have in consequence to be matched to the experimental data.

Fitts' law, being of empirical nature, has been verified by many research groups and to obtain a closer fit to certain data, several variants of the law are known, but these improvements come with a loss in structure, see [322] for a review.

Fitts motivates his formula using the information theory of Shannon and Weaver by assuming that the motor control task is equivalent to the transport of information. He assumes that due to noise the human motor system has a limited capacity of transport and the maximal amount of information which can be transfered by the system leads then to the log-linear relationship [96]. This information theory background caused numerous criticism for the formula which seems empirically valid [322].

Since Fitts' law is closely related to the inherent noise of the human motor system caused by the basic build-up and the biomechanical characteristics [322], it seems reasonable that the minimum variance model (it assumes that humans control their arm motions in a manner to achieve minimal variance at the goal position while taking the noisy muscle characteristics

into account, see section 6.4) does correctly reproduce the speed-accuracy trade-off [141]. From the biological perspective the implications of this method seem somewhat implausible and inefficient [298], because this would imply that the human learning process has to test various task-dependent motion times.

B.2.2 Donders' Law and Listing's Law

Analyzing saccadic eye motions with a fixed head, Donders observed that at the end of the motion the eye has a constant orientation independent of the actual trajectory to the final configuration; this observation is called Donders' law. A stronger relation including Donders' law is introduced in the following by Listings, who proposed that only postures are achieved that can be attained by one rotation, i.e., a fixed rotation axis for the overall eye motion. For more information on human eye movements see [190].

In the case of redundant arm movements, it is a straight-forward idea to analyze whether Donders' law can be used to describe the final configurations of a human arm, too. A detailed analysis of pointing tasks [285] questions Donders' law for arm movements, because small but systematic deviations from a unique configuration are observed. In a similar manner, studies of general three-dimensional movements [3] reveal that the model cannot give a good prediction for the data. Consequently, the only workaround to use Donders' law for determining arm configurations seems to be to restrict the relation to the motion of the upper arm element only [196]. If one assumes that the motions of the lower arm element depend linearly on the movements of the upper element, the characteristic of the upper arm can be used to determine the hand path on a kinematic basis [197].

B.2.3 Two-Thirds Power Law

Finally, for the special task of tracing a simple planar figure with the hand, we want to mention the two-thirds power law capturing that the angular velocity ω of the hand is coupled to the curvature κ of the traced curve:

$$\omega = c(\kappa)^{2/3},$$

where c is a task-dependent constant. Several variants of this law for the relation of hand motion and curvature are developed in literature, see [328] and previous works of the authors mentioned therein.

Another approach [328] to explain the characteristics of curve-following is to use the theory of minimizing hand jerk, the third time-derivative of the hand position (cf. section 6.4). It is noted in [259] that the velocity profiles are predicted accurately for a given trajectory and the minimization idea even captures a few details of the hand motion which elude the two-thirds power law [328, 306].

Three-dimensional path-following tasks are analyzed in [206] and it is shown that the two-thirds power law does not hold in 3D. Therefore a new power law is introduced to connect curvature and hand speed in 3D tasks.

B.3 Motor Control

Having discussed some characteristics of human arm motions, the focus of this section is on properties of the general motor control task. This includes the problem of ambiguity in

controls and the proposition of internal models as a basic element in motion planning.

Point-to-point arm movements are the special type of motor control tasks we will focus on. A common feature of motor control task is that the task requirements can be met by infinitely many diverse movements. Thus, stating only the boundary conditions of the motion for given dynamics leads to an ill-defined problem. The ambiguity causing this problem can be resolved if an optimality principle is applied.

The basis of many scientific theories is formed by optimality principles. In the context of human arm movements, the underlying assumption is that humans minimize an unknown cost function. An open question is if a single cost function can explain the observations or whether multiple criteria, each suitable for a small set of contexts [144], are needed [3]. The performance of the models might depend on the context and instruction to the participant [307], for example, the end-state of a manipulation task seems to have an effect on motor planning [357].

To apply such a minimization idea resulting in an optimal control problem, several choices have to be made by the modeler ranging from the choice of optimization principles over the model of the musculoskeletal plant to a measure for task-performance [302]. Each of these aspects is discussed for the examples in this work (cf. sections 6.5 and 6.4), but in the following we focus on how humans are supposed to solve this optimal control problem.

The general question is whether humans use internal models capturing input-output characteristics of the human motor plant to control their movements or not; this resulted in a long-standing discussion. Several experiments to support each side are presented in literature; while [235] argues that from a psychological point of view little empirical evidence is found that plainly supports the internal model idea, others consider it well-established that the central nervous system makes use of the computational principles of internal models [175, 267, 275].

One possibility to control the arm without a model could be closed-loop control, where at each time instance feedback about the momentary state of the arm is available (cf. section B.5). Such an approach where coordinated arm movements are executed solely under feedback control is discarded by [175] due to the biological feedback loops which are slow and have small gain.

The internal models themselves are divided into two classes, the forward models and the inverse models. A forward model predicts the state of the arm if a control is specified, whereas the inverse model gets as an input the desired state and has to compute a corresponding control. This clear distinction between forward and inverse models seems to hold in the human brain [342].

Evidences supporting the idea of internal models are increasing [130, 210, 279]; for detailed reviews see [76, 175]. Where the results of [345] and [211] using different experimental setups clearly support the existence of a predictive forward model, other experiments seem to support the idea of inverse models (compare section B.6 on learning and adaptation). In consequence, a framework using multiple internal models of both types is introduced in [347] and experimental results of [28, 97, 185] support this idea.

The concept of internal models leads directly to the question of human motion generation.

B.4 Motion Generation

Several hypotheses how humans could plan and control their movements have been introduced in literature; ranging from purely geometrical considerations to dynamically controlling the muscle force.

B.4.1 Redundancy

Many tasks of human arm motion can be achieved in many different ways, e.g., there might exist several arm configurations yielding the same hand position. This property is called redundancy [25]. In consequence the question arises how humans determine their characteristic arm movements. Humans seem to use this freedom to fulfill the task in the most convenient way.

If various arm trajectories yield the same performance with respect to the goal of the task, the set of all these trajectories is called the uncontrolled manifold. Various theories exist how humans choose their trajectories, e.g., avoidance of joint limits [67] or task specific optimization [232]. Other approaches are based on the idea of minimal intervention [43, 307] which will be discussed later.

A posture-based planning framework to determine valid goal positions is presented by [262]. The model has been constantly developed [261, 263] to account for further experimental observations. The central idea is to combine stored postures to get to the specified positions; most important, the goal postures are planned prior to the movement. Costs of possible postures and postural transitions are taken into account; in the later versions of the model a hierarchy of requirements is introduced and a two-stage process is used to obtain better goal postures. The model has been extended to full three-dimensional motions instead of only motions in a plane and to include obstacle avoidance, but so far it does not consider the dynamics for the actual motion generation, using a generic velocity profile instead [261].

A further theory to solve the redundancy problem is hierarchical control. Most hierarchical approaches originate in optimal feedback control (cf. section B.5), but an open-loop approach based on task-decomposition is presented in [176]. Full-body control is computed by projecting the state and the control onto the relevant spaces. In the context of feedback control a related method based on dimensionality reduction [308] is used. The combination of a low-level feedback controller and a high-level controller based on an abstract and compact state representation is adopted to mimic the different human feedback loops influencing redundant control. The critical point of this approach is to find the right representations for the higher level. Using this approach hierarchical control is computed in [199] for a redundant three-dimensional arm considering nonlinear muscle behavior.

B.4.2 Motion Control

Mainly two principles for controlling human movements are discussed in literature. First, the idea that humans control their motions on the level of muscle force generation is introduced in [161]. This approach is close to the engineering perspective on the related problem of steering a robotic arm. Where the human uses his muscle forces to generate torques moving the limbs, robotic systems have several drives for torque generation. The second principle is the equilibrium point hypothesis introduced by [158]. The goal is a unifying framework for posture and motion by using the stability properties of the human arm.

Whether human arm motions are really controlled as a dynamic optimization problem only is not answered conclusively. Aiming for a biologically plausible model for the generation of arm movements, a geometrical stage between the sensory input and the physical execution is introduced in [309]. The shortest path in joint space is utilized at the geometrical stage before the kinematics are computed. As a motivation for their approach it is noted that "there are brain areas with force-independent representations of movement" [309]. Imprecision for three-dimensional motion directed towards the head are reported in [247]. A related approach is presented in [29] where the positions are first determined by the shortest path on the Riemann sphere and then the velocity profile results from a jerk minimization along this path. This strategy yields geodesic paths reducing the torques due to fact that only the driving torques remain. Results are presented for the dynamical model of a three-dimensional arm.

The theory of force control is questioned for human arm motions from a psychological point of view in [235]. It is noted that observed EMG data does not match the model predictions. Further problems arise if physiologically realistic muscle and reflex models are incorporated into the force control model, which lead to the basic question of how the change between stable posture and motion should happen. If a person is at rest, the body reacts resistively to any perturbations. Experimental observations show that such reactions are not suppressed during movement and that postural control and co-activation of muscles are in principle separate mechanisms [235]. Consequently, even the combination of force control and postural resetting by using paired inverse and forward models [28] does not resolve the problem, because it is based on a change in muscle activation [235].

Where the spring-like properties reported for the human muscles and reflex loops cause problems from the force control perspective, these characteristics are the starting point for the equilibrium hypothesis using the observation that the viscoelasticity can be controlled by adjusting the co-contraction level. The idea is that the arm is controlled by a series of stable equilibrium positions between the beginning and the endpoint of the movement. The differences between the actual and the equilibrium positions in combination with the used stiffness and viscosity determine the forces exerted on the arm. In [31] the conclusion is drawn that the measured forearm movements of the monkeys (one degree of freedom only) can be described as a series of equilibrium points towards the final position.

Several variants of the hypothesis are known, which differ in the model details like spinal reflexes. They all have the following three levels in common [127]: First, the spring-like properties of the neuromuscular system are utilized in movement control. Second, the brain uses an equilibrium point trajectory as descending motor commands to the spinal cord. Third, the equilibrium point trajectory can simply be planned; thus, the brain does not need to solve the dynamics problem [93].

This equilibrium point hypothesis raised the question whether the spring-like properties can, in addition to stabilizing the posture, bring about the motion by itself [127]. Experiments of [126, 189] show that the equilibrium hypothesis cannot predict effects observed by changing dynamical properties, which might indicate that the human brain actually does consider the dynamic influences on motion generation and not only relies on the elastic properties [126]. On the other hand, a similarity between the equilibrium point trajectory and the actual one is reported by [100, 127] for movements with high arm stiffness. However, [173] claim that planning an equilibrium point trajectory resulting in the observed motions is as difficult as explicitly solving the dynamic equations due to the low hand stiffness during motion [24]. Furthermore, a good approximation is only obtained if high stiffness is assumed and trajectories computed with a low stiffness differ from the observed human ones [173].

The results of the experiments of [237] show that the relative contributions of viscoelasticity and internal model control change in the course of adaptation to a setting. While the viscoelastic effects dominate in an unknown environment, the part attributed to an internal model increases during adaptation, which can be seen as an improvement process of the internal model. For further details on human adaptation characteristics see section B.6.

Summing up, the equilibrium point hypothesis is based on characteristic properties of the human arm, but it fails to account for motions with low arm stiffness and for changes due to modifications of a dynamical environment. On the other hand, if one assumes that the arm motion is generated by muscle forces, one can reproduce the observed hand paths and account for changes of the systems dynamics, but there might be no biological or psychological foundation for this approach. In this work we will use the force control perspective and, consequently, the goal of the discussed models is to capture the main features of the experiments; in particular, we do not aim to explain phenomena from a biological or psychological perspective (cf. section B.1).

B.5 Open-Loop and Closed-Loop Control

Given the characteristics of human arm motion and using the idea of internal models, a common assumption is that humans try to execute their movements optimally. Strategies towards this optimality are based on adaptation to new tasks and learning of new influences. Various approaches have been made to model these learning strategies themselves (see section B.6). Here we assume that these learning processes are completed and that "the sensorimotor control is best described as being near optimal" [308].

Now, the goal is to find a framework that can explain the human motion under the optimality assumption. Two main approaches have to be distinguished: On the one hand, the open-loop approach is based on the idea that the whole control sequence is generated before the motion starts and is executed without any feedback. On the other hand, the closed-loop frameworks actively use feedback during the human movements. If one wants to analyze human learning strategies such a feedback loop is essential, but the derivation of optimal control strategies is by far more complex for the closed-loop approach than for the open-loop one [307]. Naturally, at the beginning research was focused on the open-loop idea; thus we will start with a discussion of the main research directions within this scope.

A large number of models of open-loop motor control exist and each model claims to describe human motion, but several models are incompatible with others. The starting point for the derivation of a cost function are characteristics of the human arm movements and the human organism. Humans might minimize the sequence of control signals [12, 60, 150], or limb states [104, 141, 225, 314]. These minimization strategies are related to physiological and task variables such as smoothness of the hand path [104, 314], accuracy [46, 141] or error and effort [9, 88, 116, 240, 307]. Some of these costs seem to be more natural than others, because the human organism has only sensors for some of these properties [337].

Supporting experimental data exists for most of these cost functions. To solve this ambiguity many experiments are discussed, for example, the spring-based experiment in [67] exploring the differences between kinematic and dynamic costs, but no clear results in favor of only one cost or planning space are obtained. In section 6.4 the open-loop cost functions are introduced in more detail and the pros and cons for the individual costs and possible combinations are discussed.

Open-loop control discussed so far computes optimal controls for a given task and these controls are used if the motion is carried out. This frame is static and does not use any feedback. Humans, however, use sensory feedback, for example, vision, to control their arm motions, since perturbations caused by the environment and random effects resulting from the noise of the human motor plant influence the motion. If no feedback is used when a stochastic partially-observable plant such as the human musculoskeletal system is controlled, only sub-optimal performance can be achieved [303, 307], which would be contrary to the observed human characteristics. The goal of closed-loop control is to model these feedback loops and to compute the optimal controller utilizing the accessible feedback at each time instance to obtain a stable control, meaning that one has to "solve a control problem repeatedly rather than repeat its solution" [25]. The problem with the closed-loop theory is that the utilized methods to approximate optimal feedback controllers are complex and computationally expensive [302]. The information represented by the optimal value function might visualize this issue, because it combines the optimality information for motions between all points in the state space [215]. The hypothesis of [299] is that fundamental developments in the control theory are still necessary to understand motor function in detail.

Searching for the biological foundations of feedback, various experiments are discussed in literature, compare [188, 267]. Resulting hypotheses are that the posterior parietal cortex is updating the motor plan, while the cerebellum might compute the feed-forward control signals. Further structures of the human nervous system might contain the state estimator and comparator [267]. The influences of visual feedback on human arm motions are a topic of discussion [76], but experimental results, e.g., [271], suggest that humans continuously use visual feedback throughout their arm motions. Other experiments [99, 189] show, for example, that the sensory feedback and the predictions of the forward model are combined by humans in a statistically optimal way. Additionally, humans seem to be able to predict the influences of external loads, which raises the question if internal models of the environment are utilized and updated by feedback. Controlling a human arm solely by feedback is impossible, because the human sensorimotor feedback loops are too slow [342]. Consequently, internal estimation of the state is needed in combination with the delayed feedback of noisy sensors [342].

To model the feedback loop in a closed-loop framework, knowledge about the human adaptation strategies is helpful. Adaptation of the reaching movements can be done on two different levels. In the first case the task is varied and the controls have to be adapted to fulfill the goals. This type of adaptation changes the controls, but the structure of control remains the same, i.e., if one assumes that internal models and a cost function are utilized to describe human motions, they remain unchanged by the considered task variations. In the other case the environment is changed such that a new control strategy has to be learned. This learning is also based on feedback and will be discussed in section B.6. We focus here on the first type of adaptation demonstrating the structure of the feedback control.

The standard technique to analyze adaptation to variations of the task is to introduce perturbations of the final position at the beginning of the movement. The observation is that the hand path is smoothly corrected in agreement with multiple models, e.g., [102, 156, 309]. In [198] perturbations taking place towards the end of the movement are analyzed and it is reported that the resulting deviations are not fully corrected, i.e., a systematic endpoint error is observed and has to be explained by a suitable closed-loop framework.

In the following we want to give an introduction to several approaches that try to model as many of the observed human characteristics as possible. A first approach to extend the results of the open-loop control theory to a framework including feedback is discussed by [102].

Using the minimum jerk idea, changes of the goal position result in smooth changes of the trajectory similar to the observed human behavior. If optimal feedback control is combined with the minimum jerk hypothesis, the resulting trajectories are consistent with results of several target perturbation studies [156]. Other approaches to model this characteristic of human arm motions are based on a virtual spring pulling the hand toward the final position [156, 309].

If a controller is based on optimal feedback theory, a minimum intervention principle holds for a redundant task, meaning that only deviations from the average behavior that interfere with the task performance are corrected [307]. From the biological perspective such a principle seems preferable, because intervention generates control-dependent noise and energy costs [302]. Experiments indicate that this minimum intervention principle observed in human motions cannot be explained by signal-dependent noise only [317] . The controller presented in [307] employing locally linearized models of the problem, i.e., an iterative LQG strategy [194], is capable to explain some phenomena of human motions. For others it seems to be necessary to tackle the more complicated problem of studying optimal feedback control for nonlinear models, but even the computation of the optimal feedback control for movements with one degree of freedom assuming a linear plant prove to be rather complex [303]. To overcome these difficulties a strategy based on space discretization and dynamic programming is presented in [198]. A composite cost function combining end point accuracy, velocity and accelerations at the end and the integral of squared controls is utilized and the resulting strategy reproduces the undershoot in the final hand position of humans in context with perturbations introduced late in the motion.

To conclude the discussion of control strategies based on feedback, some recent examples realizing closed-loop control for three-dimensional arm motions are mentioned. The approach of [304] using a reformulation of the Bellman equation considers a simplified structure of the cost function and the noise to apply convex optimization. Assuming separation of static and dynamic forces in combination with a constant effort hypothesis, some human movements can be reproduced [135], but the focus of this work is on stereotyped motions, leaving out non-symmetric velocity profiles or avoidance of extreme joint limits [247]. The iterative LQG algorithm used in [215] minimizes energy but assures compliance during the motion; the physical constraints of the arm are incorporated into the cost function. Considering hand dynamics in form of a noisy spring-damper system and modeling kinematics and dynamics of the arm in the form of a force field, the control strategy for signal-dependent noise is reported to be asymptotically and globally stable [247].

B.6 Adaptation

In this section we want to discuss the part of adaptation of human motions related to the transfer of previously learned motor skills to new contexts. Learning new dynamics or new influences from the environment seems to be a continuous process to improve behavioral performance [302] with the ultimate goal of optimality [225]. To do so, learning strategies might modify various parts of the strategies discussed so far; for example, updating internal models and searching for the optimal controller. Such learning processes are used by humans on a daily basis when manipulating objects [164], but experiments of [270, 349] show that the task of learning an optimal control strategy needed for a given setup can be a hard if not impossible task, i.e., human adaptation capabilities have limits.

Several experiments have been conducted to analyze human learning behavior and two main approaches can be distinguished [279]. On the one hand, the perceived kinematics are altered by changing the visual feedback. On the other hand, the dynamics of the system can be altered by using technical devices; the two most common instruments are a manipulandum, e.g., [82], and an exoskeleton platform, e.g., [214]. The final goal of these devices is to allow motions with unaltered dynamics, but on the other hand it should be possible to create arbitrary dynamical environments.

The results of several experiments analyzing human learning of new environments are discussed in literature, see for example [175, 185, 189, 279]. It is observed that the subjects learn to compensate for the new dynamics and, after the learning period, generate motions similar to those in the original environment. If the external influences are removed, the opposite process is observed which might support the idea that humans learns internal models [48, 88]. Especially the experiment of [47] where a divergent force field is used to generate an unstable interaction with the human arm shows that humans overcome the instability by increasing the mechanical impedance of the arm selectively in the unstable direction. It is noted in [237, 297] that building an inverse dynamics model and impedance control reducing the influences of model errors are two coexisting but separate mechanisms of human learning. In case of unstable interaction the impedance is preferentially controlled [117], whereas in the stable case the focus is on building an inverse model of the mean dynamics [236]. The impedance of the arm is increased at the beginning of the learning in both cases, but is gradually reduced in the stable environment [118].

The formation of the inverse model can be described by feedback error learning, i.e., feedback information is utilized to modify the feed-forward motor commands. The standard approach is to assume a quadratic cost if a learning task is modeled, but experiments [180] show that humans might use a cost function that penalizes outliers significantly less than quadratic. Consequently, a quadratic cost might only be a good approximation if nonlinearities have only a small influence on the learning process. Closely related is the question how prior knowledge is combined with the new information from feedback. In [179] it is shown that human behavior is close to the Bayesian optimal way. In the conducted experiments the bias and the noise level of the visual feedback is manipulated in point-to-point reaching tasks. Other experiments [204, 310] are based on an explicit reward function and humans seem to quickly maximize the potential reward within certain limits [181]. More studies utilizing reward functions depending on several parameters could give deeper insight into the human learning strategies [180, 181].

Learning frameworks are now discussed that capture the adaptation process observed in the experiments, as opposed to the optimal control setups which are based exclusively on optimization of one specified cost function and consequently, can only predict final outcome after learning. Two examples of such learning frameworks trying to reproduce the changes in force and impedance are [46, 116]. Others [84, 346] base their frameworks on multiple internal models; their motivation comes from the parallels to machine learning, e.g., [165], where a network outperforms a single agent. In motor control each of the internal models specializes on a specific task or environment. Learning can then be interpreted as an adaptation of the selection process. Experiments supporting the multiple internal model idea can be found, for example, in [28, 97, 185]. The multiple internal model idea leads to the field of motion primitives which are discussed in section 6.7.

B.7 Discussion of Implications

The goal of our inverse optimal control approach is to find an optimal combination of basic cost functions that reproduces common characteristics of human motions. The optimality of a combination depends on the analyzed task, the dynamical model of the plant and the family of considered basic cost functions. Consequently, the result of the bilevel optimization can be interpreted as a model capturing certain characteristics of the data, but one should not mistake it for a biologically plausible model; especially, it cannot be assumed that the optimal cost function captures the biological or psychological reasons for the observed characteristics (cf. section B.1). Since the considered application scenarios are related to robotics, the main requirement of the obtained optimal control model is that changes in the task specification and in the kinematic or dynamic properties can be accounted for. The dynamical arm models used in this work are based on the assumption that the arm movements are executed by controlling the joint torques or the muscle forces. This is an assumption considered reasonable by most works on this topic, but nevertheless it might not capture all aspects of the biological plant (cf. section B.4). We assume that the additional degrees of freedom causing redundancy are used by the human to minimize the cost values.

The basic assumption of our work is that the observed human motions can be reproduced by open-loop control. In general, humans do use different kinds of feedback to correct the movement execution. Especially in the context of adaptation to new tasks or new dynamical environments feedback is needed (cf. sections B.5 and B.6). The learning strategies used by the human to improve performance are assumed to generate or update certain motion models in the human brain (cf. sections B.3). Consequently, we assume that the recorded human motion is the end product of a human learning strategy and thus nearly optimal for an unknown cost function subject to the dynamics of the plant.

Bibliography

[1] P. Abbeel and A.Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2004.

[2] W. Abend, E. Bizzi, and P. Morasso. Human arm trajectory formation. *Brain*, 105(2):331–348, 1982.

[3] M.A. Admiraal, M.J. Kusters, and S.C. Gielen. Modeling kinematics and dynamics of human arm movements. *Motor Control*, 8(3):312–338, 2004.

[4] S. Albrecht, P. Basili, S. Glasauer, M. Leibold, and M. Ulbrich. Modeling and analysis of human navigation with crossing interferer using inverse optimal control. In *Proceedings of the Vienna International Conference on Mathematical Modeling*, 2012.

[5] S. Albrecht, M. Leibold, and M. Ulbrich. A bilevel optimization approach to obtain optimal cost functions for human arm movements. *Numerical Algebra, Control and Optimization*, 2(1):105–127, 2012.

[6] S. Albrecht, C. Passenberg, M. Sobotka, A. Peer, M. Buss, and M. Ulbrich. Optimization criteria for human trajectory formation in dynamic virtual environments. *Lecture Notes in Computer Science (Haptics: Generating and Perceiving Tangible Sensations)*, 6192:257–262, 2010.

[7] S. Albrecht, K. Ramırez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz. Imitating human reaching motions using physically inspired optimization principles. In *Proceedings of the International Conference on Humanoid Robots*, pages 602–607, 2011.

[8] R.McN. Alexander. A model of bipedal locomotion on compliant legs. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 338(1284):189–198, 1992.

[9] R.McN. Alexander. A minimum energy cost hypothesis for human arm trajectories. *Biological Cybernetics*, 76(2):97–105, 1997.

[10] S. Ambler and A. Paquet. Recursive methods for computing equilibria of general equilibrium dynamic stackelberg games. *Economic Modelling*, 14(2):155–173, 1997.

[11] G. Anandalingam and T.L. Friesz, editors. *Hierarchical Optimization*, volume 34 of *Annals of Operations Research*. Springer, 1992.

[12] F.C. Anderson and M.G. Pandy. Dynamic optimization of human walking. *Biomechanical Engineering*, 123(5):381–390, 2001.

[13] M. Anitescu. On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(4):1203–1236, 2005.

[14] B.D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[15] U.M. Ascher, R.M.M. Mattheij, and R.D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Society for Industrial and Applied Mathematics (SIAM), 1995.

[16] C.G. Atkeson and J.M. Hollerbach. Kinematic features of unrestrained vertical arm movements. *Biological Cybernetics*, 44:67–77, 1985.

[17] C.A. Balafoutis, R.V. Patel, and P. Misra. Efficient modeling and computation of manipulator dynamics using orthogonal cartesian tensors. *Robotics and Automation*, 4(6):665–676, 1988.

[18] J.F. Bard. An algorithm for solving the general bilevel programming problem. *Mathematics of Operations Research*, 8(2):260–272, 1983.

[19] J.F. Bard. Convex two-level optimization. *Mathematical Programming*, 40(1):15–27, 1988.

[20] J.F. Bard. Some properties of the bilevel programming problem. *Optimization Theory and Applications*, 68(2):371–378, 1991.

[21] J.F. Bard. *Practical Bilevel Optimization: Algorithms and Applications*. Springer, 1998.

[22] T. Başar and G.J. Olsder. *Dynamic Noncooperative Game Theory*. Society for Industrial and Applied Mathematics (SIAM), 1999.

[23] P.M. Bays and D.M. Wolpert. Computational principles of sensorimotor control that minimize uncertainty and variability. *Physiology*, 578(2):387–396, 2007.

[24] D.J. Bennett, J.M. Hollerbach, Y. Xu, and I.W. Hunter. Time-varying stiffness of human elbow joint during cyclic voluntary movement. *Experimental Brain Research*, 88(2):433–442, 1992.

[25] N.A. Bernstein. *The Co-ordination and Regulation of Movements*. Pergamon Press, 1967.

[26] B. Berret, E. Chiovetto, F. Nori, and T. Pozzo. Evidence for composite cost functions in arm movement planning: an inverse optimal control approach. *PLoS Computational Biology*, 7(10):e1002183(1–18), 2011.

[27] J.T. Betts. *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*. Society for Industrial and Applied Mathematics (SIAM), 2010.

[28] N. Bhushan and R. Shadmehr. Computational nature of human adaptive control during learning of reaching movements in force fields. *Biological Cybernetics*, 81(1):39–60, 1999.

[29] A. Biess, D.G. Liebermann, and T. Flash. A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics. *Neuroscience*, 27(48):13045–13064, 2007.

[30] A.G. Billard. Learning motor skills by imitation: a biologically inspired robotic model. *Cybernetics and Systems*, 32(1):155–193, 2001.

[31] E. Bizzi, N. Accornero, W. Chapple, and N. Hogan. Posture control and trajectory formation during arm movement. *Neuroscience*, 4(11):2738–2744, 1984.

[32] E. Bizzi, A. d'Avella, P. Saltiel, and M. Tresch. Modular organization of spinal motor systems. *The Neuroscientist*, 8(5):437–442, 2002.

[33] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In *Modelling of Chemical Reaction Systems*, pages 102–125. Springer, 1982.

[34] C.L. Bottasso, B.I. Prilutsky, A. Croce, E. Imberti, and S. Sartirana. A numerical procedure for inferring from experimental data the optimization cost functions using a multibody model of the neuro-musculoskeletal system. *Multibody System Dynamics*, 16(2):123–154, 2006.

[35] L. Boutin, A. Eon, S. Zeghloul, and P. Lacouture. An auto-adaptable algorithm to generate human-like locomotion for different humanoid robots based on motion capture data. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1256–1261, 2010.

[36] J. Bracken and J.T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973.

[37] J. Bracken and J.T. McGill. Defense applications of mathematical programs with optimization problems in the constraints. *Operations Research*, 22(5):1086–1096, 1974.

[38] M.H. Breitner and H.J. Pesch. Reentry trajectory optimization under atmospheric uncertainty as a differential game. In T. Başar and A. Haurie, editors, *Differential Games and Applications*, pages 70–87. Birkhäuser, 1994.

[39] M.H. Breitner, H.J. Pesch, and W. Grimm. Complex differential games of pursuit-evasion type with state constraints, part 1: Necessary conditions for optimal open-loop strategies. *Optimization Theory and Applications*, 78(3):419–441, 1993.

[40] M.H. Breitner, H.J. Pesch, and W. Grimm. Complex differential games of pursuit-evasion type with state constraints, part 2: Numerical computation of optimal open-loop strategies. *Optimization Theory and Applications*, 78(3):443–463, 1993.

[41] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics (SIAM), 1996.

[42] I.E. Brown, S.H. Scott, and G.E. Loeb. Mechanics of feline soleus: Design and validation of a mathematical model. *Muscle Research and Cell Motility*, 17(2):221–233, 1996.

[43] H. Bruyninckx and O. Khatib. Gauss' principle and the dynamics of redundant and constrained manipulators. In *Proceedings of the International Conference on Robotics and Automation*, volume 3, pages 2563–2568, 2002.

[44] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Hemisphere Publishing Corporation, 1975.

[45] R. Bulirsch. Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung. *Report der Carl-Cranz-Gesellschaft*, 1971.

[46] E. Burdet and T.E. Milner. Quantization of human motions and learning of accurate movements. *Biological Cybernetics*, 78(4):307–318, 1998.

[47] E. Burdet, R. Osu, D.W. Franklin, T.E. Milner, and M. Kawato. The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, 414(6862):446–449, 2001.

[48] E. Burdet, K.P. Tee, I. Mareels, T.E. Milner, C.M. Chew, D.W. Franklin, R. Osu, and M. Kawato. Stability and motor adaptation in human arm movements. *Biological Cybernetics*, 94(1):20–32, 2006.

[49] C. Büskens. Optimierungsmethoden und Sensitivitätsanalyse für optimale Steuerungsprozesse mit Steuer- und Zustandsbeschränkungen. Dissertation, Institut für Numerische Mathematik, Universität Münster, 1998.

[50] T. Butz. Optimaltheoretische Modellierung und Identifizierung von Fahrereigenschaften. In *Fortschritt-Berichte VDI*, number 1080 in Reihe 8. VDI Verlag, 2005.

[51] R.H. Byrd, J.C. Gilbert, and J. Nocedal. On the local behavior of an interior point method for nonlinear programming. *Numerical Analysis*, pages 37–56, 1997.

[52] A.F. C. Hamilton, K.E. Jones, and D.M. Wolpert. The scaling of motor noise with muscle strength and motor unit number in humans. *Experimental Brain Research*, 157(4):417–430, 2004.

[53] S. Calinon and A.G. Billard. Statistical learning by imitation of competing constraints in joint space and task space. *Advanced Robotics*, 23(15):2059–2076, 2009.

[54] R. Callies and P. Rentrop. Optimal control of rigid-link manipulators by indirect methods. *GAMM-Mitteilungen*, 31(1):27–58, 2008.

[55] R. Callies and T. Schenk. Recursive modelling of optimal control problems for multi-link manipulators. Technical Report NUM 13, Zentrum Mathematik, Technische Universität München, 2005.

[56] R. Callies and G. Wimmer. Optimal hypersonic flight trajectories with full safety in case of mission abort. In *Proceedings of the Atmospheric Flight Mechanics Conference*, 2000.

[57] W. Candler and R. Townsley. A linear two-level programming problem. *Computers and Operations Research*, 9(1):59–76, 1982.

[58] L. Cesari. *Optimization - Theory and Applications*. Springer, 1983.

[59] C.L. Chen and J.B. Cruz. Stackelberg solution for two-person games with biased information patterns. *IEEE Transactions on Automatic Control*, 17(6):791–798, 1972.

[60] C.K. Chow and D.H. Jacobson. Studies of human locomotion via optimal programming. *Mathematical Biosciences*, 10(3):239–306, 1971.

[61] F.H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley, 1983.

[62] F.H. Clarke and G.R. Munro. Coastal states and distant water fishing nations: conflicting views of the future. *Natural Resource Modeling*, 5(3):345–369, 1990.

[63] J.J. Collins. The redundant nature of locomotor optimization laws. *Biomechanics*, 28(3):251–267, 1995.

[64] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49:1–23, 1943.

[65] J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1986.

[66] H. Cruse and M. Brüwer. The human arm as a redundant manipulator: the control of path and joint angles. *Biological Cybernetics*, 57(1):137–144, 1987.

[67] H. Cruse, E. Wischmeyer, M. Brüwer, P. Brockfeld, and A. Dress. On the cost functions for the control of the human arm movement. *Biological Cybernetics*, 62(6):519–528, 1990.

[68] M. Damsgaard, J. Rasmussen, S.T. Christensen, E. Surma, and M. de Zee. Analysis of musculoskeletal systems in the anybody modeling system. *Simulation Modelling Practice and Theory*, 14(8):1100–1111, 2006.

[69] D. Del Vecchio, R.M. Murray, and P. Perona. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica*, 39(12):2085–2098, 2003.

[70] S.L. Delp, F.C. Anderson, A.S. Arnold, P. Loan, A. Habib, C.T. John, E. Guendelman, and D.G. Thelen. Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 54(11):1940–1950, 2007.

[71] S.L. Delp and J.P. Loan. A graphics-based software system to develop and analyze models of musculoskeletal structures. *Computers in Biology and Medicine*, 25(1):21–34, 1995.

[72] V. DeMiguel, M.P. Friedlander, F.J. Nogales, and S. Scholtes. A two-sided relaxation scheme for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 16(2):587–609, 2005.

[73] J. Demiris and G. Hayes. Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model. In *Imitation in Animals and Artifacts*, pages 327–362. MIT Press, 2002.

[74] S. Dempe. *Foundations of Bilevel Programming*. Kluwer Academic Publishers, 2002.

[75] S. Dempe and N. Gadhi. Necessary optimality conditions for bilevel set optimization problems. *Global Optimization*, 39(4):529–542, 2007.

[76] M. Desmurget and S. Grafton. Forward modeling allows feedback control for fast reaching movements. *Trends in Cognitive Sciences*, 4(11):423–431, 2000.

[77] M. Desmurget, M. Jordan, C. Prablanc, and M. Jeannerod. Constrained and unconstrained movements involve different control strategies. *Neurophysiology*, 77(3):1644–1650, 1997.

[78] P. Deuflhard and F. Bornemann. *Numerische Mathematik: Gewöhnliche Differentialgleichungen*, volume 2. de Gruyter, 2002.

[79] M. Diehl, D.B. Leineweber, and A.A.S. Schäfer. Muscod-II user's manual. IWR-Preprint, University of Heidelberg, 2001.

[80] R. Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2):109–116, 2004.

[81] M. Do, P. Azad, T. Asfour, and R. Dillmann. Imitation of human motion on a humanoid robot using non-linear optimization. In *Proceedings of the International Conference on Humanoid Robots*, pages 545–552, 2008.

[82] O. Donchin, J.T. Francis, and R. Shadmehr. Quantifying generalization from trial-by-trial behavior of adaptive systems that learn with basis functions: theory and experiments in human motor control. *Neuroscience*, 23(27):9032–9045, 2003.

[83] K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

[84] K. Doya, K. Samejima, K. Katagiri, and M. Kawato. Multiple model-based reinforcement learning. *Neural Computation*, 14(6):1347–1369, 2002.

[85] C. Dube and J. Tapson. Kinematics design and human motion transfer for a humanoid service robot arm. In *Proceeding of the Robotics and Mechatronics Symposium*, 2009.

[86] T.A. Edmunds and J.F. Bard. Algorithms for nonlinear bilevel mathematical programs. *Transactions on Systems, Man and Cybernetics*, 21(1):83–89, 1991.

[87] H. Ehtamo and T. Raivio. On applied nonlinear and bilevel programming for pursuit-evasion games. *Optimization Theory and Applications*, 108(1):65–96, 2001.

[88] J.L. Emken, R. Benitez, A. Sideris, J.E. Bobrow, and D.J. Reinkensmeyer. Motor adaptation as a greedy optimization of error and effort. *Neurophysiology*, 97(6):3997–4006, 2007.

[89] S.E. Engelbrecht. Minimum principles in motor control. *Mathematical Psychology*, 45(3):497–542, 2001.

[90] J.E. Falk and J. Liu. On bilevel programming, part i: general nonlinear cases. *Mathematical Programming*, 70(1):47–72, 1995.

[91] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publisher, 1987.

[92] R. Featherstone. *Rigid Body Dynamics Algorithms*. Springer, 2008.

[93] A.G. Feldman and M.F. Levin. The origin and use of positional frames of reference in motor control. *Behavioral and Brain Sciences*, 18(4):723–744, 1995.

[94] W.O. Fenn and B.S. Marsh. Muscular force at different speeds of shortening. *Physiology London*, 85(3):277–297, 1935.

[95] F. Fisch. Development of a framework for the solution of high-fidelity trajectory optimization problems and bilevel optimal control problems. Dissertation, Lehrstuhl für Flugsystemdynamik, Technische Universität München, 2011.

[96] P.M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Experimental Psychology*, 47(6):381–391, 1954.

[97] J.R. Flanagan, E. Nakano, H. Imamizu, R. Osu, T. Yoshioka, and M. Kawato. Composition and decomposition of internal models in motor learning under altered kinematic and dynamic environments. *Neuroscience, Rapid Communications*, 19(RC34):1–5, 1999.

[98] J.R. Flanagan and D.J. Ostry. Trajectories of human multi-joint arm movements: evidence of joint level planning. In V. Hayward and O. Khatib, editors, *Experimental Robotics I*, volume 139 of *Lecture Notes in Control and Information Sciences*, pages 594–613. Springer, 1990.

[99] J.R. Flanagan and A.M. Wing. The role of internal models in motion planning and control: evidence from grip force adjustments during movements of hand-held loads. *Neuroscience*, 17(4):1519–1528, 1997.

[100] T. Flash. The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics*, 57(4):257–274, 1987.

[101] T. Flash and I. Gurevich. Models of motor adaptation and impedance control in human arm movements. *Advances in Psychology*, 119:423–481, 1997.

[102] T. Flash and E. Henis. Arm trajectory modifications during reaching towards visual targets. *Cognitive Neuroscience*, 3(3):220–230, 1991.

[103] T. Flash and B. Hochner. Motor primitives in vertebrates and invertebrates. *Current Opinion in Neurobiology*, 15(6):660–666, 2005.

[104] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Neuroscience*, 5(7):1688–1703, 1985.

[105] M.L. Flegel. Constraint qualifications and stationarity concepts for mathematical programs with equilibrium constraints. Dissertation, Institut für Mathematik, Julius-Maximilians-Universität Würzburg, 2005.

[106] M.L. Flegel and C. Kanzow. On the Guignard constraint qualification for mathematical programs with equilibrium constraints. *Optimization*, 54(6):517–534, 2005.

[107] R. Fletcher, N.I.M. Gould, S. Leyffer, P.L. Toint, and A. Wächter. Global convergence of trust-region SQP-filter algorithms for general nonlinear programming. *SIAM Journal on Optimization*, 13(3):635–659, 2002.

[108] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91(2):239–269, 2002.

[109] R. Fletcher and S. Leyffer. Numerical experience with solving mpecs as nlps. In *Technical Report NA/210, Department of Mathematics and Computer Science, University of Dundee, Dundee*, 2002.

[110] R. Fletcher and S. Leyffer. Solving mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software*, 19(1):15–40, 2004.

[111] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 17(1):259–286, 2007.

[112] R. Fletcher, S. Leyffer, and P.L. Toint. On the global convergence of an SLP-filter algorithm. *Technical Report 98/13, Département de Mathématique, Namur*, 1998.

[113] J. Fliege and L.N. Vicente. Multicriteria approach to bilevel optimization. *Optimization Theory and Applications*, 131(2):209–225, 2006.

[114] A. Fod, M.J. Matarić, and O.C. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, 2002.

[115] A. Forsgren, P.E. Gill, and M.H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44(4):525–597, 2002.

[116] D.W. Franklin, E. Burdet, K. Peng Tee, R. Osu, C.M. Chew, T.E. Milner, and M. Kawato. CNS learns stable, accurate, and efficient movements using a simple algorithm. *Neuroscience*, 28(44):11165–11173, 2008.

[117] D.W. Franklin, G. Liaw, T.E. Milner, R. Osu, E. Burdet, and M. Kawato. Endpoint stiffness of the arm is directionally tuned to instability in the environment. *Neuroscience*, 27(29):7705–7716, 2007.

[118] D.W. Franklin, R. Osu, E. Burdet, M. Kawato, and T.E. Milner. Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model. *Neurophysiology*, 90(5):3270–3282, 2003.

[119] G.E. Fruchter and P.R. Messinger. Optimal management of fringe entry over time. *Economic Dynamics and Control*, 28(3):445–466, 2003.

[120] M. Fukushima and G.-H. Lin. Smoothing methods for mathematical programs with equilibrium constraints. In *Proceedings of the International Conference on Informatics Research for Development of Knowledge Society Infrastructure*, pages 206–213, 2004.

[121] B.A. Garner and M.G. Pandy. The obstacle-set method for representing muscle paths in musculoskeletal models. *Computer Methods in Biomechanics and Biomedical Engineering*, 3(1):1–30, 2000.

[122] C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer, 2002.

[123] M. Gerdts. *Optimal Control of ODEs and DAEs*. de Gruyter, 2012.

[124] M. Gerdts, S. Karrenberg, B. Müller-Beßler, and G. Stock. Generating locally optimal trajectories for an automatically driven car. *Optimization and Engineering*, 10(4):439–463, 2009.

[125] P.E. Gill, W. Murray, and M.A. Saunders. Snopt: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[126] H. Gomi and M. Kawato. Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science*, 272(5258):117–120, 1996.

[127] H. Gomi and M. Kawato. Human arm stiffness and equilibrium-point trajectory during multijoint movement. *Biological Cybernetics*, 76(3):163–171, 1997.

[128] F.J. Gould and J.W. Tolle. A necessary and sufficient qualification for constrained optimization. *SIAM Journal on Applied Mathematics*, pages 164–172, 1971.

[129] P.L. Gribble, L.I. Mullin, N. Cothros, and A. Mattar. Role of cocontraction in arm movement accuracy. *Neurophysiology*, 89(5):2396–2405, 2003.

[130] P.L. Gribble and D.J. Ostry. Compensation for interaction torques during single-and multijoint limb movement. *Neurophysiology*, 82(5):2310–2326, 1999.

[131] A. Griewank and G.F. Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. Society for Industrial and Applied Mathematics (SIAM), 1991.

[132] D. Grimes, R. Chalodhorn, and R. Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Proceedings of Robotics: Science and Systems*, 2006.

[133] W. Grimm and A. Markl. Adjoint estimation from a direct multiple shooting method. *Optimization Theory and Applications*, 92(2):263–283, 1997.

[134] M. Guignard. Generalized Kuhn–Tucker conditions for mathematical programming problems in a banach space. *SIAM Journal on Control*, 7:232, 1969.

[135] E. Guigon, P. Baraduc, and M. Desmurget. Computational motor control: redundancy and invariance. *Neurophysiology*, 97(1):331–347, 2007.

[136] W.W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87(2):247–282, 2000.

[137] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations: Nonstiff Problems*, volume 1. Springer, 1993.

[138] A.F.C. Hamilton and D.M. Wolpert. Controlling the statistics of action: obstacle avoidance. *Neurophysiology*, 87(5):2434–2440, 2002.

[139] P. Hansen, B. Jaumard, and G. Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13:1194–1217, 1992.

[140] C.R. Hargraves and S.W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Guidance, Control, and Dynamics*, 10(4):338–342, 1087.

[141] C.M. Harris and D.M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394(6695):780–784, 1998.

[142] C.B. Hart and S.F. Giszter. Modular premotor drives and unit bursts as primitives for frog motor behaviors. *Neuroscience*, 24(22):5269–5282, 2004.

[143] R.S. Hartenberg and J. Denavit. *Kinematic Synthesis of Linkages*. McGraw-Hill, 1964.

[144] M. Haruno, D.M. Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural Computation*, 13(10):2201–2220, 2001.

[145] M. Haruno, D.M. Wolpert, and M. Kawato. Hierarchical mosaic for movement generation. In *Selected Topics of the International Symposium on Limbic and Association Cortical Systems*, volume 1250, pages 575–590, 2003.

[146] Z. Hasan. Optimized movement trajectories and joint stiffness in unperturbed, inertially loaded movements. *Biological Cybernetics*, 53(6):373–382, 1986.

[147] K. Hatz, J.P. Schlöder, and H.G. Bock. Estimating parameters in optimal control problems. *SIAM Journal on Scientific Computation*, 34(3):A1707–A1728, 2012.

[148] H. Hatze. A general myocybernetic control model of skeletal muscle. *Biological Cybernetics*, 28(3):143–157, 1978.

[149] H. Hatze. Neuromusculoskeletal control systems modeling–a critical survey of recent developments. *IEEE Transactions on Automatic Control*, 25(3):375–385, 1980.

[150] H. Hatze and J.D. Buys. Energy-optimal controls in the mammalian neuromuscular system. *Biological Cybernetics*, 27(1):9–20, 1977.

[151] B. Heißing and M. Ersoy. *Fahrwerkhandbuch: Grundlagen, Fahrdynamik, Komponenten, Systeme, Mechatronik, Perspektiven*. Vieweg+Teubner Verlag, 2007.

[152] F. Hermens and S. Gielen. Posture-based or trajectory-based movement planning: a comparison of direct and indirect pointing movements. *Experimental Brain Research*, 159(3):340–348, 2004.

[153] M. Hersch and A.G. Billard. Reaching with multi-referential dynamical systems. *Autonomous Robots*, 25(1):71–83, 2008.

[154] A.V. Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 126(843):136–195, 1938.

[155] B. Hoff. A model of duration in normal and perturbed reaching movement. *Biological Cybernetics*, 71(6):481–488, 1994.

[156] B. Hoff and M.A. Arbib. Models of trajectory formation and temporal interaction of reach and grasp. *Motor Behavior*, 25(3):175–192, 1993.

[157] N. Hogan. An organizing principle for a class of voluntary movements. *Neuroscience*, 4(11):2745–2754, 1984.

[158] N. Hogan. The mechanics of multi-joint posture and movement control. *Biological Cybernetics*, 52(5):315–331, 1985.

[159] N. Hogan and T. Flash. Moving gracefully: quantitative theories of motor coordination. *Trends in Neurosciences*, 10(4):170–174, 1987.

[160] T. Hoheisel, C. Kanzow, and A. Schwartz. Convergence of a local regularization approach for mathematical programmes with complementarity or vanishing constraints. *Optimization Methods and Software, to appear*, 2011.

[161] J.M. Hollerbach and T. Flash. Dynamic interactions between limb segments during planar arm movement. *Biological Cybernetics*, 44(1):67–77, 1982.

[162] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Proceedings of the 2002 Conference on Advances in Neural Information Processing Systems*, pages 1547–1554, 2003.

[163] R.B. Ivry and R. Spencer. The neural representation of time. *Current Opinion in Neurobiology*, 14(2):225–232, 2004.

[164] J. Izawa, T. Rane, O. Donchin, and R. Shadmehr. Motor adaptation as a process of reoptimization. *Neuroscience*, 28(11):2883–2891, 2008.

[165] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[166] D.H. Jacobson, M.M. Lele, and J.L. Speyer. New necessary conditions of optimality for control problems with state-variable inequality constraints. *Mathematical Analysis and Applications*, 35:255–284, 1971.

[167] A. Jain. Unified formulation of dynamics for serial rigid multibody systems. *Guidance, Control, and Dynamics*, 14(3):531–542, 1991.

[168] A. Jain and G. Rodriguez. Multibody mass matrix sensitivity analysis using spatial operators. *Multiscale Computational Engineering*, 1(2&3):219–234, 2003.

[169] R.G. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical programming*, 32(2):146–164, 1985.

[170] K.E. Jones, A.F.C. Hamilton, and D.M. Wolpert. Sources of signal-dependent noise during isometric force production. *Neurophysiology*, 88(3):1533–1544, 2002.

[171] T. Kashima and Y. Isurugi. Trajectory formation based on physiological characteristics of skeletal muscles. *Biological Cybernetics*, 78(6):413–422, 1998.

[172] T. Kashima, Y. Isurugi, and M. Shima. Analysis of a muscular control system in human movements. *Biological Cybernetics*, 82(2):123–131, 2000.

[173] M. Katayama and M. Kawato. Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models. *Biological Cybernetics*, 69(5):353–362, 1993.

[174] M. Kawato. Trajectory formation in arm movements: minimization principles and procedures. In H.N. Zelaznik, editor, *Advances in Motor Learning and Control*, pages 225–259. Human Kinetics, 1996.

[175] M. Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9(6):718–727, 1999.

[176] O. Khatib, L. Sentis, J. Park, and J. Warren. Whole body dynamic behavior and control of human-like robots. *Humanoid Robotics*, 1(1):29–43, 2004.

[177] M. Knauer. Bilevel-Optimalsteuerung mittels hybrider Lösungsmethoden am Beispiel eines deckengeführten Regalbediengerätes in einem Hochregallager. Dissertation, Fachbereich Mathematik und Informatik, Universität Bremen, 2009.

[178] M. Knauer and C. Büskens. Hybrid solution methods for bilevel optimal control problems with time dependent coupling. In M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels, editors, *Recent Advances in Optimization and its Applications in Engineering*, pages 237–246. Springer, 2010.

[179] K.P. Koerding and D.M. Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244–247, 2004.

[180] K.P. Koerding and D.M. Wolpert. The loss function of sensorimotor learning. *Proceedings of the National Academy of Sciences*, 101(26):9839–9842, 2004.

[181] K.P. Koerding and D.M. Wolpert. Bayesian decision theory in sensorimotor control. *Trends in Cognitive Sciences*, 10(7):319–326, 2006.

[182] Y. Koike and M. Kawato. Estimation of dynamic joint torques and trajectory formation from surface electromyography signals using a neural network model. *Biological Cybernetics*, 73(4):291–300, 1995.

[183] B. Koopman, H.J. Grootenboer, and H.J. de Jongh. An inverse dynamics model for the analysis, reconstruction and prediction of bipedal walking. *Biomechanics*, 28(11):1369–1376, 1995.

[184] D. Kraft. On converting optimal control problems into nonlinear programming problems. *NATO ASI Series*, F15(Computational Mathematical Programming):261–280, 1985.

[185] J.W. Krakauer, M.F. Ghilardi, and C. Ghez. Independent learning of internal models for kinematic and dynamic control of reaching. *Nature Neuroscience*, 2(11):1026–1031, 1999.

[186] S. Kraus. Fahrverhaltensanalyse zur Parametrierung situationsadaptiver Fahrzeugführungssysteme. Dissertation, Lehrstuhl für Fahrzeugtechnik, Technische Universität München, submitted 2012.

[187] S. Kraus, S. Albrecht, M. Sobotka, B. Heißing, and M. Ulbrich. Optimisation-based identification of situation determined cost functions for the implementation of a human-like driving style in an autonomous car. In *Proceedings of the International Symposium on Advanced Vehicle Control*, pages 412–417, 2010.

[188] I.L. Kurtzer, J.A. Pruszynski, and S.H. Scott. Long-latency reflexes of the human arm reflect an internal model of limb dynamics. *Current Biology*, 18(6):449–453, 2008.

[189] J.R. Lackner and P. Dizio. Rapid adaptation to Coriolis force perturbations of arm trajectory. *Neurophysiology*, 72(1):299, 1994.

[190] R.J. Leigh and D.S. Zee. *The Neurology of Eye Movements*. Oxford University Press, 1999.

[191] M.A. Lemay and P.E. Crago. A dynamic model for simulating movements of the elbow, forearm, and wrist. *Biomechanics*, 29(10):1319–1330, 1996.

[192] F.L. Lewis and V.L. Syrmos. *Optimal Control*. Wiley-Interscience, 1995.

[193] S. Leyffer. Complementarity constraints as nonlinear equations: Theory and numerical experience. In S. Dempe and V. Kalashnikov, editors, *Optimization with Multivalued Mappings*, pages 169–208. Springer, 2006.

[194] W. Li and E. Todorov. Iterative linear-quadratic regulator design for nonlinear biological movement systems. In *Proceedings of theInternational Conference on Informatics in Control, Automation, and Robotics*, pages 222–229, 2004.

[195] W. Li, E. Todorov, and X. Pan. Hierarchical optimal control of redundant biomechanical systems. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 4618–4621, 2005.

[196] D.G. Liebermann, A. Biess, J. Friedman, C.C.A.M. Gielen, and T. Flash. Intrinsic joint kinematic planning. I: Reassessing the Listing's law constraint in the control of three-dimensional arm movements. *Experimental Brain Research*, 171(2):139–154, 2006.

[197] D.G. Liebermann, A. Biess, C.C.A.M. Gielen, and T. Flash. Intrinsic joint kinematic planning. II: Hand-path predictions based on a Listing's plane constraint. *Experimental Brain Research*, 171(2):155–173, 2006.

[198] D. Liu and E. Todorov. Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. *Neuroscience*, 27(35):9354–9368, 2007.

[199] D. Liu and E. Todorov. Hierarchical optimal control of a 7-dof arm model. In *Proceeding of the Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 50–57, 2009.

[200] G.E. Loeb, I.E. Brown, and E.J. Cheng. A hierarchical foundation for models of sensorimotor control. *Experimental Brain Research*, 126(1):1–18, 1999.

[201] Z. Luo, M. Svinin, K. Ohta, T. Odashima, and S. Hosoe. On optimality of human arm movements. In *Proceedings of the 2004 International Conference on Robotics and Biomimetics*, pages 256–261, 2005.

[202] Z.Q. Luo, J.S. Pang, and D. Ralph. *Mathematical Programs With Equilibrium Constraints*. Cambridge University Press, 1996.

[203] K. Malanowski, C. Büskens, and H. Maurer. Convergence of approximations to nonlinear optimal control problems. In A.V. Fiacco, editor, *Mathematical programming with data perturbations*, volume 195 of *Lecture Notes in Pure and Applied Mathematics*, pages 253–284. Dekker, 1997.

[204] L.T. Maloney, J. Trommershäuser, M.S. Landy, and W. Gray. Questions without words: A comparison between decision making under risk and movement planning under risk. In W.D. Gray, editor, *Integrated Models of Cognitive Systems*, pages 297–313. Oxford University Press, 2007.

[205] M. Mann. *Benutzerorientierte Entwicklung und fahrergerechte Auslegung eines Querführungsassistenten*, volume 2. Cuvillier Verlag, 2008.

[206] U. Maoz, A. Berthoz, and T. Flash. Complex unconstrained three-dimensional hand movement and constant equi-affine speed. *Neurophysiology*, 101(2):1002–1015, 2009.

[207] N. Maratos. Exact penalty function algorithms for finite dimensional and control optimization problems. Dissertation, University of London, 1978.

[208] W. Maurel and D. Thalmann. A case study on human upper limb modelling for dynamic simulation. *Computer Methods in Biomechanics and Biomedical Engineering*, 2(1):65–82, 1999.

[209] P. Mazzoni, A. Hristova, and J.W. Krakauer. Why don't we move faster? Parkinson's disease, movement vigor, and implicit motivation. *Neuroscience*, 27(27):7105–7116, 2007.

[210] J. McIntyre, M. Zago, A. Berthoz, and F. Lacquaniti. Does the brain model Newton's laws? *Nature Neuroscience*, 4(7):693–694, 2001.

[211] B. Mehta and S. Schaal. Forward models in visuomotor control. *Neurophysiology*, 88(2):942–953, 2002.

[212] A. Migdalas, P.M. Pardalos, and P. Värbrand, editors. *Multilevel Optimization: Algorithms and Applications*. Kluwer Academic Publishers, 1998.

[213] A.E. Minetti and R. Alexander. A theory of metabolic costs for bipedal gaits. *Theoretical Biology*, 186(4):467–476, 1997.

[214] M. Mistry, P. Mohajerian, and S. Schaal. Arm movement experiments with joint space force fields using an exoskeleton robot. In *Proceedings of the International Conference on Rehabilitation Robotics*, pages 408–413, 2005.

[215] D. Mitrovic, S. Nagashima, S. Klanke, T. Matsubara, and S. Vijayakumar. Optimal feedback control for anthropomorphic manipulators. In *Proceedings of the International Conference on Robotics and Automation*, pages 4143–4150, 2010.

[216] A. Mitsos, B. Chachuat, and P.I. Barton. Towards global bilevel dynamic optimization. *Global Optimization*, 45(1):63–93, 2009.

[217] K. Mombaur, A. Truong, and J.P. Laumond. From human to humanoid locomotion: an inverse optimal control approach. *Autonomous Robots*, 28(3):369–383, 2010.

[218] P. Morasso. Spatial control of arm movements. *Experimental Brain Research*, 42(2):223–227, 1981.

[219] P. Morasso. Three dimensional arm trajectories. *Biological Cybernetics*, 48(3):187–194, 1983.

[220] D.D. Morrison, J.D. Riley, and J.F. Zancanaro. Multiple shooting method for two-point boundary value problems. *Communications of the ACM*, 5(12):613–614, 1962.

[221] F.A. Mussa-Ivaldi and E. Bizzi. Motor learning through the combination of primitives. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 355(1404):1755–1769, 2000.

[222] F.A. Mussa-Ivaldi and S.A. Solla. Neural primitives for motion control. *Oceanic Engineering*, 29(3):640–650, 2004.

[223] E. Nakano, H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka, and M. Kawato. Quantitative examinations of internal representations for arm trajectory planning: minimum commanded torque change model. *Neurophysiology*, 81(5):2140–2155, 1999.

[224] I.P. Natanson. *Theorie der Funktionen einer reellen Veränderlichen*. Verlag Harri Deutsch, 1975.

[225] W.L. Nelson. Physical principles for economies of skilled movements. *Biological Cybernetics*, 46(2):135–147, 1983.

[226] A.Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 663–670, 2000.

[227] P. Nie. Discrete time dynamic stackelberg games with the leaders in turn. *Nonlinear Analysis: Real World Applications*, 11(3):1685–1691, 2010.

[228] P. Nie, L. Chen, and M. Fukushima. Dynamic programming approach to discrete time dynamic feedback stackelberg games with independent and dependent followers. *European Journal of Operational Research*, 169(1):310–328, 2006.

[229] B.M. Nigg and W. Herzog. *Biomechanics of the Musculo-skeletal System*. Wiley, 2007.

[230] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[231] Y. Nubar and R. Contini. A minimal principle in biomechanics. *Bulletin of Mathematical Biology*, 23(4):377–391, 1961.

[232] K. Ohta, M.M. Svinin, Z.W. Luo, S. Hosoe, and R. Laboissiere. Optimal trajectory formation of constrained human arm reaching movements. *Biological Cybernetics*, 91(1):23–36, 2004.

[233] T. Okadome and M. Honda. Kinematic construction of the trajectory of sequential arm movements. *Biological Cybernetics*, 80(3):157–169, 1999.

[234] E.C.B. Olsen, S.E. Lee, W.W. Wierwille, and M.J. Goodman. Analysis of distribution, frequency, and duration of naturalistic lane changes. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pages 1789–1793, 2002.

[235] D.J. Ostry and A.G. Feldman. A critical evaluation of the force control hypothesis in motor control. *Experimental Brain Research*, 153(3):275–288, 2003.

[236] R. Osu, E. Burdet, D.W. Franklin, T.E. Milner, and M. Kawato. Different mechanisms involved in adaptation to stable and unstable dynamics. *Neurophysiology*, 90(5):3255–3269, 2003.

[237] R. Osu, D.W. Franklin, H. Kato, H. Gomi, K. Domen, T. Yoshioka, and M. Kawato. Short- and long-term changes in joint co-contraction associated with motor learning as revealed from surface EMG. *Neurophysiology*, 88(2):991–1004, 2002.

[238] R. Osu, N. Kamimura, H. Iwasaki, E. Nakano, C.M. Harris, Y. Wada, and M. Kawato. Optimal impedance control for task achievement in the presence of signal-dependent noise. *Neurophysiology*, 92(2):1199–1215, 2004.

[239] R. Osu, Y. Uno, Y. Koike, and M. Kawato. Possible explanations for trajectory curvature in multijoint arm movements. *Experimental Psychology: Human Perception and Performance*, 23(3):890–913, 1997.

[240] I. O'Sullivan, E. Burdet, and J. Diedrichsen. Dissociating variability and effort as determinants of coordination. *PLoS Computational Biology*, 5(4):e1000345(1–8), 2009.

[241] J.V. Outrata. On the numerical solution of a class of Stackelberg problems. *Mathematical Methods of Operations Research*, 34(4):255–277, 1990.

[242] J.V. Outrata, M. Kočvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications, and Numerical Results*. Kluwer Academic Publishers, 1998.

[243] H.B. Pacejka and E. Bakker. The magic formula tyre model. *Vehicle System Dynamics*, 21(Supplement 1):1–18, 1992.

[244] M.G. Pandy, B.A. Garner, and F.C. Anderson. Optimal control of non-ballistic muscular movements: a constraint-based performance criterion for rising from a chair. *Biomechanical Engineering*, 117(1):15–26, 1995.

[245] J.S. Pang and M. Fukushima. Complementarity constraint qualifications and simplified b-stationarity conditions for mathematical programs with equilibrium constraints. *Computational Optimization and Applications*, 13(1):111–136, 1999.

[246] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Proceedings of the International Conference on Robotics and Automation*, pages 763–768, 2009.

[247] B. Petreska and A.G. Billard. Movement curvature planning through force field internal models. *Biological Cybernetics*, 100(5):331–350, 2009.

[248] J. Pfefferer. Efficient recursive formulation of optimal control problems for industrial manipulators. Diplomarbeit, Zentrum Mathematik, Technische Universität München, 2007.

[249] R. Plamondon, A.M. Alimi, P. Yergeau, and F. Leclerc. Modelling velocity profiles of rapid movements: a comparative study. *Biological Cybernetics*, 69(2):119–128, 1993.

[250] N.S. Pollard, J.K. Hodgins, M.J. Riley, and C.G. Atkeson. Adapting human motion for the control of a humanoid robot. In *Proceedings of the International Conference on Robotics and Automation*, volume 2, pages 1390–1397, 2002.

[251] M.J.D. Powell. The Bobyqa algorithm for bound constrained optimization without derivatives. Cambridge DAMPT Report NA2009/06, University of Cambridge, 2009.

[252] R. Raikova. A general approach for modelling and mathematical investigation of the human upper limb. *Biomechanics*, 25(8):857–867, 1992.

[253] N. Raissi. Features of bioeconomics models for the optimal management of a fishery exploited by two different fleets. *Natural Resource Modeling*, 14(2):287–310, 2001.

[254] D. Ralph and S.J. Wright. Some properties of regularization and penalization schemes for MPECs. *Optimization Methods and Software*, 19(5):527–556, 2004.

[255] A.V. Rao, D.A. Benson, C. Darby, M.A. Patterson, C. Francolin, I. Sanders, and G.T. Huntington. Algorithm 902: Gpops, a matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *ACM Transactions on Mathematical Software*, 37(2):22, 2010.

[256] N.D. Ratliff, J.A. Bagnell, and M.A. Zinkevich. Maximum margin planning. In *Proceedings of the International Conference on Machine Learning*, pages 729–736, 2006.

[257] N.D. Ratliff, D. Silver, and J.A. Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

[258] S. Rauch. Analyse, Entwicklung und Implementierung von Regelalgorithmen zur Umsetzung menschlicher Fahrverhaltensweise beim autonomen Spurwechsel. Diplomarbeit, Lehrstuhl für Fahrzeugtechnik, Technische Universität München, 2008.

[259] M.J.E. Richardson and T. Flash. Comparing smooth arm movements with the two-thirds power law and the related segmented-control hypothesis. *Neuroscience*, 22(18):8201–8211, 2002.

[260] H.-J. Risse. Das Fahrverhalten bei normaler Fahrzeugführung. In *Fortschritt-Berichte VDI*, number 160 in Reihe 12. VDI-Verlag, 1991.

[261] D.A. Rosenbaum, R.G. Cohen, A.M. Dawson, S.A. Jax, R.G. Meulenbroek, R. van der Wel, and J. Vaughan. The posture-based motion planning framework: New findings related to object manipulation, moving around obstacles, moving in three spatial dimensions, and haptic tracking. *Progress in Motor Control*, 629(5):485–497, 2009.

[262] D.A. Rosenbaum, L.D. Loukopoulos, R.G.J. Meulenbroek, J. Vaughan, and S.E. Engelbrecht. Planning reaches by evaluating stored postures. *Psychological Review*, 102(1):28–66, 1995.

[263] D.A. Rosenbaum, R.J. Meulenbroek, J. Vaughan, and C. Jansen. Posture-based motion planning: Applications to grasping. *Psychological Review*, 108(4):709–734, 2001.

[264] D.E. Rosenthal. Triangularization of equations of motion for robotic systems. *Guidance, Control, and Dynamics*, 11(3):278–281, 1988.

[265] H.L. Royden. *Real Analysis*. Macmillian, 1968.

[266] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1987.

[267] P.N. Sabes. The planning and control of reaching movements. *Current Opinion in Neurobiology*, 10(6):740–746, 2000.

[268] A. Safonova, N.S. Pollard, and J.K. Hodgins. Optimizing human motion for the control of a humanoid robot. In *Proceedings of the International Conference on Robotics and Automation*, 2003.

[269] T. Sakamoto, N. Fukumura, and Y. Uno. Variability in human reaching movements depends on perception of targets. *IEIC Technical Report*, 102(730):19–24, 2003.

[270] T.D. Sanger. Failure of motor learning for large initial errors. *Neural Computation*, 16(9):1873–1886, 2004.

[271] J.A. Saunders and D.C. Knill. Humans use continuous visual feedback from the hand to control fast reaching movements. *Experimental Brain Research*, 152(3):341–352, 2003.

[272] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

[273] S. Schaal, A. Ijspeert, and A.G. Billard. Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):537, 2003.

[274] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert. Learning movement primitives. In P. Dario and R. Chatila, editors, *Robotics Research*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 561–572. Springer, 2005.

[275] S. Schaal and N. Schweighofer. Computational motor control in humans and robots. *Current Opinion in Neurobiology*, 15(6):675–682, 2005.

[276] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. *Mathematics of Operations Research*, 25(1):1–22, 2000.

[277] S. Scholtes. Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 11:918–936, 2001.

[278] A. Schwartz. Mathematical programs with complementarity constraints: Theory, methods, and applications. Dissertation, Institut für Mathematik, Julius-Maximilians-Universität Würzburg, 2011.

[279] R. Shadmehr and F.A. Mussa-Ivaldi. Adaptive representation of dynamics during learning of a motor task. *Neuroscience*, 14(5):3208–3224, 1994.

[280] Y.P. Shimansky, T. Kang, and J. He. A novel model of motor learning capable of developing an optimal movement control law online from scratch. *Biological Cybernetics*, 90(2):133–145, 2004.

[281] O. Sigaud and J. Peters. *From Motor Learning to Interaction Learning in Robots*. Springer, 2010.

[282] W.M. Silver. On the equivalence of lagrangian and newton-euler dynamics for manipulators. *Robotics Research*, 1(2):60–70, 1982.

[283] G. Simmons and Y. Demiris. Biologically inspired optimal robot arm control with signal-dependent noise. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 1, pages 491–496, 2004.

[284] G. Simmons and Y. Demiris. Optimal robot arm control using the minimum variance model. *Robotic Systems*, 22(11):677–690, 2005.

[285] J.F. Soechting, C.A. Buneo, U. Herrmann, and M. Flanders. Moving effortlessly in three dimensions: does Donders' law apply to arm movement? *Neuroscience*, 15(9):6271–6280, 1995.

[286] G.A. Sohl and J.E. Bobrow. A recursive multibody dynamics and sensitivity algorithm for branched kinematic chains. *Dynamic Systems, Measurement, and Control*, 123(3):391–399, 2001.

[287] J.L. Speyer and D.H. Jacobson. *Primer on Optimal Control Theory*. Society for Industrial and Applied Mathematics (SIAM), 2010.

[288] O. Sporns and G.M. Edelman. Solving Bernstein's problem: a proposal for the development of coordinated movement by selection. *Child Development*, 64(4):960–981, 1993.

[289] A. Sporrer, G. Prell, J. Buck, and S. Schaible. Realsimulation von Spurwechselvorgängen im Straßenverkehr. *Verkehrsunfall und Fahrzeugtechnik*, 36(3), 1998.

[290] S. Steffensen and M. Ulbrich. A new relaxation scheme for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 20(5):2504–2539, 2010.

[291] S. Stroeve. Impedance characteristics of a neuromusculoskeletal model of the human arm i. posture control. *Biological Cybernetics*, 81(5):475–494, 1999.

[292] S. Stroeve. Impedance characteristics of a neuromusculoskeletal model of the human arm ii. movement control. *Biological Cybernetics*, 81(5):495–504, 1999.

[293] S. Sueda, A. Kaufman, and D.K. Pai. Musculotendon simulation for hand animation. *ACM Transactions on Graphics*, 27(3):83, 2008.

[294] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[295] U. Syed, M. Bowling, and R.E. Schapire. Apprenticeship learning using linear programming. In *Proceedings of the International Conference on Machine Learning*, pages 1032–1039, 2008.

[296] U. Syed and R.E. Schapire. A game-theoretic approach to apprenticeship learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Proceedings of the 2007 Conference on Advances in Neural Information Processing Systems*, pages 1–8, 2008.

[297] C.D. Takahashi, R.A. Scheidt, and D.J. Reinkensmeyer. Impedance control and internal model formation when reaching in a randomly varying dynamical environment. *Neurophysiology*, 86(2):1047–1051, 2001.

[298] H. Tanaka, J.W. Krakauer, and N. Qian. An optimization principle for determining movement duration. *Neurophysiology*, 95(6):3875–3886, 2006.

[299] E. Theodorou and F.J. Valero-Cuevas. Optimality in neuromuscular systems. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 4510–4516, 2010.

[300] K.A. Thoroughman and R. Shadmehr. Learning of action through adaptive combination of motor primitives. *Nature*, 407(6805):742–747, 2000.

[301] M. Thuy, M. Goebl, F. Rattei, M. Althoff, F. Obermeier, S. Hawe, R. Nagel, S. Kraus, C. Wang, F. Hecker, M. Russ, M. Schweitzer, F.P. León, G. Färber, M. Buss, K. Diepold, J. Eberspächer, B. Heißing, and H.-J. Wünsche. Kognitive Automobile - Neue Konzepte und Ideen des Sonderforschungsbereiches/TR-28. In *Proceedings of the Conference on Aktive Sicherheit durch Fahrerassistenz*, 2008.

[302] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9):907–915, 2004.

[303] E. Todorov. Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural Computation*, 17(5):1084–1108, 2005.

[304] E. Todorov. Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences*, 106(28):11478–11483, 2009.

[305] E. Todorov and Z. Ghahramani. Unsupervised learning of sensory-motor primitives. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 1750–1753, 2004.

[306] E. Todorov and M.I. Jordan. Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. *Neurophysiology*, 80(2):696–714, 1998.

[307] E. Todorov and M.I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.

[308] E. Todorov, W. Li, and X. Pan. From task parameters to motor synergies: A hierarchical framework for approximately optimal control of redundant manipulators. *Robotic Systems*, 22(11):691–710, 2005.

[309] E.B. Torres and D. Zipser. Reaching to grasp with a multi-jointed arm. I. Computational model. *Neurophysiology*, 88(5):2355–2367, 2002.

[310] J. Trommershäuser, L.T. Maloney, and M.S. Landy. Statistical decision theory and the selection of rapid, goal-directed movements. *JOSA A (Optical Society of America)*, 20(7):1419–1433, 2003.

[311] J.L. Troutman. *Variational Calculus and Optimal Control*. Springer, 1996.

[312] M. Ulbrich and S. Ulbrich. *Nichtlineare Optimierung*. Birkhäuser, 2012.

[313] M. Ulbrich, S. Ulbrich, and L.N. Vicente. A globally convergent primal-dual interior-point filter method for nonlinear programming. *Mathematical Programming*, 100(2):379–410, 2004.

[314] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, 61(2):89–101, 1989.

[315] Y. Uno, R. Suzuki, and M. Kawato. Minimum muscle-tension-change model which reproduces human arm movement. In *Proceedings of the Symposium on Biological and Physiological Engineering*, pages 299–302, 1989.

[316] F.J. Valero-Cuevas, H. Hoffmann, M.U. Kurse, J.J. Kutch, and E.A. Theodorou. Computational models for neuromuscular function. *IEEE Reviews in Biomedical Engineering*, 2:110–135, 2009.

[317] F.J. Valero-Cuevas, M. Venkadesan, and E. Todorov. Structured variability of muscle activations supports the minimal intervention principle of motor control. *Neurophysiology*, 102(1):59–68, 2009.

[318] R.J. van Beers, P. Haggard, and D.M. Wolpert. The role of execution noise in movement variability. *Neurophysiology*, 91(2):1050–1063, 2004.

[319] B.M. van Bolhuis and C. Gielen. A comparison of models explaining muscle activation patterns for isometric contractions. *Biological Cybernetics*, 81(3):249–261, 1999.

[320] F.C.T. van der Helm. Analysis of the kinematic and dynamic behavior of the shoulder mechanism. *Biomechanics*, 27(5):527–550, 1994.

[321] F.C.T. van der Helm. A finite element musculoskeletal model of the shoulder mechanism. *Biomechanics*, 27(5):551–553, 1994.

[322] G.P. van Galen and W.P. de Jong. Fitts' law as the outcome of a dynamic noise filtering model of motor control. *Human Movement Science*, 14(4):539–571, 1995.

[323] F. Vanden Berghen and H. Bersini. Condor, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Computational and Applied Mathematics*, 181(1):157–175, 2005.

[324] S. Veelken. A new relaxation scheme for mathematical programs with equilibrium constraints: Theory and numerical experience. Dissertation, Fakultät für Mathematik, Technische Universität München, 2009.

[325] A.F. Vereshchagin. Computer simulation of the dynamics of complicated mechanisms of robot manipulators. *Engineering Cybernetics*, 6:65–70, 1974.

[326] L.N. Vicente and P.H. Calamai. Bilevel and multilevel programming: A bibliography review. *Global Optimization*, 5(3):291–306, 1994.

[327] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space. In *Proceedings of the International Conference on Machine Learning*, volume 1, pages 288–293, 2000.

[328] P. Viviani and T. Flash. Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. *Experimental Psychology*, 21:32–53, 1995.

[329] H. von Stackelberg. *Marktform und Gleichgewicht*. Springer, 1934.

[330] O. von Stryk. Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen. In *Fortschritt-Berichte VDI*, number 441 in Reihe 8. VDI-Verlag, 1995.

[331] A. Wächter and L.T. Biegler. Line search filter methods for nonlinear programming: Local convergence. *SIAM Journal on Optimization*, 16(1):32–48, 2006.

[332] A. Wächter and L.T. Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31, 2006.

[333] A. Wächter and L.T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[334] Y. Wada, Y. Kaneko, E. Nakano, R. Osu, and M. Kawato. Quantitative examinations for multi joint arm trajectory planning–using a robust calculation algorithm of the minimum commanded torque change trajectory. *Neural Networks*, 14(4):381–393, 2001.

[335] M.W. Walker and D.E. Orin. Efficient dynamic computer simulation of robotic mechanisms. *Dynamic Systems, Measurement, and Control*, 104:205–211, 1982.

[336] B. Weigl. Optimaltheoretische Modellierung und Analyse menschlichen Fahrverhaltens zur Regelung autonomer Fahrzeuge. Diplomarbeit, Lehrstuhl für Fahrzeugtechnik, Technische Universität München, 2010.

[337] J. Wiemeyer. Prinzipien und Merkmale gelungener Bewegungen. In T. Rossmann and C. Tropea, editors, *Bionik*, pages 561–574. Springer, 2005.

[338] D.A. Winter. *Biomechanics and Motor Control of Human Movement*. Wiley, 2004.

[339] J.M. Winters and L. Stark. Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models. *IEEE Transactions on Biomedical Engineering*, 32(10):826–839, 1985.

[340] J.M. Winters and L. Stark. Muscle models: What is gained and what is lost by varying model complexity. *Biological Cybernetics*, 55(6):403–420, 1987.

[341] D.M. Wolpert, K. Doya, and M. Kawato. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):593–602, 2003.

[342] D.M. Wolpert and J.R. Flanagan. Forward models. In T. Bayne, A. Cleeremans, and P. Wilken, editors, *The Oxford Companion to Consciousness*, pages 294–296. Oxford University Press, 2009.

[343] D.M. Wolpert and Z. Ghahramani. Computational principles of movement neuroscience. *Nature Neuroscience*, 3:1212–1217, 2000.

[344] D.M. Wolpert, Z. Ghahramani, and M.I. Jordan. Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study. *Experimental Brain Research*, 103(3):460–470, 1995.

[345] D.M. Wolpert, Z. Ghahramani, and M.I. Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882, 1995.

[346] D.M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7):1317–1329, 1998.

[347] D.M. Wolpert, R.C. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998.

[348] S.J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics (SIAM), 1997.

[349] S.W. Wu, J. Trommershäuser, L.T. Maloney, and M.S. Landy. Limits to human movement planning in tasks with asymmetric gain landscapes. *Vision*, 6(1), 2006.

[350] K. Yamane, Y. Ariki, and J. Hodgins. Animating non-humanoid characters with human motion data. In *Proceedings of the Symposium on Computer Animation*, pages 169–178, 2010.

[351] J.J. Ye. Necessary conditions for bilevel dynamic optimization problems. *SIAM Journal on Control and Optimization*, 33(4):1208–1223, 1995.

[352] J.J. Ye. Optimal strategies for bilevel dynamic problems. *SIAM Journal on Control and Optimization*, 35(2):512–531, 1997.

[353] J.J. Ye. Necessary and sufficient optimality conditions for mathematical programs with equilibrium constraints. *Mathematical Analysis and Applications*, 307(1):350–369, 2005.

[354] J.J. Ye and D.L. Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, 1995.

[355] F.E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*, 17(4):359–411, 1989.

[356] R. Zhang. Problems of hierarchical optimization: Nonsmoothness and analysis of solutions. Dissertation, University of Washington, Seattle, 1990.

[357] W. Zhang and D.A. Rosenbaum. Planning for manual positioning: the end-state comfort effect for manual abduction–adduction. *Experimental Brain Research*, 184(3):383–389, 2008.

[358] B.D. Ziebart, A. Maas, J.A. Bagnell, and A.K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence*, pages 1433–1438, 2008.