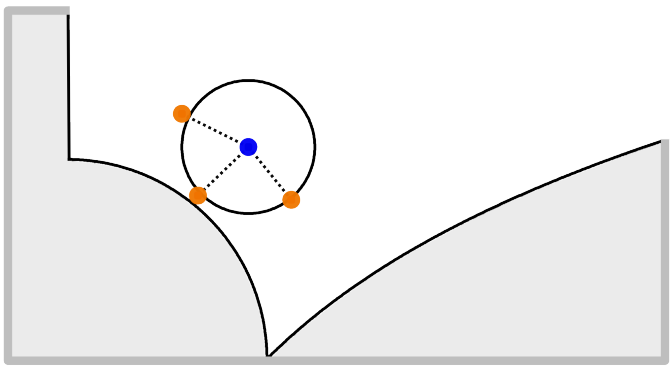
KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Electrical Engineering

Robust Optimization of Dynamic Systems

Boris Houska



Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Engineering Science

August 2011

Robust Optimization of Dynamic Systems

Boris Houska

Jury:
Prof. Dr. Paul Van Houtte, Chair
Prof. Dr. Moritz Diehl, Promotor
Prof. Dr. Joos Vandewalle
Prof. Dr. Stefan Vandewalle
Prof. Dr. Wim Michiels
Prof. Dr. Jan Swevers
Prof. Dr. Philippe Toint
(Université de Namur (FUNDP))
Prof. Dr. Aharon Ben-Tal
(Technion - Israel Institute of Technology)

Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Engineering Science

August 2011

© Katholieke Universiteit Leuven – Faculty of Engineering
Address, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2011/7515/96
ISBN: 978-94-6018-394-2

Abstract

This thesis is about robust optimization, a class of mathematical optimization problems which arise frequently in engineering applications, where unknown process parameters and unpredictable external influences are present. Especially, if the uncertainty enters via a nonlinear differential equation, the associated robust counterpart problems are challenging to solve. The aim of this thesis is to develop computationally tractable formulations together with efficient numerical algorithms for both: finite dimensional robust optimization as well as robust optimal control problems.

The first part of the thesis concentrates on robust counterpart formulations which lead to "min-max" or bilevel optimization problems. Here, the lower level maximization problem must be solved globally in order to guarantee robustness with respect to constraints. Concerning the upper level optimization problem, search routines for local minima are required. We discuss special cases in which this type of bilevel problems can be solved exactly as well as cases where suitable conservative approximation strategies have to be applied in order to obtain numerically tractable formulations. One main contribution of this thesis is the development of a tailored algorithm, the sequential convex bilevel programming method, which exploits the particular structure of nonlinear min-max optimization problems.

The second part of the thesis concentrates on the robust optimization of nonlinear dynamic systems. Here, the differential equation can be affected by both: unknown time-constant parameters as well as time-varying uncertainties. We discuss set-theoretic methods for uncertain optimal control problems which allow us to formulate robustness guarantees with respect to state constraints. Algorithmic strategies are developed which solve the corresponding robust optimal control problems in a conservative approximation. Moreover, the methods are extended to open-loop controlled periodic systems, where additional stability aspects have to be taken into account.

The third part is about the open-source optimal control software `ACADO` which is the basis for all numerical results in this thesis. After explaining the main algorithmic concepts and structure of this software, we elaborate on fast model predictive control implementations for small scale dynamic system as well as on an inexact sequential quadratic programming method for the optimization of large scale differential algebraic equations. Finally, the performance of the algorithms in `ACADO` is tested with robust optimization and robust optimal control problems which arise from various fields of engineering.

Acknowledgements

First of all, I thank my supervisor professor Moritz Diehl for the fruitful discussions and for inspiring many of the results, which are presented in this thesis. Besides his intensive and excellent mathematical advice, I also owe him many thanks for being a constant source of motivation and for creating a very enjoyable and international research environment at the Optimization in Engineering Center. I also thank Moritz Diehl for his unique talent to bring people from different communities and universities together, which gave me many chances to get in contact with researchers all over the world.

I owe many thanks to the members of my jury committee, professor Joos Vandewalle, professor Stefan Vandewalle, professor Wim Michiels, professor Jan Swevers, professor Philippe Toint, and professor Aharon Ben-Tal, who contributed with very constructive comments. Their professional advice helped a lot to improve the quality and technical correctness of this thesis.

I also thank professor Jinyan Fan for organizing the stay with her at the mathematics department of the Jiao Tong university in Shanghai, including many fruitful discussions on optimization algorithms. During this research stay, I got the possibility to attend a graduate colloquium in mathematics collecting inspirations from various fields and learning so many things about research and life in China.

Moreover, I want to thank Hans Joachim Ferreau for many discussions on programming and for implementing the optimal control software ACADO Toolkit together with me. Without this joint programming effort, the numerical results in this thesis would not have been possible. In addition, I thank Dr. Filip Logist, who contributed with many applications from the field of chemical engineering, which helped a lot to improve and debug the algorithms, which are used in this thesis. The cooperation with Filip Logist also led to joint publications in the field of multi-objective optimization.

I want to thank all my colleagues within my research group for the coffee and chocolate muffin support as well as many joint dinners at Alma. In particular, I thank Dr. Carlo Savorgnan and Quoc Tran-Dinh for the discussions – typically at lunch – which indirectly contributed to my thesis, too.

Finally, I want to thank my parents and family for the encouragement at home. My special thanks goes to my girlfriend, Lei Wang, the only person, who managed to get the work on my thesis from time to time completely out of my head. I thank her for her endless love, patience, and encouragement.

Contents

Acronyms and Notation **x**

1 Introduction **1**

1.1 Formulation of Robust Optimization Problems 2

1.2 Robust Optimal Control Problems 4

1.3 Existing Approaches for Robust Optimization 6

1.4 Contribution of the Thesis and Overview 13

I Robust Optimization **19**

2 Robust Convex Optimization **21**

2.1 The Convex Optimization Perspective 21

2.2 The S-Procedure for Quadratic Forms 28

2.3 Inner and Outer Ellipsoidal Approximation Methods 35

3 Robust Nonconvex Optimization **51**

3.1 Formulation of Semi-Infinite Optimization Problems 51

3.2 Convexification of Robust Counterparts 58

3.3 Necessary and Sufficient Optimality Conditions 70
3.4 Mathematical Programming with Complementarity Constraints 78

4 Sequential Algorithms for Robust Optimization 85
4.1 Tailored Sequential Quadratic Programming Methods 86
4.2 Sequential Convex Bilevel Programming 92
4.3 Local Convergence Analysis 98
4.4 Global Convergence Analysis 102
4.5 A Numerical Test Example 113

II Robust Optimal Control 117

5 The Propagation of Uncertainty in Dynamic Systems 119
5.1 Uncertain Nonlinear Dynamic Systems 119
5.2 Robust Positive Invariant Tubes for Linear Dynamic Systems 126
5.3 Uncertainty Propagation in Nonlinear Dynamic Systems 137

6 Robust Open-Loop Control 145
6.1 Robust Optimization of Open-Loop Controlled Systems 146
6.2 Interlude: Robust Optimal Control of a Tubular Reactor 153
6.3 Robust Optimization of Periodic Systems 160
6.4 Open-Loop Stable Orbits of an Inverted Spring Pendulum 168

III Software & Applications 173

7 ACADO Toolkit – Automatic Control and Dynamic Optimization 175

7.1 Introduction 175
7.2 Problem Classes Constituting the Scope of the Software 178
7.3 Software Modules and Algorithmic Features 181
7.4 Tutorial Examples and Numerical Tests 189

8 An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC 195
8.1 Introduction 195
8.2 The Real-Time Iteration Algorithm for Nonlinear Optimal Control 197
8.3 The ACADO Code Generation Tool 202
8.4 The Performance of the Auto-Generated NMPC Algorithm 207

9 A Quadratically Convergent Inexact SQP Method for DAE Systems 213
9.1 Introduction 213
9.2 Discretization of DAE Optimization Problems 215
9.3 Properties of the New Relaxation Function 221
9.4 Inexact SQP Methods for DAE Systems 228
9.5 Numerical Test Examples 235

10 Approximate Robust Optimization of a Biochemical Process 241
10.1 Introduction 241
10.2 Approximate Robust Optimization with Implicit Dependencies 242
10.3 Robustified Optimal Control for Periodic Processes 245
10.4 Periodic Optimal Control of a Biochemical Process 247
10.5 Robust Optimization of a Biochemical Process 251

11 Conclusions 255

11.1 An Interpretation of the Developed Robust Optimization Methods 255
11.2 Future Research Directions 260

Bibliography **261**

Curriculum Vitae **283**

Acronyms

ACADO	Automatic Control and Dynamic Optimization
AD	Automatic Differentiation
BDF	Backward Differentiation Formulas
BFGS	Broyden-Fletcher-Goldfarb-Shanno
DAE	Differential Algebraic Equation
ELICQ	Extended Linear Independence Constraint Qualification
GSIP	Generalized Semi-Infinite Programming
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
LMI	Linear Matrix Inequality
LP	Linear Programming
LQG	Linear-Quadratic-Gaussian (control)
MFCQ	Mangasarian-Fromovitz Constraint Qualification
MPC	Model Predictive Control
MPCC	Mathematical Programming with Complementarity Constraints
NCP	Nonlinear Complementarity Problem
NLP	Nonlinear Programming
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
QCQP	Quadratically Constrained Quadratic Programming
QP	Quadratic Programming
SCC	Strict Complementarity Condition
SCP	Sequential Convex Programming
SDP	Semi-Definite Programming
SIP	Semi-Infinite Programming
SOCP	Second Order Cone Programming
SOSC	Second Order Sufficient Condition
SQP	Sequential Quadratic Programming

Notation

Without recalling mathematical standard notation, we collect in the following list some remarks on the syntax in this thesis, which might be less common in some fields of mathematics and engineering:

- **Symmetric Matrices:** We use the notation $\mathbb{S}^n := \left\{ M \in \mathbb{R}^{n \times n} \mid M = M^T \right\}$ to denote the set of symmetric matrices. Similarly, $\mathbb{S}^n_+$ denotes set of symmetric matrices in $\mathbb{R}^{n \times n}$, which are positive semi-definite, while $\mathbb{S}^n_{++}$ denotes the set of positive definite $n \times n$ matrices.
- **Inequalities:** Besides the standard inequalities for scalars, we also write $a \leq b$ (or equivalently $b \geq a$), if $a, b \in \mathbb{R}^n$ are vectors, which satisfy $a_i \leq b_i$ for all components $i \in \{1, \ldots, n\}$. The corresponding strict versions "$<$" and "$>$" are analogously defined. For matrix inequalities, we always use the symbols $\preceq$ and $\succeq$, i.e., we write $A \preceq B$ (or equivalently $B \succeq A$) for symmetric matrices $A, B \in \mathbb{S}^n$, if $B - A \in \mathbb{S}^n_+$, and $A \prec B$ (or equivalently $B \succ A$), if $B - A \in \mathbb{S}^n_{++}$.
- **Sets and Operations with Sets:** For any set X, we use the syntax $\Pi(X)$ to denote the associated power set, i.e., the set of all subsets of X including the empty set. Moreover, for two sets $X, Y \subseteq \mathbb{R}^n$, we use the notation

  $$X + Y := \{ x + y \in \mathbb{R}^n \mid x \in X \text{ and } y \in Y \}$$

  to denote their Minkowski sum. Similarly, the definition of expressions like $\sum_{i=1}^m X_i$ for a set valued sequence $X_1, \ldots, X_m \subseteq \mathbb{R}^n$ is throughout this thesis always based on the Minkowski sum.
- **Ellipsoids:** There are many ways to notate ellipsoids. In this thesis, we will use the notation

  $$\mathcal{E}(Q, q) := \left\{ q + Q^{\frac{1}{2}} v \,\middle|\, \exists v \in \mathbb{R}^n : \; v^T v \leq 1 \right\} \subseteq \mathbb{R}^n .$$

to denote an ellipsoid with center $q \in \mathbb{R}^n$ and positive-semi definite matrix $Q \in \mathbb{S}^n_+$. Here, we will also use the short-hand $\mathcal{E}(Q) := \mathcal{E}(Q, 0)$ for centered ellipsoids. Note that the above definition is independent of the choice of the square-root $Q^{\frac{1}{2}}$ of the positive-semi definite matrix Q.

Chapter 1

Introduction

Nowadays, most of the processes which arise in engineering and industrial applications are optimized with respect to one or the other criterion. For example, we want to minimize time, we want to save energy and reduce emissions, or – especially in industrial production processes – we want to minimize costs while meeting given criteria like specifications on the quality of a product. In many of such cases, a blind application of optimization tools yields extreme solutions, which drive a process to its bounds without taking imperfections, model errors, or external uncertainties into account. As a consequence, a safe operation cannot be ensured and important constraints might be violated when unforeseen disturbances arise.

The aspect of safety in dynamic processes is especially important when human beings are involved in it and when it is crucial that hard constraints on the state of the system have to be satisfied for a whole ensemble of worst case scenarios. Here, we might think of cars, trains, airplanes, or other transportation technologies for which we might optimize the traveling time, robots which interact with humans, chemical processes which involve dangerous ingredients, or even nuclear power plants. In this context, the problem of guaranteeing safety is often two-sided: first, we do not have models which predict the behavior of the dynamic system with sufficient accuracy. In a typical situation, the model for the process of our interest is only validated by a finite number of noise-affected experiments and consequently the identified system parameters cannot be expected to be exactly known. And second, there might be external influences or disturbances – for example wind turbulences, temperature variations, structural imperfections, ground oscillations, weather changes etc. – which can usually not be predicted accurately, but

which might affect the corresponding dynamic processes in an unfortunate way. Thus, there arises the question of how we can optimize systems in such a way that we can still guarantee that given safety constraints are met for a reasonably chosen set of possible scenarios.

In this introduction, we discuss how optimization problems can mathematically be formulated, if we want to take uncertainties or disturbances into account. In this context, we should be aware of the fact that mathematical formulations of real-word phenomena are usually based on a set of assumptions or physical principles which appear natural and can approximately be validated by experiments. In the typical situation of robust optimization, we have to rely on assumptions on the uncertainty under which we can provide safety guarantees. In other words, a robustly optimized process can be just as unsafe as a nominally optimized one, if the "real" uncertainty does simply not satisfy our assumptions. Thus, the appropriate mathematical modeling of the uncertainties and disturbances can be as important as the modeling of the dynamic process itself.

We start with a general formulation of robust optimization problems, which is explained in Section 1.1. These consideration are extended for uncertain optimal control problems in Section 1.2, while Section 1.3 provides a literature overview. Section 1.4 outlines the structure and contribution of the thesis.

1.1 Formulation of Robust Optimization Problems

A standard optimization problem consists typically of a given continuous objective function $F_0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}$ and a compact set $\mathcal{F} \subseteq \mathbb{R}^{n_x}$ of feasible points. Here, our aim is to minimize the function F_0 over the variables which are in the set $\mathcal{F}$. In other words, we are interested in an optimization problem of the form

$$\min_{x \in \mathcal{F}} \; F_0(x, w) \; .$$

In this notation, F_0 can depend on a parameter or data vector $w \in \mathbb{R}^{n_w}$. If we know this parameter w exactly, there is so far nothing special about this optimization problem. However, if the parameter w is chosen by nature or by someone else who is playing against us, we might be in the situation that we do not know the exact value of w. Rather, we assume that our information about w is that this parameter is in a given compact set $W \subseteq \mathbb{R}^{n_w}$. In order to take this knowledge about the uncertainty w into account, we follow the classical concept of robust counterpart formulations, which has been established

by Ben-Tal and Nemirovski [17, 19]. Here, the assumption is that we want to minimize the worst possible value of the function F_0, i.e., we are interested in a min-max problem of the form

$$\min_{x \in \mathcal{F}} \max_{w \in W} \; F_0(x, w) \; .$$

This problem formulation can intuitively be motivated by interpreting the variable w as the optimization variable of an adverse player, who is trying to maximize the function F_0, while we are - as opposed to our adverse player - trying to minimize F_0. For most of the applications, we may assume that we have an explicit model for the set $\mathcal{F}$, which is given in form of continuous constraint functions $F_1, \ldots, F_m : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}$, such that the set $\mathcal{F}$ can be written as:

$$\mathcal{F} \;=\; \left\{ x \in \mathbb{R}^{n_x} \;\middle|\; \begin{array}{c} \max\limits_{w \in W} \; F_1(x, w) \;\leq\; 0 \\ \vdots \\ \max\limits_{w \in W} \; F_m(x, w) \;\leq\; 0 \end{array} \right\} .$$

Similar to the maximization of the objective value, we assume here that our counterpart player always chooses the worst possible value for the functions $F_1, \ldots, F_m$.

As an alternative to the above notation, we can also require the constraint functions F_i to be negative for all possible values of the uncertainty $w \in W$ and for all indices $i \in \{1, \ldots, m\}$. In other words, we can equivalently write the set $\mathcal{F}$ in the form

$$\mathcal{F} \;=\; \left\{ x \in \mathbb{R}^{n_x} \;\middle|\; \begin{array}{c} \forall w \in W : \;\; F_1(x, w) \;\leq\; 0 \\ \vdots \\ \forall w \in W : \;\; F_m(x, w) \;\leq\; 0 \end{array} \right\} .$$

In this notation, we do not have to solve global maximization problems to check whether a point x is feasible, but we have in general an infinite number of constraints. For this reason, robust optimization problems are sometimes also called semi-infinite optimization problems expressing that we have on the one hand infinitely many constraints, but on the other hand at least only a finite number of optimization variables.

The semi-infinite optimization perspective has sometimes advantages. For example, if we want to extend our notation to vector or matrix valued functions F_i in combination with generalized inequalities – which arise for example in the context of conic constraints – the semi-infinite point of view transfers in a natural way. However, in this thesis we will mainly

focus on scalar valued functions F_i and the standard ordering "$\leq$" in $\mathbb{R}$, for which the semi-infinite optimization perspective and the min-max formulation are entirely equivalent.

At this point, we should mention that the above way of formulating robust optimization problems is not the only option. We could also regard the case that w is a random variable with a given probability distribution. Especially, in applications where the uncertainty w is of a stochastic nature, it makes sense to regard chance constraints, i.e., constraints which have to be satisfied with a certain probability only. This can be important in applications, where the min-max formulation is too restrictive. However, the main focus of this thesis are the above outlined worst case formulations. Concerning the class of chance constrained optimization problems we only provide short remarks at one or the other place as well as a literature overview within Section 1.3.

1.2 Robust Optimal Control Problems

Optimal control problems are a special class of optimization problems which focus on the optimization of dynamic systems. A fairly general formulation of a nonlinear optimal control problem reads as follows:

$$\inf_{x(\cdot),u(\cdot),p,T_\mathrm{e}} \quad m(\,p, T_\mathrm{e}, x(T_\mathrm{e})\,)$$

$$\text{s.t.} \quad \left\{ \begin{array}{rcl} x(0) & = & x_0 \\ \dot{x}(\tau) & = & f(\tau, u(\tau), p, x(\tau), w(\tau)) \\ 0 & \geq & h(\tau, u(\tau), p, x(\tau), w(\tau)) \quad \text{for all } \tau \in [0, T_\mathrm{e}]\ . \end{array} \right. \tag{1.2.1}$$

Here, $T_\mathrm{e} \in \mathbb{R}_{++}$ denotes the duration of the dynamic process, $x : [0, T_\mathrm{e}] \to \mathbb{R}^{n_x}$ is a state vector, $u : [0, T_\mathrm{e}] \to \mathbb{R}^{n_u}$ a time varying control input, and $p \in \mathbb{R}^{n_p}$ a time constant parameter. Note that in contrast to the standard formulation of finite dimensional optimization problems, the above formulation of an optimal control problem requires the introduction of the function valued optimization variables x and u.

Besides the optimization variables x, u, p, and T_e, there are three model functions, denoted by f, h, and m, which are typically introduced within standard optimal control problem formulations: first, the possibly nonlinear right-hand side function or dynamic process model $f : \mathbb{R} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is needed to define the differential equation which has to be satisfied by the differential state x. Note that this right-hand side function

may in its first argument explicitly depend on the time τ. Besides the dependence of f on x, the dynamic equation can be influenced by the control input u, the parameter p, and an external input w. From a pure optimization perspective, the optimal control problem is simply defined to be infeasible whenever the differential equation does not admit a solution on the interval $[0, T_\mathrm{e}]$. However, within this thesis, we will typically require suitable Lipschitz conditions on the function f, such that we can rely on the unique existence of solutions of the associated differential equation.

Second, the function $h : \mathbb{R} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_h}$ can be used to formulate constraints on the states, controls, and parameters. And third, the objective function $m : \mathbb{R}^{n_p} \times \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}$ is in our formulation a Mayer term, which is evaluated at the end of the time horizon. Here, we note that additional integral terms (Lagrange terms) in the objective can always be reformulated into a Mayer term by introducing an additional differential state as a slack variable.

Similar to the considerations from the previous Section 1.1, where finite dimensional robust optimization problems have been introduced, we allow the optimal control problem to depend on a possibly time-varying input function $w : [0, T_\mathrm{e}] \to \mathbb{R}^{n_w}$ and a vector $x_0 \in \mathbb{R}^{n_x}$. If these two variables are exactly known and given, problem (1.2.1) is a standard optimal control problem. However, this thesis is about the case that the exact values of w and x are unknown. Here, we can either interpret w as a model error or as an external disturbance which can influence the behavior of the dynamic system, while x_0 is the initial value for the state. Our only knowledge about the input w and the initial value x_0 is of the form $(w, x_0) \in \mathcal{W}$, i.e., we assume that we have a given bounded set $\mathcal{W}$ for which we know that it contains the pair (w, x_0).

The idea of robust counterpart or min-max formulations transfers conceptionally also to optimal control problems. In order to outline this aspect, we assume for a moment that the solution of the differential equation uniquely exists, i.e., we assume that the state $x(t)$ can at any time t be interpreted as a function of the inputs x_0, u, p, and w, such that we may formally write

$$\forall t \in [0, T_\mathrm{e}] : \quad x(t) \;=\; \xi[t, x_0, u(\cdot), p, w(\cdot)] \;,$$

where the functional ξ can numerically be evaluated by integrating the differential equation on the interval $[0, t]$ using the corresponding arguments x_0, u, p, and w as initial value, control input, and disturbance input, respectively. With this notation, the

robust counterpart problem of the optimal control problem (1.2.1) can be written as

$$\begin{aligned} \inf_{x(\cdot),u(\cdot),p,T_{\mathrm{e}}} \quad & \sup_{(w(\cdot),x_0)\in\mathcal{W}} \quad \int_0^{T_{\mathrm{e}}} m(\, p, T_{\mathrm{e}}, \xi[T_{\mathrm{e}}, x_0, u(\cdot), p, w(\cdot)]\,) \\ \text{s.t.} \quad & \sup_{(w(\cdot),x_0)\in\mathcal{W}} \quad h_i(\tau, u(\tau), p, \xi[\tau, x_0, u(\cdot), p, w(\cdot)], w(\tau)) \;\leq\; 0\;, \end{aligned}$$

where the constraints have to be required for all times $\tau \in [0, T_{\mathrm{e}}]$ and all components $i \in \{1, \ldots, n_h\}$ of the constraint function h. If we would discretize the optimization variables as well as the constraint functions in the above inf-sup problem, we can in principle regard the corresponding discretized problem as a robust optimization problem with a finite number of variables such that the problem is reduced to the formulation from Section 1.1. However, in later chapters of this thesis, we shall see that robust optimal control problems have a particular structure which can be exploited by the formulation techniques and algorithms. Thus, we will usually treat this class of robust optimization problems separately.

1.3 Existing Approaches for Robust Optimization

Within the last decades, robust optimization has been a focus within many research communities starting with the field of control, convex optimization, mathematical programming, or even economics, and many fields of engineering science. Basically, whenever an optimization problem is formulated, the question arises whether really all parameters and inputs are exactly known and what changes if they are not. In this sense, it is not surprising that many researchers were and are attracted by the challenges of robust optimization.

Stochastic Programming

Starting with the work of Dantzig [63] on uncertain linear programming in the 1950s many articles in the field of stochastic programming occurred. The notion of chance constrained programming has been introduced by Charnes, Cooper, and Symonds in [55], Miller and Wagner [169], as well as by Prékopa [193]. Here, the main idea is to regard the uncertain parameter in the optimization problem as a random variable for which a given probability distribution is assumed. In the corresponding chance constrained formulation of a robust

optimization problem, the probability of a constraint violation is asked to be below a given confidence probability. For a more recent article on this topic and the relations to convex optimization, we refer to the work of Nemirovski and Shapiro [177, 178], which also provide a recommendable overview about this research field.

Classical Robust Control Theory

The historic origins of the rigorous worst-case robust optimization formulations can be found in the field of robust control. Here, the main motivation was to overcome the limitations of Kalman's linear quadratic control theory [34, 137], as LQG controllers were found to be non-robust with respect to uncertainties: Doyle published in [81] his classical article with the title "Guaranteed margins for LQG regulators", followed by the rather short abstract: "There are none." The development of the robust control theory was mainly influenced by Glover and Schweppe [102, 209], who analyzed linear control systems with set constrained disturbances, as well as by Zames [241], who was significantly contributing to the development of H_∞-control. For a more general overview on the achievements in classical robust control theory, including H_∞-control, we refer to the text books [83], [217] and [244] – as well as the references therein.

Convex Robust Optimization

An early article on robustness on convex optimization is by Soyster [219]. However, the main development phase of the robust counterpart methodology in convex optimization must be dated in the late 1990s. This phase was initialized and significantly driven by the work of Ben-Tal and Nemirovski [18, 19, 20] and also independently by the work of El-Ghaoui and Lebret [85]. These approaches are based on convex optimization techniques [46] and make intensive use of the concept of duality in convex programming, which helps us to transform an important class of min-max optimization problems into tractable convex optimization problems. Here, a commonly proposed assumption is that the uncertainty set is ellipsoidal (or an intersection of ellipsoidal sets), which is in many cases the key for working out robust counterpart formulations. For example, a linear program (LP) with uncertain data can be formulated as a second order cone program (SOCP), or an uncertain SOCP can – at least if the uncertainty set has a particularly structured ellipsoidal format – again be written as an SOCP. However, especially in the control context, polytopic uncertainty sets are also a common choice [14, 28]. Note that

the field of research addressing robust convex optimization problems has expanded during the last years and is still in progress, as reported in [16, 22]. Although these developments tend more and more towards approximation techniques, where the robust counterpart problem is replaced by more tractable formulations, they also cover an increasing amount of applications. For an extensive overview on robust optimization from the convex perspective, we refer to the recent text book by Ben-Tal, El-Ghaoui, and Nemirovski [17]. Finally, we refer to the work of Scherer [206] and the references therein, as well as to the work of Löfberg [156], where (modern) convex optimization techniques, especially linear matrix inequalities, in the context of robust control are exploited.

Nonconvex Robust Optimization

Looking at the non-convex case we can find some approaches in literature [71, 123, 133, 174] which suggest approximation techniques based on the assumption that w lies in a "small" uncertainty set W or equivalently that the curvatures of the objective function F_0 as well as the constraint functions $F_1, \ldots, F_m$ with respect to w are bounded by given constants such that the dependence of $F_0, F_1, \ldots, F_m$ can be described by a Taylor expansion where the second order term is over-estimated such that a conservative approximation is obtained. This linearization allows us in some cases to compute the maxima in an explicit way. As in the convex case, these approaches usually assume that the uncertainty sets are ellipsoidal (while the ellipsoids might however be nonlinearly parameterized in x) such that the sub maximization problems can easily be eliminated while the conservatively robustified minimization problem is solved with existing NLP algorithms. Note that Nagy and Braatz [174, 175] have established this approach. They also considered the case of more general polynomial chaos expansions, i.e., the case where higher order Taylor expansions with respect to the unknowns have to be regarded. However, in practice it is often already quite expensive to compute linearizations of the functions $F_0, F_1, \ldots, F_m$ with respect to the uncertainty - especially if we think of optimal control problems where such an evaluation requires us to solve nonlinear differential equations along with their associated variational differential equations. This cost might increase dramatically if higher order expansions have to be computed while the polynomial sub maximization problems can themselves only approximately be solved which requires again a level of conservatism. However, for the important special case that the constraint functions are polynomials in w, while the dimension n_w is small, there exists efficient robustification techniques which are based on positive polynomials and LMI-reformulations for which we

refer to the work of Lasserre [149], and the references therein, but also to the work of Parillo [184].

For the case that polynomial approximations of the problem functions with respect to the uncertainties are not acceptable, the completely nonlinear robust optimization problem must be considered. This completely nonlinear case has been studied in the mathematical literature in the context of semi-infinite programming. A recommendable overview article on this topic is by Hettich and Kortanek [119]. As mentioned above, the term "semi-infinite" arises from the observation that the constraints of an uncertainty have to be satisfied for all possible realizations of the variables w in the given uncertainty set $W(x)$, i.e., an infinite number of constraints must be regarded. Here, the problems in which the set W may depend on x are usually called generalized semi-infinite programming (GSIP) problems while the name semi-infinite programming (SIP) is reserved for the case that the uncertainty set W is constant. Within the last decades the growing interest in semi-infinite and generalized semi-infinite optimization yielded many results about the geometry of the feasible set, for which we refer to the work of Jongen [135], Rückmann [203], and Stein [220]. Moreover, first and second order optimality conditions for SIP and GSIP problems have been studied intensively [120, 135, 236]. However, when it comes to numerical algorithms, semi-infinite optimization problems turn out to be in their general form rather expensive to solve. Some authors have discussed discretization strategies for the uncertainty set in order to replace the infinite number of constraint by a finite approximation [119, 225, 226]. Although this approach works acceptably for very small dimensions n_w, the curse of dimensionality hurts for $n_w \gg 1$ such that discretization strategies are in this case rather conceptual. Note that the situation is very different if additional concavity assumptions are available. Indeed, as semi-infinite optimization problems can under mild assumptions [221] be regarded as a Stackelberg game [214], the lower level maximization problems can - in the case of concavity - equivalently be replaced by their first order optimality conditions, which leads to a mathematical program with complementarity constraints (MPCC). In this context, we also note that semi-infinite optimization problems can be regarded as a special bilevel optimization problem [13]. However, as we shall see this in this thesis, semi-infinite programming problems should not be treated as if they were a general bilevel optimization problem as important structure is lost otherwise.

Being at this point, semi-infinite optimization problems give rise to convexification methods with the aim to equivalently replace or to conservatively approximate the lower level maximization problems with a concave optimization problem. As discussed above, one way to obtain a convexification is linearization. However, in the field of global optimization

more general Lagrangian underestimation (or, for maximization problems, overestimation) techniques are a well-known tool [212, 213, 231] for convexification which is often used as a starting point for the development of branch-and-bound algorithms. In the context of generalized semi-infinite programming such a concave overestimation technique has been suggested by Floudas and Stein [100] to deal with the problem of finding the global solution of the lower level maximization problems discussing the case where the uncertainty is assumed to be in a given one-dimensional interval. The corresponding technique is called α-relaxation and works in principle also for uncertainties with dimension $n_w > 1$ which are bounded by a box. For $n_w \gg 1$ the α-relaxation can be used as a conservative approximation while the authors in [100] suggest for the case of small n_w to combine this α-overestimation with a branch-and-bound technique (α-BB method) which converges to the exact solution.

Classical Optimal Control Theory

Concerning the field of optimization in (open-loop) control it should be mentioned first that there exists a huge amount of articles on general nonlinear optimal control problems. In this thesis we will not provide an overview of all of them, but discuss some selected articles which had a significant influence. Early articles on optimal control are from the 1960s by Pontryagin [189] as well as by Bryson and Ho [49, 50], who analyzed optimality conditions for optimal control problems. The work of Pontryagin has lead to the so called indirect approach, which is based on the concept "*first optimize, then discretize*", i.e., we first apply Pontryagin's optimality principle and then we discretize the corresponding continuous time constrained boundary value problem in order to apply numerical techniques. However, modern optimal control techniques are typically based on direct methods, which have for example been introduced by Sargent and Sullivan [204]. In contrast to the indirect methods, the direct approaches discretize the dynamic system first approximating the continuous time optimal control problem with a discrete, finite dimensional nonlinear programming problem which can then be solved numerically. Thus, the concept of direct methods can be summarized as "*first discretize, then optimize*". Modern optimal control software is usually based on direct methods. Here, two main approaches exist: the first approach is based on direct collocation, for which we refer to the work of Cuthrell and Biegler [29, 30, 62]. And the second approach is based on single- or multiple shooting methods, for which we refer to the work of Bock and Plitt [37, 38, 187] as well as Bock and Leineweber [40, 150]. For an overview text on practical methods in optimal control, the book by Betts [26] might also be helpful. Note that there exist many software

implementations of standard algorithms for nonlinear optimal control problems, for which we refer at this point only to [125, 152, 232]. However, in Appendix 7, in particular within Section 7.1, we provide a complete overview of existing optimal control tools, including an overview of recent software developments.

Robust Open-Loop Optimal Control

Let us now proceed with a review of existing approaches on robust optimal control, i.e., the robust optimization of dynamic systems. In order to avoid confusion at this point, we should clearly point out that, we have to distinguish two situations which are both contained in the name "*robust control*": the first case is based on the assumptions that we can only control the system in open-loop mode, where we assume that we do not have any possibility to react to disturbances once the process is started. While, in the second case, we know that we will have measurements such that we can react to future disturbances online. Starting with the open-loop case there are some approaches available [71, 174, 175], which have been applied to nonlinear dynamic system, but are rather based on heuristic than providing mathematical robustness guarantees. In contrast, for robust open-loop control of linear dynamic systems more approaches exist. In this context, we highlight once more the work of Schweppe [209]. Moreover, Kurzhanski, Valyi, and Varaiya [144, 145, 146] contributed significantly with their analysis of ellipsoidal methods for linear dynamic systems. In addition, most of the approaches for the robust optimization of closed loop controlled systems transfer naturally also to the robust optimization of open-loop controlled systems.

Note that an important sub-problem of robust optimal control is to analyze the influence or propagation of uncertainty in dynamic systems. This type of analysis is also known under the name reachability analysis for dynamic systems as for example elaborated by Kurzhanski and Varaiya [146] or Lygeros, Tomlin, and Sastry [164]. In this context, we can find mature literature on set theoretic methods including Aubin's viability theory [12] and Isaacs' differential games [134]. Concerning modern numerical techniques for the computation of reachable sets with high numerical precision we refer to the work of Mitchell, Bayen, and Tomlin [170] and the references therin, where the computational techniques are inspired from the field of partial differential equations. Here, the main idea is to analyze viscosity solutions of Hamilton-Jacobi-Isaacs equations [86].

Periodic Systems and Stability Optimization

Periodic optimal control problems are a special class of optimization problems for dynamic systems which are considered on an infinite time-horizon assuming that we are interested in periodic trajectories. For these periodic systems, we are besides the robustness with respect to constraints also interested in the question whether the system is stable. Starting with Lyapunov's original work [163] which appeared at the beginning of the 20th century, the question of the existence and stability of periodic orbits has lead to many contributions in this field. For example, at the end of the 20th century, Matthieu and Hill have analyzed an interesting class of differential equations, the Matthieu-Hill differential equations, for which it can be proven that non-trivial open-loop stable periodic orbits exist [238] and which can be seen as an important prototype class of problems for which nontrivial open-loop stable orbits can be observed.

In general, it is extremely difficult to analyze the periodic orbits of a nonlinear dynamic system. For example Hilbert's 16th problem (published in 1900) is asking for the number and configuration of the periodic limit cycles of a general polynomial vector field in the plane. In fact, this problem is up to now still unsolved [155] and must be considered as one of the hardest problems ever posed in mathematics. This illustrates how difficult the analysis of such periodic cycles can be – and here we talk about a dynamic system with two differential states only. On the other hand, in practical applications, we have often at least a rough idea or physical intuition of when and where periodic cycles can be expected. Here, we can think of periodically driven spring-damper systems, periodic thermodynamic Carnot processes, bicycles, humanoid and walking robots, controllable kites, many periodically operating power generating devices, etc. Thus the question how to find and optimize the stability of periodic orbits numerically is highly relevant and, of course, this question has also been addressed by many authors.

Starting with the work of Kalman [138] and Bittanti [34] periodic Lyapunov and Ricatti equations became an important field of research for analyzing the stability of linear periodic systems. Some of the existing modern robust stability optimization techniques are based on the optimization of the so called pseudo-spectral abscissa. In this context we refer to the work of Burke, Lewis, Overton, and Henrion [53, 52] as well as to the work of Trefethen and Embree [228]. In these approaches non-smooth (but derivative based) optimization algorithms are developed. Similar approaches have been proposed in [229] and [78], where a smoothed version of the spectral abscissa is optimized such that existing derivative based, local optimal control techniques can be employed. For interesting applications of open-loop stability optimization in the field of robotics we refer to the work of Mombaur [171].

Robust Closed-Loop Control

From a nonlinear optimization perspective, the difference between open-loop and closed-loop controlled systems is not significant, as we may for example assume a linear or affine parameterization of the control law such that the closed loop problem can in principle be cast as a robust open-loop optimal control problem. However, the resulting robust optimization problems are typically non-convex – even if the system is jointly affine in the state and the control input. Such affine feedback parameterization have for example been analyzed by Ben-Tal and Nemirovski [17] in the context of so called affinely adjustable robust counterparts. For the optimization of linear feedback laws, we also refer to approaches of Apkarian and Noll [9] as well as to [129]. In this context, it should also be noted that a linear feedback parameterization can be sub-optimal – especially if control constraints are present. Complementing the classical robust control theory, which has already been reviewed above, most of the modern approaches on robust closed-loop control can be found in the model predictive control theory. In this context, we refer to the extensive research in this field, most prominently driven by the fundamental work of Rawlings [197], the min-max model predictive control techniques of Kerrigan and Maciejowski [139, 140], the affine disturbance-feedback parameterization approach by Kerrigan, Goulart, and Maciejowski [105], as well as the work of Langson and Chryssochoos [147], Mayne [167], and Rakovic [195, 194] on tube based model predictive control, a technique, which has originally been pioneered by Bertsekas and Rhodes [25, 24]. These approaches are typically based on set propagation techniques, where usually exact state feedback as well as constraints on both the disturbances and the controls are given. Moreover, there exist min-max model predictive control schemes based on robust dynamic programming, which have been developed my Björnberg and Diehl [70]. For similar approximate dynamic programming strategies in the context of stochastic control we refer to the work of Wang and Boyd [235]. Finally, for robust control techniques based on invariant sets, we refer to work of Blanchini [35] and Kolmanovski and Gilbert [142], as well as to a very recommendable book on set theoretic methods in control by Blanchini and Miani [36].

1.4 Contribution of the Thesis and Overview

This thesis is divided into three parts, named: *Robust Optimization*, *Robust Optimal Control*, and *Software & Applications*.

Outline of Part I: Robust Optimization

The goal of Part I, *Robust Optimization*, is to develop a consistent framework for the formulation, tractable approximation, and numerical solution of nonlinear min-max problems, which arise in the context of general robust optimization problems. The contribution is splitted into three chapters which are based on each other:

- Chapter 2 is about selected, for the most part existing results in convex robust optimization. This chapter is not designed to be encyclopedic, but mainly to recall the main concepts and calculus in convex robust optimization which are needed in order to understand the contributions of this thesis. It introduces the concept of Lagrangian duality, including a review of the S-procedure, which is frequently used to reformulate or approximate min-max problems with tractable standard minimization problems. Moreover, ellipsoidal based set approximation strategies are discussed which are in later chapters employed for the robust optimization of dynamic systems. Although the results in this chapter are not new, they are presented from a perspective which cannot be found in existing text books on convex or robust optimization. In addition, some of the examples and derivations are original ideas of this thesis.

- Chapter 3 is about the formulation and approximation of non-convex robust optimization problems. Here, a Lagrangian overestimation technique is developed, which is needed to obtain tractable, lower level convex approximations of nonlinear min-max optimization problems. We illustrate this approximation technique for robust counterpart problems with examples, prove that the presented strategy is superior to existing Taylor expansion based approximation methods, and discuss special cases in which this approximation is exact. Moreover, first order necessary and second order sufficient conditions for general semi-infinite programming problems are reviewed. In this context, we point out several structural properties of min-max problems and the relation to mathematical programs with complementarity constraints. The corresponding technical results are the basis of the sequential convex bilevel programming algorithm.

- Chapter 4 is about numerical algorithms for nonlinear robust optimization problems. We first discuss the advantages and disadvantages of applying existing sequential quadratic programming algorithms to nonlinear min-max optimization problems. The main part of this chapter is about a sequential convex bilevel programming

algorithm which exploits the structure of nonlinear min-max problems more efficiently than existing techniques. This is one of the main contributions of this thesis. We motivate the algorithm and discuss implementation details as well as local and global convergence results. The algorithm is also applied to a numerical test example.

Outline of Part II: Robust Optimal Control

The goal of Part II, *Robust Optimal Control*, is to review and extend set theoretic methods which allow first to assess and compute the influence of uncertainty in dynamic systems, and second, to formulate and solve optimal control problems taking the uncertainty into account. Here periodic systems and stability optimization problems are regarded, too.

- Chapter 5 is about uncertainty propagation in dynamic systems. After discussing several options to model uncertainty sets for possibly time-varying unknown inputs and time constant parameters in nonlinear dynamic systems, we introduce the notation of robust positive invariant tubes. The proposed computational methods for approximating the tubes, in which the state of an uncertain dynamic system is known to be, are based on parameterized ellipsoids. In this context, we first review and extend existing ellipsoidal methods for the computation of robust positive invariant tubes for uncertain linear dynamic systems. However, one main contribution of this chapter is that we also generalize these computational techniques for nonlinear dynamic systems aiming at numerically tractable ways for approximating the propagation of uncertainty in a conservative way.
- Chapter 6 is about robust optimization of open-loop controlled dynamic systems, one of the core topics and highlights of this thesis. Here, we discuss how to formulate robust nonlinear optimal control problems and how to solve them in a conservative approximation. The corresponding techniques are applied to a robust optimal control problem for a nonlinear jacketed tubular reactor. Inside this reactor a highly nonlinear and uncertain exothermic chemical reaction takes place while there are hard safety constraints on the temperature which must be satisfied for all possible scenarios. Moreover, we extend our framework for periodic systems, too, where additional open-loop stability requirements have to be met. The corresponding stability optimization techniques are demonstrated at an open-loop controlled inverted spring pendulum, which is stabilized without needing any feedback.

Outline of Part III: Software & Applications

The goal of Part III, *Software & Applications*, is to explain the concept of the optimal control software ACADO Toolkit which is the basis for all the numerical computations in this thesis. Here, we first provide an overview of the toolkit in general and then elaborate on three main algorithmic features: ultra-fast nonlinear model predictive control algorithms for small scale systems, efficient exploitation of structure and automatic differentiation for the optimization of large scale systems comprising differential algebraic equations, and efficient robust optimal control formulations and algorithms.

- Chapter 7 is about the open-source software ACADO, which has been developed as part of a joint development effort in collaboration with my colleague Hans Joachim Ferreau. In ACADO Toolkit direct methods for optimal control, in particular multiple-shooting based sequential quadratic programming algorithms, are implemented. In this context, we highlight in particular ACADO's unique capability to deal with symbolic expressions in optimal control problems, which allows us to use automatic differentiation, code export, and automatic structure detection. Note that this chapter has been accepted as a journal publication [131].
- Chapter 8 is about an extension of ACADO, which enables automatic code generation for model predictive control algorithms. The algorithm itself is based on a real-time Gauss-Newton method which is designed for fast nonlinear model predictive control algorithms. The main contribution of this tool is its efficiency: we demonstrate that for a nonlinear dynamic systems with four states and a control horizon of ten samples, sampling times of much less than a millisecond are possible. Note that this chapter has been accepted for publication and will appear in Automatica [132].
- Chapter 9 is about a quadratically convergent inexact SQP method which has been designed for optimal control problems which comprise differential algebraic equations (DAEs). While the code export techniques from Chapter 8 illustrate how to implement fast algorithms for small-scale systems, the tailored inexact SQP algorithm is designed for large scale systems with many algebraic states. The corresponding algorithm is implemented in ACADO and we demonstrate its efficiency by optimizing a distillation column with 82 differential and 122 algebraic states. The chapter is based on a journal publication which is currently under review [128].
- Chapter 10 is about an application of an approximate robust optimization technique, which is designed to robustly optimize periodic stationary states of dynamic systems.

Here, the application is a periodic biochemical process with uncertain system parameters. The algorithm itself is based on adjoint differentiation techniques, which are especially efficient if the dynamic system is affected by many uncertainties while only a few constraints have to be satisfied in a robust way. Note that this chapter has successfully been published in [133].

Note that the chapters in Part III are all composed from publications which have already been accepted or are currently under review as outlined above. In addition, large parts of the results in Chapter 5 and 6 have appeared in [124, 125, 129], while the contributions from Chapters 3 and 4 are submitted and currently under review [127]. Finally, the work on `ACADO Toolkit` has also led to joint publications [90, 91, 157, 158, 159, 160] which are, however, not part of this thesis.

Part I

Robust Optimization

Chapter 2

Robust Convex Optimization

2.1 The Convex Optimization Perspective

Let us start with an introduction to robust optimization problems from an convex optimization perspective. For this aim, we regard functions $F_1, \ldots, F_m : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}$ and define an associated feasible set $\mathcal{F} \subseteq \mathbb{R}^{n_x}$ of the form

$$\mathcal{F} \;:=\; \left\{ x \in \mathbb{R}^{n_x} \;\middle|\; \begin{array}{rcl} \forall w \in W: \quad F_1(x,w) & \leq & 0 \\ \vdots & & \\ \forall w \in W: \quad F_m(x,w) & \leq & 0 \end{array} \right\}.$$

In this context, x denotes a variable which we can choose, while $w \in W$ is a variable which our adverse player can choose assuming that the uncertainty set $W \subseteq \mathbb{R}^{n_w}$ is given. In other words, the feasible set $\mathcal{F}$ can be interpreted as the set of all x for which we can guarantee that the functions $F_1, \ldots, F_m$ do all take negative values no matter how the uncertainty $w \in W$ is realized. A general robust optimization problem can now be written as

$$\min_x \max_w \; F_0(x,w) \quad \text{s.t.} \quad x \in \mathcal{F}\,, \tag{2.1.1}$$

where $F_0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}$ is a given objective function.

Note that the above definition of the set $\mathcal{F}$ requires us in general to evaluate infinitely many constraints. Only for the special case that the uncertainty set W contains a finite number of points, this problem can directly be transformed into a standard mathematical

program with a finite number of constraints. For this reason, problems of the form (2.1.1) are called semi-infinite optimization problems.

Instead of formulating infinitely many constraints, an alternative is to evaluate the constraints only at global uncertainty maximizers. For this aim, we assume first that the functions F_i are continuous while the set W is compact such that we can define lower level robust counterpart functions $V_i : \mathbb{R}^{n_x} \to \mathbb{R}$ by

$$\forall x \in \mathbb{R}^n : \qquad V_i(x) \;=\; \max_{w \in W} \; F_i(x,w) \qquad \text{with } \; i \in \{0,\ldots,m\} \; . \tag{2.1.2}$$

Using this notation, the optimization problem (2.1.1) can equivalently be written as an optimization problem of the form

$$\min_x \; V_0(x) \quad \text{s.t.} \quad V_i(x) \leq 0 \quad \text{for all } \; i \in \{1,\ldots,m\} \; . \tag{2.1.3}$$

The difficulty of the above robust counterpart problem is two-sided: first, we need to solve parameterized maximization problems in order to evaluate the functions V_i and second, we need to solve a minimization problem to find the robust minimizer x^* of the upper-level problem (2.1.3). Due to this specific bi-level structure, robust counterpart problems of the form (2.1.3) are also called min-max problems.

Clearly, if we succeed in working out explicit expressions for the functions V_i, the problem (2.1.3) reduces to a standard minimization problem. Unfortunately, it is only in a very limited amount of cases possible to work out such explicit expressions. On the other hand, there are some "simple" but relevant cases where we can succeed in deriving explicit expressions. Thus, we start our consideration of robust optimization problems by collecting some of these cases. As most of these cases are based on ellipsoidal uncertainty sets, we first introduce the following notation:

Definition 2.1 (Ellipsoid): *We associate with each positive-semi definite matrix $Q \in \mathbb{S}^n_+$ and any vector $q \in \mathbb{R}^n$ an ellipsoid $\mathcal{E}(Q,q) \subseteq \mathbb{R}^n$. This ellipsoid is defined as*

$$\mathcal{E}(Q,q) \;=\; \left\{ \, q + Q^{\frac{1}{2}} v \,\middle|\, \exists v \in \mathbb{R}^n : \;\; v^T v \;\leq\; 1 \right\} \; . \tag{2.1.4}$$

Depending on the context, we will also use the short-hand $\mathcal{E}(Q) := \mathcal{E}(Q,0)$, whenever we are interested in ellipsoids which are centered at the origin.

Now, we consider the following special cases of robust optimization in which it is possible to work out the robust counterpart functions V_i explicitly. In Example 2.1 we concentrate

on how to exploit the tight version of the Cauchy-Schwarz inequality for that purpose, while Examples 2.2 and 2.3 employ the tight version of the triangle-inequality for Euclidean norms.

Example 2.1: Let the functions F_i be uncertainty affine such that we have

$$F_i(x,w) \;=\; c_i(x)^T w + d_i(x)$$

for some functions $c_i : \mathbb{R}^{n_x} \to \mathbb{R}^{n_w}$ and $d_i : \mathbb{R}^{n_x} \to \mathbb{R}$, while the set $W \;:=\; \mathcal{E}(Q,q)$ is an ellipsoid with $Q \in \mathbb{S}_+^{n_w}$ and $q \in \mathbb{R}^{n_w}$. Then we can find explicit expressions for the worst case functions V_i which can be written as

$$V_i(x) \;=\; \max_{w \in \mathcal{E}(Q,q)} c_i(x)^T w + d_i(x) \;\;=\; \sqrt{c_i(x)^T Q c_i(x)} + c_i(x)^T q + d_i(x)\;.$$

Thus, in this special case, the associated robust counterpart problem reduces to a standard minimization problem of the form

$$\begin{array}{ll} \min\limits_{x} & \left\| Q^{\frac{1}{2}} c_0(x) \right\|_2 + c_0(x)^T q + d_0(x) \\ \text{s.t.} & \left\| Q^{\frac{1}{2}} c_i(x) \right\|_2 + c_i(x)^T q + d_i(x) \;\leq\; 0 \quad \text{for all} \quad i \in \{1,\ldots,m\}\;. \end{array}$$

Moreover, if the functions c_i and d_i are all affine in x, the above optimization problem is a convex second order cone programming (SOCP) problem.

Example 2.2 (Robust Least-Squares Optimization): Let us consider the case that the function F_i is a term of the following form

$$F_i(x,w) \;:=\; \| \, (A+\Delta)x \, \|_2 \;-\; d$$

assuming that the data matrix $A \in \mathbb{R}^{m \times n}$ and the scalar offset $d \in \mathbb{R}$ are given while the matrix $\Delta \in \mathbb{R}^{m \times n}$ is unknown, i.e., the uncertainty vector can be written as $w := \mathrm{vec}(\Delta)$. For the case that the uncertainty set is ellipsoidal, we may - after suitable scaling - assume that

$$W \;\;:=\;\; \{\; \Delta \mid \|\Delta\|_{\mathrm{F}} \;\leq\; 1 \,\}\;.$$

In order to compute the associated robust counterpart function, we employ the triangle inequality

$$\| \, (A+\Delta)x \, \|_2 \;\leq\; \| \, Ax \, \|_2 + \| \, \Delta x \, \|_2 \;\leq\; \| \, Ax \, \|_2 + \| \, x \, \|_2 \;.$$

Note that we can always construct a $\Delta^* \in W$ such that the above inequality is tight. One way to check this is by choosing

$$\Delta^* := \frac{A\,xx^T}{\|Ax\|\,\|x\|} .$$

In other words, we have found an explicit expression for the robust counterpart function

$$V_i(x) = \max_{\Delta \in W} \| (A+\Delta)x \|_2 - d = \| Ax \|_2 + \| x \|_2 - d .$$

Note that the above consideration has applications in robust estimation. For example, if we apply the triangle inequality with $A := \left(\hat{A}, b\right)$, $\Delta := \left(\hat{\Delta},\, \delta\right)$ and $x := \left(y^T,\, 1\right)^T$ we obtain

$$\min_y \max_{\|\Delta\|_{\mathrm{F}}^2 + \|\delta\|_2^2 \le 1} \left\|(\hat{A}+\hat{\Delta})y + (b+\delta)\right\|_2 = \min_y \left\|\hat{A}y + b\right\|_2 + \sqrt{\|y\|_2^2 + 1} ,$$

which can be interpreted as the robust counterpart formulation of an uncertain least-squares optimization problem. El-Ghaoui and Lebret have worked out several generalizations of this result for which we refer to [85].

Example 2.3: Let us regard a generalization of Example 2.2 for functions of the form

$$F_i(x,w) := \| (A+\Delta)x \|_2 - (c+\delta)^T x ,$$

where the matrix $A \in \mathbb{R}^{m\times n}$ and the vector $c \in \mathbb{R}^n$ are given while the matrix $\Delta \in \mathbb{R}^{m\times n}$ and the vector $\delta \in \mathbb{R}^n$ are unknown. For the case that Δ and δ are known to be bounded by independent ellipsoids, we may - after suitable scaling - assume that the uncertainty set has the form

$$W = \{\, (\Delta,\delta) \mid \|\Delta\|_{\mathrm{F}} \le 1 \text{ and } \|\delta\|_2 \le 1 \,\}$$

Combining the results from the previous two examples we easily find an explicit expression for the robust counterpart function

$$V_i(x) := \max_{(\Delta,\delta)\in W} \| (A+\Delta)x \|_2 - (c+\delta)^T x = \| Ax \|_2 - c^T x + 2\, \| x \|_2 .$$

As the above inequality for x can easily be transformed into a second order cone constraint using slack variables, a SOCP with uncertain data bounded by two independent ellipsoids, is again an SOCP. Ben-Tal and Nemirovski have worked out several generalization of this result for which we refer to [17, 19, 22]. Note that the same triangle-inequality trick can be transferred also to LPs, QPs, or QCQPs with uncertain data, as they can all be written as SOCPs.

In the special cases from the examples above we learn that it is sometimes possible to find explicit expressions for the robust counterpart functions V_i. In order to extend the class of problems for which such explicit strategies are possible, we review some more systematic concepts. Here, we follow the classical framework of Ben-Tal, Nemirovski, and El-Ghaoui [17] employing duality techniques, which are known from the field of convex optimization and which help us to reformulate "min-max" problems explicitly into "min-min" problems. For this aim, we first define what we understand under lower level convexity:

Definition 2.2 (Lower Level Convexity): *We say that an optimization problem of the form* (2.1.3) *is lower level convex if the uncertainty set W is convex, while the functions $F_i(x,\cdot) : W \to \mathbb{R}$ are for all indices $i \in \{1,\ldots,m\}$ and for all $x \in \mathcal{F}$ concave functions in w.*

In the following, we assume that we have a given component-wise convex constraint function $B : \mathbb{R}^{n_w} \to \mathbb{R}^{n_B}$ such that the uncertainty set W can be written as

$$W \;=\; \{\; w \in \mathbb{R}^{n_w} \;\mid\; B(w) \;\leq\; 0 \;\}\;.$$

The main strategy can now be outlined as follows: if the robust counterpart problem is lower level convex while the uncertainty set W has a non-empty interior (Slater's constraint qualification), we can express the functions V_i equivalently via their dual problem:

$$V_i(x) \;=\; \inf_{\lambda_i > 0} D_i(x,\lambda_i)\;.$$

Here, the dual functions $D_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_B}_+ \to \mathbb{R}$ are for all $i \in \{0,\ldots,m\}$ defined as

$$D_i(x,\lambda_i) \;:=\; \max_w F_i(x,w) - \lambda_i^T B(w)\;.$$

In some special cases, it is possible to work out explicit expressions for the Lagrange dual functions D_i. In such a situation, we can augment the upper level optimization variable x by the dual optimization variables $\lambda \;:=\; (\,\lambda_0\,\ldots,\lambda_m\,)$, i.e., the original "min-max" problem (2.1.3) can be re-formulated into an equivalent "min-min" problem of the form

$$\inf_{x,\lambda>0}\; D_0(x,\lambda_0) \quad \text{s.t.} \quad D_i(x,\lambda_i) \;\leq\; 0\;.$$

One of the most important prototype cases where the above strategy is applicable is discussed within the following linear programming example:

Example 2.4: Let the functions F_i be uncertainty affine such that we have

$$F_i(x,w) \;=\; c_i(x)^T w + d_i(x)\;.$$

Moreover, we assume that the uncertainty set is a polytope of the form

$$W \;:=\; \{\, w \mid Aw \;\leq\; b \,\} \tag{2.1.5}$$

for some matrix $A \in \mathbb{R}^{n_B \times n_w}$ and some vector $b \in \mathbb{R}^{n_B}$. In this case, it is difficult to find an explicit expression for the worst case functions V_i, but we can express the objective value of the maximization problem as a minimization problem by using dual linear programming:

$$\begin{aligned} V_i(x) \;&=\; \max_{w} \;\; c_i(x)^T w + d_i(x) \qquad \text{s.t.} \qquad Aw \;\leq\; b \\ &=\; \min_{\lambda_i \geq 0} \;\; b^T\lambda_i + d_i(x) \qquad\quad \text{s.t.} \qquad A^T\lambda_i \;=\; c_i(x)\;. \end{aligned}$$

Thus, the robust counterpart problem can be reduced to a standard minimization problem of the form

$$\begin{aligned} \min_{x,\lambda_0,\ldots,\lambda_m} \quad & b^T\lambda_0 + d_0(x) \\ \text{s.t.} \quad & 0 \;\geq\; b^T\lambda_i + d_i(x) \\ & 0 \;\leq\; \lambda_i \\ & 0 \;=\; A^T\lambda_i - c_i(x) \qquad \text{for all} \quad i \in \{1,\ldots,m\}\;. \end{aligned} \tag{2.1.6}$$

Moreover, for the case that the functions c_i and d_i are itself affine in x the above optimization problem is a convex linear programming problem.

Remark 2.1: *The above example generalizes almost one-to-one to the case that the functions F_i are affine in w, as above, but the uncertainty set is defined via semi-definite inequalities, i.e.,*

$$W \;:=\; \left\{\, w \;\middle|\; \sum_{j=1}^{n_w} A_j w_j \;\preceq\; B \,\right\}\;,$$

where $A_1,\ldots,A_m,B \in \mathbb{R}^{n_B \times n_B}$ are given matrices. In this case the robust counterpart functions are of the form

$$\begin{aligned} V_i(x) \;&=\; \max_{w \in W} \;\; c_i(x)^T w + d_i(x) \\ &=\; \min_{\Lambda_i \succeq 0} \;\; \mathrm{Tr}\left(B^T\Lambda_i\right) + d_i(x) \quad \text{s.t.} \quad \mathrm{Tr}\left(A_j^T\Lambda_i\right) \;=\; c_{i,j}(x) \end{aligned}$$

with $j \in \{1, \ldots, n_w\}$.

It is an important observation that we always have higher level convexity of the upper level problem if the functions F_i are convex in x. This result is independent of how the uncertainty w enters.

Definition 2.3 (Upper Level Convexity): *We say that the robust optimization problem of the form* (2.1.3) *is upper level convex if the associated robust counterpart functions* $V_i(x) : \mathcal{F} \to \mathbb{R}$ *are for all indices* $i \in \{0, \ldots, m\}$ *convex functions.*

Lemma 2.1 (A Sufficient Condition for Upper Level Convexity): *If the parameterized functions* $F_i(\cdot, w)$ *are for all* $w \in W$ *and for all* $i \in \{0, \ldots, m\}$ *convex functions in* x *then the robust optimization problem of the form* (2.1.3) *is upper level convex.*

Proof: We can use that the maximum over convex functions is convex. □

Note that the dual functions D_i are by construction always convex in λ and also jointly convex in (x, λ), as long as the functions F_i are convex in x.

The above dual reformulation strategy as explained so far has the disadvantage that it is based on the assumption that we can work out the dual function D_i explicitly, which is not always possible or can at least become inconvenient. However, we shall see later that the numerical strategies for robust convex and non-convex optimization which we will develop in Chapter 3 avoid this problem by avoiding to construct the dual Lagrange function explicitly. Another important remark is that the convexity condition on the functions F_i with respect to x, as required by Lemma 2.1, is only sufficient but by no means necessary for upper level convexity. In order to illustrate this aspect, we consider the following example:

Example 2.5: Let us consider the unconstrained scalar min-max problem

$$\min_x \max_w F_0(x, w) \qquad \text{with} \quad F_0(x, w) := -x^2 + bxw - w^2 \tag{2.1.7}$$

for some constant $b \geq 2$. The function F_0 is for no fixed w convex in x. Nevertheless, the upper level problem turns out to be convex as the associated robust counterpart function $V_0(x) = -x^2 + \frac{1}{4}(bx)^2$ is convex for $b \geq 2$. This example outlines the fact that a robust optimization problem can in some cases be "easier" to solve than any of its associated nominal optimization problems with fixed uncertainties, as robustification leads sometimes to a convexification.

In the following Section 2.2 we will discuss some more advanced strategies which can help us to exactly reformulate or conservatively approximate robust counterpart problems.

2.2 The S-Procedure for Quadratic Forms

In this section, we briefly review the concept of Lagrangian relaxation methods for quadratic forms. The corresponding technique is historically known under the name S-procedure [101, 117, 240] which must be considered as one of the basic tools in robust optimization. In particular, the S-procedure is frequently used in the field of robust linear system theory [230]. For a recommendable and more recent overview article on the S-procedure, we also refer to [188].

The basic idea is very simple and can be outlined as follows: let us regard a possibly non-convex quadratically constrained quadratic programming problem of the form

$$V \;:=\; \max_x \; x^T H_0 x + g_0^T x + s_0 \qquad \text{s.t.} \quad x^T H_i x + g_i^T x + s_i \;\leq\; 0 \qquad (2.2.1)$$

with $i \in \{1, \ldots, m\}$ and for some symmetric matrices $H_i \in \mathbb{S}^{n_x}$, some vectors $g_i \in \mathbb{R}^{n_x}$, and scalars $s_i \in \mathbb{R}$. In the following we will assume that the above QCQP is strictly feasible. Let us introduce the affine functions

$$H(\lambda) \;:=\; H_0 - \sum_{i=1}^m \lambda_i H_i \;, \quad g(\lambda) \;:=\; g_0 - \sum_{i=1}^m \lambda_i g_i \;, \quad \text{and} \quad s(\lambda) \;:=\; s_0 - \sum_{i=1}^m \lambda_i s_i \;.$$

Using this notation, we can write the dual of the quadratically constrained quadratic programming problem as

$$\begin{aligned} \hat{V} \;&:=\; \inf_{\lambda > 0} \; \max_x \; x^T H(\lambda) x \;+\; g(\lambda)^T x \;+\; s(\lambda) \\ &\\ &=\; \inf_{\lambda > 0} \; \frac{1}{4} g(\lambda)^T H(\lambda)^{-1} g(\lambda) \;+\; s(\lambda) \quad \text{s.t.} \quad H(\lambda) \;\prec\; 0 \;. \end{aligned}$$

Finally, we employ the Schur complement formula to rewrite $\hat{V}$ as the solution of a semi-definite programming problem of the form

$$\hat{V} \;:=\; \min_{\lambda \geq 0, \gamma} \; \gamma \quad \text{s.t.} \quad \begin{pmatrix} s(\lambda) \;-\; \gamma & \frac{1}{2} g(\lambda)^T \\ \frac{1}{2} g(\lambda) & H(\lambda) \end{pmatrix} \;\preceq\; 0 \qquad (2.2.2)$$

One way to summarize the S-procedure for quadratic forms is the following:

Lemma 2.2 (S-Lemma): *The optimal value $\hat{V}$ of the convex semi-definite programming problem* (2.2.2) *is an upper bound on the optimal value V of the original quadratically constrained quadratic program.*

Remark 2.2: *For special classes of quadratically constrained quadratic programming problems explicit bounds on the sub-optimality of the approximation $\hat{V}$ are known. Originally such bounds have been analyzed in the context of the Maximum Cut problem [103]. For more general sub-optimality estimates we also refer to [117, 176, 179]. In addition, there exists a tight version of the S-Lemma which will be discussed below.*

Let us illustrate a few applications of the S-Lemma within the following examples:

Example 2.6: Let us consider a quadratic programming problem (QP) with symmetric constraints of the form

$$V \;:=\; \max_x \; x^T H_0 x + g_0^T x \quad \text{s.t.} \quad -b \;\leq\; Ax \;\leq\; b\,.$$

If we square the constraints and write them in the form $x^T \left(a_i \, a_i^T \right) x \;\leq\; b_i^2$ with a_i^T being the i-th row of the matrix A, the problem can be regarded as a QCQP. With

$$H(\lambda) \;:=\; H_0 + A^T \Lambda A\,, \quad g(\lambda) \;:=\; g_0\,, \quad s(\lambda) \;=\; b^T \Lambda b\,, \quad \text{and} \quad \Lambda \;:=\; \mathrm{diag}(\lambda)$$

the SDP (2.2.2) yields a global upper bound $\hat{V}$ on the optimal value V of the above possibly non-convex QP with symmetric constraints.

Example 2.7: Let us come back to the discussion of uncertain optimization problems from the previous section. We consider the case that the functions $F_i(x,w)$ are quadratic forms in the uncertainty w with

$$F_i(x,w) \;\;=\;\; w^T H_i(x) w + g_i(x)^T w\,.$$

Moreover, we assume that the uncertainty set is an intersection of ellipsoids, i.e.,

$$W \;:=\; \bigcap_{j \in \{1,\ldots,N\}} \mathcal{E}(Q_j, q_j) \qquad \left(\text{with } \; Q_j \in \mathbb{S}^{n_w}_{++} \; \text{ and } \; q_j \in \mathbb{R}^{n_w} \right)\,.$$

Now, our aim is to find conservative approximations for the worst case functions

$$V_i(x) \;:=\; \max_{w \in W} F_i(x,w)\,.$$

Applying the above S-Lemma, we find that the functions

$$\hat{V}_i(x) \;:=\; \min_{\lambda_i \geq 0\,,\,\gamma_i} \quad \gamma_i \quad \text{s.t.} \quad \left(\begin{array}{cc} s_i(x,\lambda_i) \;-\; \gamma_i & \frac{1}{2} g_i(x,\lambda_i)^T \\ \frac{1}{2} g_i(x,\lambda_i) & H_i(x,\lambda_i) \end{array} \right) \;\preceq\; 0$$

are upper bounds on the functions V_i, i.e., we have $\hat{V}_i(x) \geq V_i(x)$ for all $x \in \mathbb{R}^{n_x}$. Here, we use the notation

$$H_i(x,\lambda_i) \;:=\; H_i(x) - \sum_{j=1}^{N} \lambda_{i,j} Q_j^{-1} \quad , \quad g_i(x,\lambda_i) \;:=\; g_i(x) + \sum_{j=1}^{N} 2\lambda_{i,j} Q_j^{-1} q_j \quad ,$$

$$\text{and} \quad s_i(x,\lambda_i) \;:=\; \sum_{j=1}^{N} 2\lambda_{i,j} \left(1 \;-\; q_j^T Q_j^{-1} q_j \right) .$$

Consequently, we can construct a conservative robust counterpart problem of the form

$$\min_x \;\; \hat{V}_0(x) \quad \text{s.t.} \quad \hat{V}_i(x) \;\leq\; 0 \quad \text{with} \quad i \in \{1,\ldots,m\} \,,$$

which can equivalently be written as

$$\min_{x,\gamma,\lambda_0,\ldots,\lambda_m} \quad \gamma_0 \quad \text{s.t.} \quad \left\{ \begin{array}{l} \forall i \in \{1,\ldots,m\}: \quad 0 \;\geq\; \gamma_i \quad , \quad 0 \;\leq\; \lambda_i \,, \\ \\ 0 \;\succeq\; \left(\begin{array}{cc} s_i(x,\lambda_i) \;-\; \gamma_i & \frac{1}{2} g_i(x,\lambda_i)^T \\ \frac{1}{2} g_i(x,\lambda_i) & H_i(x,\lambda_i) \end{array} \right) . \end{array} \right. \tag{2.2.3}$$

In particular, for the case that the functions H_0 and g_0 are affine in x the above approximate robust counterpart problem is a semi-definite programming problem.

As mentioned, there exist different formulations and derivations of the S-Lemma. The S-Lemma can be interpreted as a tool for convexification and in this thesis we want to understand this aspect from all its perspectives such that we can later use these convexification properties in the context of robust optimization. Let us convexify the quadratically constrained quadratic program (2.2.1) by using that the optimization problem

$$V \quad := \quad \max_{x,X} \;\; \mathrm{Tr}(H_0 X) + g_0^T x + s_0 \qquad \text{s.t.} \quad \left\{ \begin{array}{l} 0 \;\geq\; \mathrm{Tr}(H_i X) + g_i^T x + s_i \\ X \;=\; x x^T \end{array} \right. \tag{2.2.4}$$

(with $i \in \{1,\ldots,m\}$) is equivalent to the original QCQP (2.2.1). In this problem all the "non-convexity" is collected in the constraint $X \;=\; x x^T$ on the auxiliary matrix $X \in \mathbb{S}_+^{n_x}$.

Thus, if we simply relax this equality constraint, the objective value can only get bigger. In other words, if we define

$$\widetilde{V} \;:=\; \max_{x,X} \;\mathrm{Tr}(H_0 X) + g_0^T x + s_0 \qquad \text{s.t.} \quad \left\{ \begin{array}{l} 0 \;\geq\; \mathrm{Tr}(H_i X) + g_i^T x + s_i \\ 0 \;\preceq\; \left(\begin{array}{cc} 1 & x^T \\ x & X \end{array} \right) \end{array} \right. \tag{2.2.5}$$

with $i \in \{1, \ldots, m\}$, then we have $\widetilde{V} \;\geq\; V$. This relaxation method and the S-procedure turn out to be equivalent as the semi-definite programming problems (2.2.2) and (2.2.5) are dual to each other, i.e., both convexification strategies yield the same lower bound $\widetilde{V} \;=\; \hat{V}$ on the objective value V.

Motivated by the above considerations, we work out yet a third way to formulate the S-Lemma looking for a second order sufficient global optimality condition for non-convex quadratically constrained quadratic programs:

Lemma 2.3 (A Sufficient Condition for Global Optimality): *Let (x^*, λ^*) be a primal-dual KKT point of the QCQP* (2.2.1) *at which Slater's constraint qualification is satisfied. If the associated Hessian matrix is negative semi-definite, i.e., if we have*

$$H(\lambda^*) \;=\; H_0 - \sum_{i=1}^{m} \lambda_i^* H_i \;\preceq\; 0 \,,$$

then (x^*, λ^*) is a global maximizer of the QCQP (2.2.1)*. Moreover, if the above negative semi-definiteness condition is satisfied, then the inequality in the S-Lemma is tight, i.e., we have $V \;=\; \hat{V}$.*

Proof: Let (x^*, λ^*) be a primal-dual KKT point which satisfies the above negative semi-definiteness condition such that we can define the matrices $X^* \;:=\; x^* \, (x^*)^T$ as well as $Z^* \;:=\; -H_0 + \sum_{i=1}^m \lambda_i^* H_i \;\succeq\; 0$. Now, we note that the convex maximization problem (2.2.5) can also be written as

$$\widetilde{V} \;:=\; \max_{x,X} \;\mathrm{Tr}(H_0 X) + g_0^T x + s_0 \qquad \text{s.t.} \quad \left\{ \begin{array}{l} 0 \;\geq\; \mathrm{Tr}(H_i X) + g_i^T x + s_i \\ 0 \;\preceq\; X - x \, x^T \end{array} \right. \tag{2.2.6}$$

As we assume Slater's constraint qualification to be satisfied, the associated necessary and sufficient optimality conditions are that there exists a $Z \in \mathbb{S}^{n_x}$ and a $\lambda \in \mathbb{R}^m$ such that

$$\begin{array}{rclcrcl} 0 & = & H_0 - \sum_{i=1}^m \lambda_i H_i + Z & \qquad & 0 & = & g_0 - \sum_{i=1}^m \lambda_i g_i - 2Zx \\ 0 & \geq & \mathrm{Tr}(H_i X) + g_i^T x + s_i & & 0 & \preceq & X - x\,x^T \\ 0 & \leq & \lambda_i & & 0 & \preceq & Z \\ 0 & = & \mathrm{Tr}(ZX) - x^T Z x & & 0 & = & \sum_{i=1}^m \lambda_i \left(\mathrm{Tr}(H_i X) + g_i^T x + s_i\right) . \end{array}$$

It is readily checked that the point $(x^*, X^*, \lambda^*, Z^*)$ satisfies the above conditions by construction. Thus, we have found a global solution to problem (2.2.5) whose objective value $\hat{V}$ is coinciding with the objective value V of the original possibly non-convex QCQP. The statement of the Theorem is a direct consequence. □

Unfortunately, the above criterion is not necessary, i.e., there exist cases in which we can find a global optimizer of a non-convex QCQP at which the above sufficient condition is not satisfied. Only for convex QCQPs, i.e., if the matrix H_0 is negative semi-definite while the matrices H_i are for $i \in \{1, \ldots, m\}$ positive semi-definite the sufficient conditions in the above Lemma 2.3 are always satisfied. In general, we only know that the Hessian matrix $H(\lambda^*)$ must be negative semi-definite on the tangential sub-space spanned by the active constraints (c.f. e.g. [182] for a discussion of the details of second order necessary condition for local minima).

Example 2.8: Consider the simple scalar optimization problem

$$\max_x \; x^2 + x \quad \text{s.t.} \quad x^2 \leq 1 \, .$$

The problem has obviously a local maximum at $x = -1$ (with dual solution $\lambda = \frac{1}{2}$) while the global solution is at $x = 1$ (with dual solution $\lambda = \frac{3}{2}$). In the local solution at $x = -1$, the associated symmetric matrix turns out to be $H(1/2) = \frac{1}{2}$ while the global solution satisfies $H(3/2) = -\frac{1}{2}$, i.e., we are in the lucky case that we can verify the sufficient optimality conditions from Lemma 2.3. Indeed, the corresponding dual can be written as

$$\begin{array}{ll} \max\limits_{\gamma, \lambda \geq 0} & \gamma \\ \text{s.t.} & 0 \succeq \begin{pmatrix} \lambda - \gamma & \frac{1}{2} \\ \frac{1}{2} & 1 - \lambda \end{pmatrix} . \end{array}$$

This convex SDP has a unique solution at $(\gamma^*, \lambda^*) = (2, \frac{3}{2})$.

The above example illustrates that Lemma 2.3 can be useful in some special cases. In fact, if we have $m = 1$, i.e., if there is only one possibly non-convex quadratic constraint, the condition in Lemma 2.3 is also necessary. We summarize this result in form of the following Theorem:

Theorem 2.1 (Tight Version of the S-Procedure): *Let us consider a QCQP of the general form* (2.2.1) *for the special case that we have only one quadratic constraint, i.e., $m = 1$, and let H_1 be positive definite. In this case, the objective value V of the QCQP* (2.2.1) *is coinciding with the objective value $\hat{V}$ of the convex SDP* (2.2.2)*, i.e., we have strong duality.*

Proof (Via the "Mirror Trick"): As H_1 is assumed to be positive definite, we can also assume without loss of generality that the QCQP can equivalently be written as

$$\max_x \; x^T D x + d^T x \quad \text{s.t.} \quad \sum_{i=1}^{n} x_i^2 \; \leq \; 1 \tag{2.2.7}$$

for some diagonal matrix $D \in \mathbb{S}^n$ and $d \in \mathbb{R}^n$. If the QCQP is not in this form, we can always reformulate the problem by simple linear algebra transformations (rescale,shift,and diagonalize). Our aim is to show that every primal-dual global maximizer (x^*, λ^*) of the problem (2.2.7) satisfies $D_{i,i} - \lambda_i \; \leq \; 0$ for all $i \in \{1, \ldots, n\}$ such that we can apply Lemma 2.3. Thus, we assume by contradiction that we have a global maximizer (x^*, λ^*) and a component $i \in \{1, \ldots, n\}$ for which $D_{i,i} - \lambda_i^* \; > \; 0$. Multiplying the stationarity condition of the form $2(D_{i,i} - \lambda_i^*)x_i^* + d_i \; = \; 0$ with x_i^* we find

$$\frac{d_i x_i^*}{2} \;\; = \;\; -(D_{i,i} - \lambda_i^*)(x_i^*)^2 \; < \; 0\,,$$

as we may assume $x_i^* \neq 0$, as we only need to consider the non-trivial case $d_i \neq 0$. Now, we construct a mirror point $y^* \in \mathbb{R}^n$ of x^* by choosing $y_j^* \; := \; x_j^*$ for all $j \neq i$ as well as $y_i^* \; := \; -x_i^*$. Note that the mirror point y^* is feasible as we have

$$\sum_{i=1}^{n} (y_i^*)^2 \; = \; \sum_{i=1}^{n} (x_i^*)^2 \; \leq \; 1\,.$$

In addition, we have $(y^*)^T D y^* \;\; = \;\; (x^*)^T D x^*$ as the matrix D is diagonal as well as $d^T y^* \; > \; d^T x^*$. Note that this contradicts our assumption that x^* is a global maximizer, as y^* is a feasible point which yields a larger objective value than x^*. Consequently, we may conclude the statement of the Theorem by employing Lemma 2.3. □

Remark 2.3: *In order to understand the idea behind the proof, it might help to verify in Example 2.8 that the global solution at* $x = 1$ *is actually a mirror point of the local solution at* $x = -1$ *.*

Remark 2.4: *The above Theorem has applications in the field of trust-region methods [60], where a locally quadratic and possibly indefinite model must be minimized subject to a quadratic trust region constraint.*

Remark 2.5: *Note that there exist many interesting variants of the above Theorem. For example in [180] it is remarked that an unconstrained and possibly non-convex cubic problem of the form*

$$\min_x \frac{1}{2} x^T H x + g^T x + \mu \|x\|_2^3$$

with $\mu > 0$ *being sufficiently large can be solved globally by introducing a scalar slack variable* a *and reformulating the problem into an equivalent problem of the form*

$$\min_{x,a} \frac{1}{2} x^T H x + g^T x + \mu\, a^{\frac{3}{2}} \quad \textit{s.t.} \quad x^T x \;\leq\; a \;,$$

such that the tight version of the S*-procedure can be applied in the variable* x*. This leads to a convex problem as the term* $a^{\frac{3}{2}}$ *is convex in* $a \geq 0$*.*

The main reason why we are in this thesis interested in the above tight version of the S-procedure can be motivated by looking once more at Example 2.7: if the uncertainty set is not a general intersection of ellipsoids but just one ellipsoid, the conservative robust counterpart reformulation for the case of quadratic uncertainty becomes exact.

Example 2.9: Let us once more regard the unconstrained least squares optimization problem from Example 2.2 which has the form

$$\min_{x \in X} \;\; \| (A + \Delta) x \|_2^2 \quad \text{with} \quad W \;:=\; \{\; \Delta \mid \|\Delta\|_{\mathrm{F}} \;\leq\; 1 \,\}\,,$$

for which the data matrix $A \in \mathbb{R}^{m \times n}$ is given but the matrix $\Delta \in W$ unknown. As an alternative to an application of the triangle inequality, we can use that the least squares term is a quadratic form in Δ while we have only one convex quadratic constraint. Applying

the tight version of the S-procedure (Theorem 2.1), we find

$$\begin{aligned} V(x) &:= \max_{\Delta \in W} \| (A+\Delta)x \|_2^2 \\ &= \inf_{\lambda > \| x \|_2^2} \max_{\Delta} \operatorname{Tr}\Big((A+\Delta)^T (A+\Delta) x x^T \Big) - \lambda \operatorname{Tr}(\Delta^T \Delta) + \lambda \\ &= \inf_{\lambda > \| x \|_2^2} \| Ax \|_2^2 + \| Ax \|_2^2 \frac{\| x \|_2^2}{\lambda - \| x \|_2^2} + \lambda \\ &= (\| Ax \|_2 + \| x \|_2)^2 \ . \end{aligned}$$

The result is of course the same as with the strategy from Example 2.2.

2.3 Inner and Outer Ellipsoidal Approximation Methods

Even if a given set $\mathcal{F} \subseteq \mathbb{R}^n$ is convex, it is not always clear how we can represent it on a computer. For example, if $\mathcal{F}$ is a given polytope with a large number of facets, we might run out of memory or at least operations which involve the set $\mathcal{F}$ can become very expensive. In such cases, it is often more reasonable to study suitable set approximation techniques. In this section, we put a special emphasis on inner and outer set approximation methods which exploit the fact that ellipsoids are for many situations suitable geometric objects for a representation or approximation of sets in higher dimensions.

Let us start our review of set-theoretic methods by the introduction of the support function $V : \mathbb{R}^n \to \mathbb{R}$ of a compact and convex set $\mathcal{F}$, which is defined as

$$V(c) := \max_x \ c^T x \quad \text{s.t.} \quad x \in \mathcal{F} \ . \tag{2.3.1}$$

In the analysis of convex sets support functions can be considered as one of the most basic but also very useful tools. Here, $V(c)$ can be interpreted as the maximum extension of the set $\mathcal{F}$ in the direction which is defined by the vector c.

Example 2.10 (The Support of an Ellipsoid): With $Q \in \mathbb{S}_+^n$ and $q \in \mathbb{R}^n$ we consider the special case that the set $\mathcal{F} = \mathcal{E}(Q,q)$ is an ellipsoid. In this case, the support or maximum extension $V(c)$ is for all directions c given by

$$V(c) = \max_{x \in \mathcal{E}(Q,q)} c^T x = \sqrt{c^T Q c} + c^T q \ , \tag{2.3.2}$$

i.e., for this special case we have found an explicit expression for the support function.

The main reason why support functions are useful is that they can be employed to uniquely characterize a convex set $\mathcal{F}$. While the above definition constructs the support function V from a convex and compact set $\mathcal{F}$, we can also consider an inverse argumentation, i.e., we can re-construct the set $\mathcal{F}$ from its support function. For this aim, we define the supporting halfspaces $\mathcal{H}(c) \subseteq \mathbb{R}^n$ of a compact and convex set $\mathcal{F}$ for all directions $c \in \mathbb{R}^n$ as

$$\mathcal{H}(c) \;:=\; \left\{ x \in \mathbb{R}^n \mid c^T x \;\leq\; V(c) \right\} . \tag{2.3.3}$$

Now, we can regard the following Lemma, whose proof can be found in [46]:

Lemma 2.4: *If $\mathcal{F}$ is a compact and convex set, then it is uniquely characterized by its support function. More precisely, a convex and compact set $\mathcal{F}$ can be represented as the intersection of its supporting halfspaces:*

$$\mathcal{F} \;=\; \bigcap_{c \in \mathbb{R}^n \setminus \{0\}} \mathcal{H}(c) . \tag{2.3.4}$$

In the following discussion it will be important to recognize ellipsoids when dealing with support functions. In order to illustrate why Lemma 2.4 is helpful for that purpose, we consider the following example:

Example 2.11 (Identification of Ellipsoids): If $\mathcal{F} \subseteq \mathbb{R}^n$ is a compact and convex set whose support function satisfies

$$\forall c \in \mathbb{R}^n : \qquad V(c) \;=\; \left\| Q^{\frac{1}{2}} c \right\|_2 + c^T q$$

for some $Q \in \mathbb{S}^n_+$ and some $q \in \mathbb{R}^n$, then $\mathcal{F}$ is an ellipsoid of the form $\mathcal{E}(Q, q)$.

In the following, we are interested in constructing parameterized inner and outer approximations of convex and compact sets. Here, we have to specify first what we understand when referring to parameterized (or lifted) set approximations. For this aim, we assume that we have a suitable set $\mathbb{D}^+ \subseteq \mathbb{R}^{n_p}$ of parameters while $\Pi(\mathbb{R}^n)$ denotes the set of all subsets (including the empty set) of $\mathbb{R}^n$. Now, a function $\mathcal{F}^+ : \mathbb{D}^+ \to \Pi(\mathbb{R}^n)$ is called a parameterized outer approximation of the set $\mathcal{F}$, if we have

$$\forall \lambda \in \mathbb{D}^+ : \;\; \mathcal{F} \;\subseteq\; \mathcal{F}^+(\lambda) .$$

Moreover, we say that the parameterized outer approximation is tight if the intersection of the sets $\mathcal{F}^+(\lambda)$ coincides with the set $\mathcal{F}$, i.e., if we have

$$\mathcal{F} \;=\; \bigcap_{\lambda \in \mathbb{D}^+} \mathcal{F}^+(\lambda)\,.$$

It should be clear that parameterized inner approximations of a set $\mathcal{F}$ can be defined in an analogous manner.

Example 2.12: Let us consider the case that $\mathcal{F} := \{\, x \in \mathbb{R}^n \mid Ax \leq b \,\}$ is a given polytope with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Note that if the integer m is very large, this polytope is expensive to store. Thus, we might be interested in constructing a polytope with at most $l \ll m$ facets which approximates the polytope $\mathcal{F}$ from outside. In order to provide one possible approach to deal with this problem recall the techniques from Example 2.4, where we have discussed robust linear programming. Transferring these techniques to our current situation, we observe that the parameterized polyhedra of the form $\mathcal{F}^+(\Lambda) := \{\, x \mid C(\Lambda)x \leq d(\Lambda) \,\}$ with

$$C(\Lambda) \;:=\; \Lambda^T A \quad \text{and} \quad d(\Lambda) \;:=\; \Lambda^T b$$

are for all $\Lambda \in \mathbb{D}^+$ outer approximations of the set $\mathcal{F}$. Here, $\mathbb{D}^+$ denotes the set of all $(m \times l)$-matrices with positive components, i.e., we have used the following definition of the parameter set:

$$\mathbb{D}^+ \;:=\; \Big\{\, \Lambda \in \mathbb{R}^{m \times l} \;\mid\; \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,l\} : \quad \Lambda_{i,j} \;\geq\; 0 \,\Big\}\,.$$

Taking the intersection of the sets $\mathcal{F}^+(\Lambda)$ for all $\Lambda \in \mathbb{D}^+$ yields again the original polytope $\mathcal{F}$, as there are no duality gaps, i.e., the parameterized outer approximation is tight.

Regarding the argumentation in the above example, we can already recognize that there are connections between robust counterpart problems and set approximation techniques. The question which we are asking here is how we can systematically find or construct a suitable set $\mathbb{D}^+$ and an associated set valued function $\mathcal{F}^+$ with the above tightness properties, such that the sets of the form $\mathcal{F}^+(\lambda)$ have a suitable geometry aiming at efficient representations of these sets. For convex sets, one possible strategy is to exploit the concept of duality. In order to understand this, note that the support function V of a convex and compact set $\mathcal{F}$ is by definition the optimal value of a convex optimization problem. Thus, if $\mathcal{F}$ has a non-empty interior, we may assume that we can construct an

associated dual function $D : \mathbb{R}^n \times \mathbb{D}^+ \to \mathbb{R}$ such that V can equivalently be expressed as the optimal value of a minimization problem of the form

$$V(c) \;=\; \inf_{\lambda \in \mathbb{D}^+} D(c, \lambda) \; .$$

Using this notation, we can define a set valued function $\mathcal{F}^+$ as

$$\forall \lambda \in \mathbb{D}^+ : \qquad \mathcal{F}^+(\lambda) \;:=\; \bigcap_{c \in \mathbb{R}^n} \Big\{ x \in \mathbb{R}^n \;\mid\; c^T x \;\leq\; D(c, \lambda) \Big\} \; . \tag{2.3.5}$$

Note that the sets $\mathcal{F}^+(\lambda)$ are by construction convex as intersections of convex sets are convex. Additionally, the function $\mathcal{F}^+$ turns out to be a tight parameterized outer approximation:

Theorem 2.2 (Lifted Outer Approximations): *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a given compact set and let the function $\mathcal{F}^+$ be constructed as in equation* (2.3.5)*, then $\mathcal{F}^+$ is a tight parameterized outer approximation, i.e., we have*

$$\mathcal{F} \;=\; \bigcap_{\lambda \in \mathbb{D}^+} \mathcal{F}^+(\lambda) \; .$$

The same construction is also possible for non-convex sets $\mathcal{F}$, but in this case the parameterized outer approximation $\mathcal{F}^+$ is not tight anymore.

Proof: This Theorem is a consequence of Lemma 2.4. In order to show this, we regard two compact and convex sets $\mathcal{F}_1, \mathcal{F}_2 \subseteq \mathbb{R}^n$ together with their associated support functions $V_1, V_2 : \mathbb{R}^n \to \mathbb{R}$. Due to Lemma 2.4 we know that we have an inclusion of the form $\mathcal{F}_1 \subseteq \mathcal{F}_2$ if and only if the inequality $V_1(c) \leq V_2(c)$ is satisfied for all $c \in \mathbb{R}^n$. If we apply this observation with $\mathcal{F}_1 := \mathcal{F}$ and $\mathcal{F}_2 := \mathcal{F}^+(\lambda)$, we find that we must have

$$\mathcal{F} \;\subseteq\; \mathcal{F}^+(\lambda)$$

for all $\lambda \in \mathbb{D}^+$, as the associated support functions satisfy by construction an inequality of the form $V_1(c) = V(c) \leq D(c, \lambda) = V_2(c)$. If there is no duality gap, we find that the parameterized outer approximation is tight. □

In the following, we shall see that Theorem 2.2 is a fruitful tool for the development of set approximation methods. However, before we discuss such applications, we first extend the above general framework to inner approximations, too. Let us introduce a notation for polar sets:

Definition 2.4 (Polar Set): *Let $\mathcal{F} \in \mathbb{R}^n$ be a given set. The polar set $\mathcal{F}^* \in \mathbb{R}^n$ associated with the set $\mathcal{F}$ is defined as*

$$\mathcal{F}^* \;:=\; \left\{ y \in \mathbb{R}^n \;\middle|\; \sup_{x \in \mathcal{F}} \; x^T y \;\leq\; 1 \right\} .$$

Note that the polar set is by definition a convex set, as the supremum over linear functions is a convex function. In our context, polar sets are useful as they can be employed to swap between inner and outer approximations of absolutely convex sets. More precisely, our plan is to employ the following two propositions which can for example be found in [84]:

Proposition 2.1: *Let $\mathcal{F}_1, \mathcal{F}_2 \in \mathbb{R}^n$ be two given sets with $\mathcal{F}_1 \subseteq \mathcal{F}_2$. Then the polar set of $\mathcal{F}_2$ is an inner approximation of the polar set of $\mathcal{F}_1$, i.e., we have $\mathcal{F}_2^* \subseteq \mathcal{F}_1^*$.*

Proposition 2.2: *If $\mathcal{F}$ is compact and absolutely convex then the polar of the polar set is coinciding with the original set, i.e., $(\mathcal{F}^*)^* = \mathcal{F}$. Here, a set $\mathcal{F}$ is absolutely convex, if we have for all $x, y \in \mathcal{F}$ also*

$$\lambda_1 x + \lambda_2 y \in \mathcal{F} \quad \text{for all} \quad \lambda_1, \lambda_2 \in \mathbb{R} \quad \text{with} \quad |\lambda_1| + |\lambda_2| \;\leq\; 1 \,.$$

In order to understand how we can work with polar sets, we start with the following example:

Example 2.13 (The Polar Set of an Ellipsoid): Let $Q \in \mathbb{S}^n_{++}$ be a positive definite matrix and $\mathcal{E}(Q)$ its associated ellipsoid using the notation from Definition (2.1). In order to compute the polar of $\mathcal{E}(Q)$ we note that

$$\begin{aligned} \mathcal{E}(Q)^* \;&=\; \left\{ y \in \mathbb{R}^n \;\middle|\; \max_{x^T Q^{-1} x \,\leq\, 1} \; x^T y \;\leq\; 1 \right\} \\ &=\; \left\{ y \in \mathbb{R}^n \;\middle|\; y^T Q y \;\leq\; 1 \right\} \;=\; \mathcal{E}(Q^{-1}) \,. \end{aligned}$$

Thus, the polar of an ellipsoid is again an ellipsoid. In particular, the polar set of the unit ball $E(I)$ remains the unit ball.

One possible strategy to construct a parameterized (or lifted) inner approximation of an absolutely convex and compact set $\mathcal{F}$ is to first compute the support function $V^* : \mathbb{R}^n \to \mathbb{R}$ of the polar set of $\mathcal{F}$, i.e., we define for all $c \in \mathbb{R}^n$:

$$V^*(c) \;:=\; \sup_x \; c^T x \quad \text{s.t.} \quad x \in \mathcal{F}^* \,.$$

Now, we proceed similarly to the construction of outer approximations, i.e., we assume that we manage to find a convex set $\mathbb{D}^-$ and an associated dual function $D^* : \mathbb{R}^n \times \mathbb{D}^- \to \mathbb{R}$ such that the objective value of the above convex maximization problem can equivalently be written as

$$V^*(c) \;=\; \inf_{\lambda \in \mathbb{D}^-} D^*(c,\lambda)\;.$$

Using this notation, we can define a set valued function $\mathcal{F}_- : \mathbb{D}^- \to \Pi(\mathbb{R}^n)$ for all $\lambda \in \mathbb{D}^-$ as

$$\mathcal{F}_-(\lambda) \;:=\; \left[\; \mathcal{F}_-^*(\lambda) \;\right]^* \quad \text{with} \quad \mathcal{F}_-^*(\lambda) \;:=\; \bigcap_{c \in \mathbb{R}^n} \Big\{ x \in \mathbb{R}^n \;\mid\; c^T x \;\leq\; D^*(c,\lambda) \Big\}\;. \tag{2.3.6}$$

Here, it should be mentioned that the function D^* takes only positive values which implies that the convex sets $\mathcal{F}_-^*(\lambda)$ contain for all $\lambda \in \mathbb{D}^-$ the origin such that the corresponding polar sets $\mathcal{F}_-(\lambda)$ are well-defined. Using this definition, the function $\mathcal{F}_-$ turns out to be a tight parameterized inner approximation of the original set $\mathcal{F}$ as summarized within the following Theorem:

Theorem 2.3 (Lifted Inner Approximations): *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a given absolutely convex and compact set and let the function $\mathcal{F}_-$ be constructed as in equation* (2.3.6)*, then $\mathcal{F}_-$ is a tight parameterized inner approximation, i.e., we have*

$$\mathcal{F} \;=\; \bigcup_{\lambda \in \mathbb{D}^-} \mathcal{F}_-(\lambda)\;.$$

The same construction is also possible for non-convex sets $\mathcal{F}$, but in this case the parameterized inner approximation $\mathcal{F}_-$ is not tight anymore.

Proof: The proof of this theorem can be obtained in two steps: in the first step, we apply Theorem 2.2, which guarantees that the sets $\mathcal{F}_-^*(\lambda) \supseteq \mathcal{F}^*$ are by construction tight parameterized outer approximations of the polar set $\mathcal{F}^*$. And in the second step, we apply Proposition 2.1 to show that the polar sets $\mathcal{F}_-(\lambda)$ of the outer approximations of $\mathcal{F}^*$ must be inner approximation of the original set $\mathcal{F}$. □

So far, the above construction methods for parameterized inner and outer approximations of convex sets are discussed on a quite abstract level. However, the applicability of the framework becomes clear by studying some more concrete cases. The aim of the following sections is to work out parameterized inner and outer approximations of convex sets by studying particular constructions which are based on ellipsoids and polytopes. This analysis is driven by the observation that ellipsoids are suitable candidates for the approximation of more general sets.

Outer Approximations of Sums of Ellipsoids

In this section, we concentrate on outer approximations of the geometric sum of N ellipsoids. Let us define such a set sum as follows:

Definition 2.5 (Sum of Ellipsoids): *Let $Q_i \in \mathbb{S}_+^n$ with $i \in \{1, \ldots, N\}$ be given positive semi-definite matrices, $q_i \in \mathbb{R}^n$, and $\mathcal{E}(Q_i, q_i)$ the associated ellipsoids. The sum of these ellipsoids is defined as the standard Minkowski sum, i.e., we write*

$$\sum_{i=1}^{N} \mathcal{E}(Q_i, q_i) \;:=\; \left\{ \sum_{i=0}^{N} x_i \;\in \mathbb{R}^n \;\middle|\; x_i \in \mathcal{E}(Q_i, q_i) \quad \textit{for all } \; i \in \{1, \ldots, N\} \right\} .$$

It can easily be checked that a sum of ellipsoids is a convex set. Moreover, a finite sum of ellipsoids is compact. However, a sum of ellipsoids is in general not again an ellipsoid. For illustration, we regard the following example:

Example 2.14: Let $e_1, e_2 \in \mathbb{R}^2$ with $e_1 := (1,0)^T$ and $e_2 := (0,1)^T$ be the unit vectors in $\mathbb{R}^2$ as well as $Q_1 := e_1 e_1^T$ and $Q_2 := e_2 e_2^T$. The sum of the ellipsoids

$$\mathcal{E}(Q_1) + \mathcal{E}(Q_2) \;=\; \left\{ (x,y) \in \mathbb{R}^2 \;\middle|\; x^2 \;\leq\; 1 \text{ and } \; y^2 \;\leq\; 1 \right\}$$

is the unit square. More generally, for m generating vectors $a_1, \ldots, a_m \in \mathbb{R}^n$ the associated zonotope can be written as a sum of m ellipsoids:

$$\sum_{i=1}^{m} \mathcal{E}\left(a_i a_i^T \right) \;=\; \left\{ \sum_{i=1}^{m} \lambda_i a_i \;\in \mathbb{R}^n \;\middle|\; -1 \;\leq\; \lambda_i \;\leq\; 1 \text{ for all } \; i \in \{1, \ldots, m\} \right\} .$$

Example 2.15: Sums of ellipsoids have been analyzed for a long time - especially in the context of dynamic systems. For example Schweppe and Glover [102, 209] have used ellipsoids to approximate reachable sets of uncertain dynamic systems. These techniques have later extensively been worked out by Kurzhanski and Varaiya [146, 144] and also by Brockman and Corless in [47]. Later, in Chapter 5, we will review these techniques in all details, but in order to outline already at this point why sums of ellipsoids are important in the context of dynamic systems, we regard a linear discrete time system of the form

$$x^+ \;=\; Ax \;+\; Bw \;, \tag{2.3.7}$$

where $x \in \mathcal{E}(Q_x)$ is the current state which is known to be in the ellipsoid $\mathcal{E}(Q_x) \subseteq \mathbb{R}^{n_x}$, while the input $w \in \mathcal{E}(Q_w) \subseteq \mathbb{R}^{n_w}$. The "next" state, which is denoted by $x^+ \in \mathbb{R}^{n_x}$, is then known to be in the set

$$\mathcal{E}\left(A Q_x A^T \right) \;+\; \mathcal{E}\left(B Q_w B^T \right) ,$$

which is a sum of two ellipsoids. In the following consideration, we discuss how to approximate this set sum again with one single but parameterized ellipsoid from outside.

In the following, we assume without loss of generality that the ellipsoids are centered at 0, i.e., it is enough to study sums of the form $\mathcal{F} = \sum_{i=1}^{N} \mathcal{E}(Q_i) \subseteq \mathbb{R}^n$. Here, we use the fact that a sum of general ellipsoids may always be written as

$$\sum_{i=1}^{N} \mathcal{E}(Q_i, q_i) = \left\{ \sum_{i=1}^{N} q_i \right\} + \sum_{i=1}^{N} \mathcal{E}(Q_i) \,,$$

i.e., we can always decompose the sum into an additive offset and a sum of centered ellipsoids. Following the construction principle which has been outlined above, we first compute the support function of a sum of ellipsoids. For this aim, we assume for a moment that the matrices $Q_1, \ldots, Q_N$ are invertible such that we can write the corresponding maximization problem in the form

$$V(c) = \max_{x_1,\ldots,x_N} c^T \left(\sum_{i=1}^{N} x_i \right) \quad \text{s.t.} \quad x_i^T Q_i^{-1} x_i \leq 1 \quad \text{for all } i \in \{1, \ldots, N\} \,.$$

As this is a convex maximization problem and $x_1 = \ldots = x_N = 0$ is a feasible point, i.e., Slater's condition is satisfied, we continue by computing $V(c)$ via the associated dual problem

$$\begin{aligned} V(c) &= \inf_{\lambda > 0} \max_{x_1,\ldots,x_N} \sum_{i=1}^{N} \Big(c^T x_i - \lambda_i x_i^T Q_i^{-1} x_i + \lambda_i \Big) \\ &= \inf_{\lambda > 0} \sum_{i=1}^{N} \frac{c^T Q_i c}{4\lambda_i} + \sum_{i=1}^{N} \lambda_i \,. \end{aligned}$$

In order to understand the next reformulation step, it is helpful to first verify the general relation, which is in a similar version also known under the name "(tight) arithmetic-geometric inequality" or shorter "(tight) AM-GM inequality":

$$\inf_{\kappa > 0} \frac{a}{4\kappa} + \kappa\, b = \sqrt{ab} \,, \tag{2.3.8}$$

which holds for all $a, b \in \mathbb{R}_+$. In order to make use of this relation, we rescale the dual variables λ_i by introducing a redundant scaling factor κ finding

$$V(c) = \inf_{\lambda > 0} \inf_{\kappa > 0} \sum_{i=1}^{N} \frac{c^T Q_i c}{4\kappa\lambda_i} + \sum_{i=1}^{N} \kappa\lambda_i = \inf_{\lambda > 0} \sqrt{c^T Q(\lambda)\, c} \,. \tag{2.3.9}$$

Here, we have introduced a matrix valued function $Q : \mathbb{R}^N_{++} \to \mathbb{S}^N_+$ which is defined by

$$\forall \lambda \in \mathbb{R}^N_{++} : \qquad Q(\lambda) \; := \; \left(\sum_{i=1}^N \frac{1}{\lambda_i} Q_i \right) \left(\sum_{i=1}^N \lambda_i \right) \; .$$

The main aspect of this transformation is that equation (2.3.9) holds for all directions $c \in \mathbb{R}^n \setminus \{0\}$. In other words, we can for any $\lambda \in \mathbb{R}^N_{++}$ interpret the set $\mathcal{F}^+(\lambda) \; := \; \mathcal{E}(\, Q(\lambda)\,)$ as an ellipsoidal outer approximation. As we can rescale the variables λ_i once more such that $\sum_{i=1}^N \lambda_i \; = \; 1$, we arrive at the following Theorem:

Theorem 2.4: *Let us define the set $\mathbb{D}^+ \subseteq \mathbb{R}^N_{++}$ of feasible parameters to be a half-open unit simplex by*

$$\mathbb{D}^+ \; := \; \left\{ \; \lambda \in \mathbb{R}^N_{++} \; \middle| \; \sum_{i=1}^N \lambda_i \; \leq \; 1 \; \right\} \; .$$

The ellipsoid $\mathcal{F}^+(\lambda) \; := \; \mathcal{E}(\, Q(\lambda)\,)$ is for every $\lambda \in \mathbb{D}^+$ an outer approximation of the set $\mathcal{F} \; = \; \sum_{i=1}^N \mathcal{E}(Q_i)$. In other words, we have

$$\forall \lambda \in \mathbb{D}^+ : \qquad \mathcal{F} \; = \; \sum_{i=1}^N \mathcal{E}(Q_i) \quad \subseteq \quad \mathcal{E}\left(\sum_{i=1}^N \frac{1}{\lambda_i} Q_i \right) \; = \; \mathcal{F}^+(\lambda) \; . \qquad (2.3.10)$$

Here, the matrices $Q_1, \ldots, Q_n \in \mathbb{S}^n_+$ are not necessarily invertible. Moreover, the parameterized outer approximation is tight, i.e., we have $\mathcal{F} \; = \; \bigcap\limits_{\lambda \in \mathbb{D}^+} \mathcal{F}^+(\lambda)$.

Proof: If the matrices $Q_1, \ldots, Q_N$ are invertible, the above statement follows from equation (2.3.9) as this equation holds for all directions c while an ellipsoid can uniquely be characterized by the intersection of halfspaces as discussed in Lemma 2.4. Thus, the above statement is coinciding with the statement of Theorem 2.2 with the only difference that the statement is now specialized to the case that $\mathcal{F}$ is a sum of ellipsoids.

In order to complete the above argumentation, we still have to show that the inclusion (2.3.10) holds also for general positive semi-definite matrices $Q_1, \ldots, Q_N$. Here, the main idea is to fix some $\lambda \in \mathbb{R}^N_{++}$ add a small regularization term εI for some $\varepsilon > 0$ to the matrices $Q_1, \ldots, Q_N$, such that the inclusion (2.3.10) still holds. As the set $\mathcal{F}$ is compact, the limit for $\varepsilon \to 0$ exists on both sides of the inclusion (2.3.10). Consequently, we can conclude that this inclusion also holds for the case that $Q_1, \ldots, Q_N$ are general positive semi-definite matrices. Finally, we have to show that the relation $\mathcal{F} \; = \; \bigcap\limits_{\lambda \in \mathbb{D}^+} \mathcal{F}^+(\lambda)$ holds. For this aim, we apply the same trick, i.e., we first construct

the regularized support function

$$V_\epsilon(c) \;:=\; \inf_{\lambda>0} \; \sum_{i=1}^{N} \frac{c^T\,[\,Q_i + \epsilon I\,]\,c}{4\lambda_i} + \sum_{i=1}^{N} \lambda_i \;.$$

The main point is now that we may exchange the limits in the transformation

$$\begin{aligned} \lim_{\epsilon\to 0} V_\epsilon(c) \;&:=\; \lim_{\epsilon\to 0} \inf_{\lambda>0} \sum_{i=1}^{N} \frac{c^T\,[\,Q_i + \epsilon I\,]\,c}{4\lambda_i} + \sum_{i=1}^{N} \lambda_i \\ &=\; \inf_{\lambda>0} \lim_{\epsilon\to 0} \sum_{i=1}^{N} \frac{c^T\,[\,Q_i + \epsilon I\,]\,c}{4\lambda_i} + \sum_{i=1}^{N} \lambda_i \;=\; \inf_{\lambda>0} \sum_{i=1}^{N} \frac{c^T Q_i c}{4\lambda_i} + \sum_{i=1}^{N} \lambda_i \;, \end{aligned}$$

since the limits uniformly exist. Thus, the argumentation can be rescued for the case that the matrices $Q_1, \ldots, Q_N$ are general positive semi-definite matrices. □

Remark 2.6: *As we have already mentioned within Example 2.15, sums of ellipsoids have been analyzed by many authors in different contexts. The above Theorem has for example in a very similar version been proven by Kurzhanski and Varaiya [144, 146] in the context of computing reachable sets for dynamic systems. In the literature of robust convex optimization, for example in the work by Ben-Tal and Nemirovski [21], similar results can be found.*

Remark 2.7 (Relation to the S-procedure): *Theorem 2.4 can be derived with different techniques. A possible alternative derivation is based on the observation that we have an inclusion of the form $\sum_{i=0}^{N} \mathcal{E}(Q_i) \subseteq \mathcal{E}(Q)$ for some positive definite matrices $Q, Q_i \in \mathbb{S}^n_{++}$ if and only if the inequality*

$$1 \;\geq\; \max_x \left(\sum_{i=1}^{N} x_i\right)^T Q^{-1} \left(\sum_{i=1}^{N} x_i\right) \quad \text{s.t.} \quad x_i^T Q_i^{-1} x_i \;\leq\; 1 \quad i \in \{1, \ldots, N\} \tag{2.3.11}$$

is satisfied. On the right-hand side of inequality (2.3.11) *we find a QCQP problem for which we can apply the S-procedure in order to obtain a sufficient condition under which we can guarantee the desired inclusion. By working this out we find the following condition: if there exist multipliers $\lambda_1, \ldots, \lambda_N > 0$ which satisfy the inequality $\sum_{i=1}^{N} \lambda_i \leq 1$ as well as an LMI of the form*

$$\begin{pmatrix} Q^{-1} - \lambda_1 Q_1^{-1} & Q^{-1} & \ldots & Q^{-1} \\ Q^{-1} & Q^{-1} - \lambda_2 Q_2^{-1} & \ldots & Q^{-1} \\ \vdots & & \ddots & \vdots \\ Q^{-1} & \ldots & Q^{-1} & Q^{-1} - \lambda_N Q_N^{-1} \end{pmatrix} \;\preceq\; 0 \;, \tag{2.3.12}$$

then inequality (2.3.11) *is satisfied*[1]*. This LMI can be solved recursively by taking Schur complements and simplifying the recursion with the Sherman-Morrison-Woodbury formula. This shows that the LMI* (2.3.12) *is satisfied if and only if the condensed semi-definite inequality* $Q \succeq \sum_{i=1}^{N} \frac{1}{\lambda_i} Q_i$ *holds. Now, we can continue as in the first proof of Theorem 2.4 to extend this result for positive semi-definite matrices* $Q_1, \ldots, Q_N$*, too.*

We should be clear about the fact that the above Theorem 2.4 does not make any statement about how we can find a $\lambda \in \mathbb{D}^+$ which yields an "optimal" ellipsoidal outer approximation. Rather, Theorem 2.4 should be interpreted as a tool which yields a whole family of outer approximations. Which of these approximations can be regarded optimal depends on the context and our objective. For example, if we optimize the ellipsoidal outer approximation for a specified direction, we obtain the following result:

Corollary 2.1 (Tight Ellipsoidal Outer Approximations): *If the optimization problem*

$$\inf_{\lambda \in \mathbb{D}^+} \max_{x \in \mathcal{F}^+(\lambda)} c^T x \;=\; \inf_{\lambda \in \mathbb{D}^+} \; c^T \left(\sum_{i=1}^{N} \frac{1}{\lambda_i} Q_i \right) c \tag{2.3.13}$$

has a minimum at λ^* *the associated ellipsoidal outer approximation* $\mathcal{F}^+(\lambda^*) = \mathcal{E}(Q(\lambda^*))$ *is tight in the sense that the ellipsoid* $\mathcal{E}(Q(\lambda^*))$ *touches the set* $\mathcal{F} = \sum_{i=1}^{N} \mathcal{E}(Q_i)$ *once in the direction* c *and - due to symmetry - once more in the direction* $-c$*. Moreover, the above optimization problem is convex.*

Proof: This corollary follows directly from equation (2.3.9). Here, the convexity follows from the fact that the sum over the convex functions of the form $\frac{c^T Q_i c}{\lambda_i}$ is convex while the set $\mathbb{D}^+$ is convex, too. □

Remark 2.8: *The convex optimization problem from Corollary 2.1 can also be written as*

$$\inf_{\lambda \in \mathbb{R}^N_{++}} \; c^T Q(\lambda)\, c \quad \textit{with} \quad Q(\lambda) \;=\; \left(\sum_{i=1}^{N} \frac{1}{\lambda_i} Q_i \right) \left(\sum_{i=1}^{N} \lambda_i \right) . \tag{2.3.14}$$

The only difference is that we have dropped the scaling constraint. In this form, problem (2.3.14) *is non-convex and seems on the first view more difficult to solve.*

[1] In [188] (Theorem 4.2) an LMI of the form (2.3.12) is discussed for the case $N = 2$. In this case, the statement can also be inverted, i.e., we have $\mathcal{E}(Q_1) + \mathcal{E}(Q_2) \subseteq \mathcal{E}(Q)$ with $Q_1, Q_2 \in \mathbb{S}^n_{++}$ if and only if there exists multipliers $\lambda_1, \lambda_2 \geq 0$ for which the LMI (2.3.12) can be satisfied. However, it is important to note that this statement is in general not true for $N > 2$.

However, problem (2.3.14) *can be regarded as a geometric programming problem. This can be seen if we apply a variable substitution of the form* $\lambda_i := e^{\nu_i}$ *such that we obtain the equivalent convex optimization problem*

$$\inf_{\nu \in \mathbb{R}^n} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} c^T Q_i \, c \, e^{\nu_j - \nu_i} \; . \tag{2.3.15}$$

In later chapters we will encounter the situation where the form (2.3.14) *is more appropriate. Thus, we keep in mind that problem* (2.3.14) *is in general not convex but a geometric programming problem which can equivalently be transformed into a convex (but not strictly convex) problem of form* (2.3.15).

Note that the set $\{ \mathcal{F}^+(\lambda) \mid \lambda \in \mathbb{D}^+ \}$ of ellipsoidal outer approximations which can be generated with the technique from Theorem 2.4 does not contain all possible outer ellipsoidal approximations, although it contains for every direction c an associated ellipsoidal outer approximation which touches the set $\mathcal{F}$. For example, if we search for a $\lambda^* \in \mathbb{D}^+$ for which the associated ellipsoid $\mathcal{F}^+(\lambda^*)$ has a minimum volume, this does not imply that we have found a minimum volume ellipsoid containing the set $\mathcal{F}$, although we might hope that $\mathcal{F}^+(\lambda^*)$ is a good approximation for that purpose. Unfortunately, it is in general difficult to construct a family of outer approximations which is not only tight, but does also contain all outer approximations with a given structure.

In order to extend our consideration of outer approximations of sums of ellipsoids, we discuss a second more powerful but also more expensive way to parameterize the outer approximation. The idea is to write the support function as

$$V(c) \;\; = \;\; \max_{x_1,\ldots,x_N} \; c^T \left(\sum_{i=1}^{N} x_i \right) \quad \text{s.t.} \quad x_i x_i^T \preceq Q_i \quad \text{for all} \;\; i \in \{1, \ldots, N\} \; .$$

If we dualize the above convex maximization problem into a minimization problem, we need a matrix valued multiplier $\Lambda \in \mathbb{S}_+^n$ finding

$$V(c) \;\; = \;\; \inf_{\Lambda \succ 0} \; \sum_{i=1}^{N} \frac{c^T Q_i^{\frac{1}{2}} \Lambda_i^{-1} Q_i^{\frac{1}{2}} c}{4} + \sum_{i=1}^{N} \mathrm{Tr}\left(\Lambda_i\right) \; .$$

Now, we can apply the same strategy as above which leads to following result:

Theorem 2.5: *Let us define the set* $\mathbb{D}^+ \subseteq \left(\mathbb{S}_{++}^n\right)^N$ *of feasible parameters by*

$$\mathbb{D}^+ \; := \; \left\{ \Lambda \in \left(\mathbb{S}_{++}^n\right)^N \;\middle|\; \frac{1}{n} \sum_{i=1}^{N} \mathrm{Tr}\left(\Lambda_i\right) \; \leq \; 1 \right\} \; .$$

Now, we have an inclusion of the form

$$\forall \Lambda \in \mathbb{D}^+ : \qquad \mathcal{F} \;=\; \sum_{i=1}^{N} \mathcal{E}(Q_i) \;\;\subseteq\;\; \mathcal{E}\left(\sum_{i=1}^{N} Q_i^{\frac{1}{2}} \Lambda_i^{-1} Q_i^{\frac{1}{2}} \right) \;=\; \mathcal{F}^+(\Lambda)\,.$$

This parameterized outer approximation is tight, i.e., we have $\mathcal{F} \;=\; \bigcap_{\Lambda \in \mathbb{D}^+} \mathcal{F}^+(\Lambda)\,.$

Note that Theorem 2.5 is more general than Theorem 2.4 in the sense that an application of Theorem 2.5 with $\Lambda_i := \lambda_i I$ yields the statement of Theorem 2.4.

Inner Approximations of Sums of Ellipsoids

In the next step we are looking for inner ellipsoidal approximations of an absolutely convex set of the form $\mathcal{F} = \sum_{i=1}^{N} \mathcal{E}(Q_i)$. Here, we follow the strategy which has been outlined above, i.e., we start with a computation of the associated polar set:

$$\begin{aligned} \mathcal{F}^* \;=\; \left[\sum_{i=1}^{N} \mathcal{E}(Q_i) \right]^* \;&=\; \left\{ y \in \mathbb{R}^n \;\middle|\; \max_{x \in \sum_i \mathcal{E}(Q_i)} \; y^T x \;\leq\; 1 \right\} \\ &=\; \left\{ y \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{N} \sqrt{y^T Q_i y} \;\leq\; 1 \right\}. \end{aligned} \tag{2.3.16}$$

In the next step, we proceed by computing the support function of the set $\mathcal{F}^*$ planning to find an outer approximation of the polar set. This support function is for all directions $c \in \mathbb{R}^n$ given by

$$V^*(c) \;:=\; \max_{y} \; c^T y \quad \text{s.t.} \quad y \in \mathcal{F}^*\,.$$

Here, we assume that the matrices $Q_1, \ldots, Q_N \in \mathbb{S}^n_{++}$ are invertible such that the maximum exists. Using the above representation for $\mathcal{F}^*$ the function $V^*(c)$ turns out to be the optimal value of a second order cone program (SOCP). In order to write this SOCP in standard form we introduce a slack variable $x \in \mathbb{R}^N$:

$$V^*(c) \;=\; \max_{x,\,y} \; c^T y \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^{N} x_i \;\leq\; 1 \\ \left\| Q_i^{\frac{1}{2}} y \right\|_2 \;\leq\; x_i \quad \text{for all } \; i \in \{1, \ldots, N\}\,. \end{cases}$$

As for example the point $x_i = \frac{1}{N+1}$ (with $i \in \{1, \ldots, N\}$) together with $y = 0$ yields a strictly feasible point (Slater's condition), we can express $V^*(c)$ via the following dual second order cone program with variables $\mu \in \mathbb{R}$, $\chi_i \in \mathbb{R}^n$:

$$V^*(c) \;=\; \min_{\mu,\, \chi_1, \ldots, \chi_N} \;\; \mu \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} \sum_{i=1}^N Q_i^{\frac{1}{2}} \chi_i & = & c \\ \| \chi_i \|_2 & \leq & \mu \quad \text{for all } \; i \in \{1, \ldots, N\} \, . \end{array} \right.$$

As the matrices Q_i are assumed to be invertible, we observe inductively that in the optimal solution $(\, \mu^*, \, \chi_1^*, \, \ldots, \, \chi_N^* \,)$ of the above dual SOCP, all constraints must be active, i.e., we have

$$\mu^* \;=\; \| \, \chi_1^* \, \|_2 \;=\; \ldots \;=\; \| \, \chi_N^* \, \|_2 \;\; .$$

In other words, there exists orthogonal matrices $S_1^*, \ldots, S_N^* \in \mathbb{R}^{n \times n}$ with $S_i^T S_i \;=\; I$ such that $\chi_i^* = S_i \lambda$ for all $\; i \in \{1, \ldots, N\}$ and some given common vector $\lambda \in \mathbb{R}^n$ with $\| \lambda \| = \mu^*$. Using this change of variables, we can transform the above dual problem further:

$$V^*(c) \;=\; \min_{\lambda,\, S_1, \ldots, S_N} \;\; \| \, \lambda \, \|_2 \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} \sum_{i=1}^N Q_i^{\frac{1}{2}} S_i \lambda & = & c \\ S_i \, S_i^T & = & I \quad \text{for all } \; i \in \{1, \ldots, N\} \, . \end{array} \right.$$

Note that we may assume that the matrix of the form $\sum_{i=1}^N Q_i^{\frac{1}{2}} S_i$ is invertible, as the set $\sum_i \mathcal{E}(Q_i)$ is assumed to be non-degenerate, i.e., we use that $Q_i \succ 0$ for all $i \in \{1, \ldots, n\}$. Thus, we can define a positive definite matrix $\; P(S_1, \ldots, S_n) \in \mathbb{R}^{n \times n} \;$ as

$$P(S_1, \ldots, S_n) \;:=\; \left(\sum_{i=1}^N Q_i^{\frac{1}{2}} S_i \right) \left(\sum_{i=1}^N Q_i^{\frac{1}{2}} S_i \right)^T \;\; .$$

Using this notation, we find

$$V^*(c) \;=\; \min_{\lambda,\, S_1, \ldots, S_N} \;\; \sqrt{c^T \, [P(S_1, \ldots, S_n)]^{-1} \, c} \quad \text{s.t.} \quad S_i \, S_i^T \;=\; I \; . \tag{2.3.17}$$

with $i \in \{1, \ldots, N\}$. As this result holds for all vectors c, we can conclude that all ellipsoids of the form $\; \mathcal{E}(\, [P(S_1, \ldots, S_n)]^{-1} \,) \;$ are ellipsoidal outer approximations of the set $\mathcal{F}^*$ independent of how we choose the orthogonal matrices $S_1, \ldots, S_N$. Thus, if we take polar sets, we obtain an ellipsoidal inner approximation.

Theorem 2.6 (Ellipsoidal Inner Approximations): *Let us first define the set of parameters $\mathbb{D}^- \subseteq \left(\mathbb{R}^{n \times n} \right)^N$ to be composed of sub-orthogonal matrices, i.e., we use a definition of the form*

$$\mathbb{D}^- \;:=\; \left\{ \, S \in \left(\mathbb{R}^{n \times n} \right)^N \;\middle|\; S_i \, S_i^T \;\preceq I \quad \text{for all } \; i \in \{1, \ldots, N\} \, \right\} \, .$$

The ellipsoid $\mathcal{E}(\, P(S_1, \ldots, S_n)\,)$ *is for every set of matrices* $S \in \mathbb{D}^-$ *an inner approximation of the set* $\sum_{i=1}^N \mathcal{E}(Q_i)$ *, i.e., we have*

$$\forall S \in \mathbb{D}^- : \quad \mathcal{F} \;=\; \sum_{i=1}^{N} \mathcal{E}(Q_i) \;\;\supseteq\;\; \mathcal{E}\left(\left(\sum_{i=1}^{N} Q_i^{\frac{1}{2}} S_i \right) \left(\sum_{i=1}^{N} Q_i^{\frac{1}{2}} S_i \right)^T \right) \;=:\; \mathcal{F}_-(S)\,.$$

Here, the matrices $Q_1, \ldots, Q_n \in \mathbb{S}_+^n$ *are not necessarily invertible. Finally, the inner approximation is tight, i.e., we have* $\mathcal{F} \;=\; \bigcup\limits_{S \in \mathbb{D}^-} \mathcal{F}_-(S)$ *.*

Proof: In the case that the matrices $Q_1, \ldots, Q_N$ are invertible, the statement follows from equation (2.3.17) as the inclusion $\mathcal{F}^* \;\subseteq\; \mathcal{E}(\,[P(S_1, \ldots, S_n)]^{-1}\,)$ implies

$$\mathcal{F} \;=\; (\mathcal{F}^*)^* \;\supseteq\; \mathcal{E}(\,[P(S_1, \ldots, S_n)]^{-1}\,)^* \;=\; \mathcal{E}(\, P(S_1, \ldots, S_n)\,)$$

for all orthogonal matrices $S_1, \ldots, S_N$. Here, we can additionally use that the ellipsoid $\mathcal{E}(\, P(S_1, \ldots, S_n)\,)$ can only get smaller if we replace an orthogonal matrices S_i with a sub-orthogonal matrix.

Finally, it remains to be shown that the Theorem is also true if the matrices $Q_1, \ldots, Q_n$ are not invertible. Here, the strategy is analogous to the proof of Theorem 2.4, i.e., we can add small regularizations terms εI to the matrices $Q_1, \ldots, Q_n$ and verify that the above inclusion holds in the limit sense for vanishing regularization $\epsilon \to 0$. □

Chapter 3

Robust Nonconvex Optimization

3.1 Formulation of Semi-Infinite Optimization Problems

In this section we start with an introduction to nonlinear semi-infinite or min-max optimization. Here, we are interested in robust counterpart problems of the form

$$\min_{x\in\mathbb{R}^n} \quad V_0(x) \quad \text{s.t.} \quad V_i(x) \leq 0 \quad \text{for all } i \in \{1,\ldots,m\}\,, \tag{3.1.1}$$

where the robust counterpart functions $V_i : \mathbb{R}^{n_x} \to \mathbb{R}$ are of the form

$$V_i(x) := \max_{w\in W} F_i(x,w)\,.$$

with twice continuously differentiable functions $F_0, F_1, \ldots, F_n : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}$ depending on an optimization variable $x \in \mathbb{R}^{n_x}$ and on an uncertain parameter w, which is bounded by a compact set $W \subseteq \mathbb{R}^{n_w}$. The main difference to the considerations from the previous chapter is that we do not require any convexity assumption on the functions F_i, i.e., we have in general neither lower-level nor upper-level convexity. As mentioned, problems of this form are usually called non-convex semi-infinite optimization problems as we could reformulate the robust counterpart problem into a standard minimization problem if we would allow to formulate infinitely many constraints. We also note that the set W is assumed to be independent of x. In the case that the uncertainty set W can itself be shaped by the optimization variable x, i.e., for the case that we have a generalized semi-infinite optimization problem, we can in most of the practically relevant cases reformulate it into a standard min-max problem with fixed uncertainty set by changing variables.

Example 3.1: Let us consider an illustrative robust counterpart optimization problem of the form (3.1.1) for the case that the objective function is given as $F_0(x,w) := x_1^2 + x_2^2$ while the function

$$F_1(x,w) := f_1(x_1+w_1) - (x_2+w_2) = e^{x_1+w_1} - (x_2+w_2)$$

denotes an uncertain constraint function. Here, the uncertainty is assumed to satisfy $\|w\|_2 \leq 1$. In other words, we are interested in a min-max problem of the form

$$\min_x \; x_1^2 + x_2^2 \quad \text{s.t.} \quad \max_{\|w\|_2 \leq 1} e^{x_1+w_1} - (x_2+w_2) \leq 0$$

In order to interpret this problem graphically, we define the nominally feasible set $\mathcal{F}_\mathsf{n} \subseteq \mathbb{R}^2$ as

$$\mathcal{F}_\mathsf{n} := \left\{ x \in \mathbb{R}^2 \mid f_1(x_1) - x_2 \leq 0 \right\}$$

The above min-max problem asks for a point x with minimal norm such that a ball with radius 1 centered at the point x is completely contained in the nominally feasible set $\mathcal{F}_\mathsf{n}$ as visualized in the left part of Figure 3.1. Note that the problem is upper-level convex as the objective F_0 and the constraint function F_1 are for all w convex functions in x (c.f. Lemma 2.1). However, the lower level maximization problem of the form

$$\max_{\|w\|_2 \leq 1} e^{x_1+w_1} - (x_2+w_2) \tag{3.1.2}$$

is a non-convex optimization problem.

It is interesting to remark that for the special case in Example 3.1 every local maximizer of the sub-problem is also a global maximizer. Intuitively, this can already be seen by looking at Figure 3.1: the uncertainty ball seems to have a larger curvature than the e-function. In order to show this mathematically, we directly generalize the above example for functions of the form

$$F(x,w) = f(x_1+w_1) - (x_2+w_2)\,,$$

where $x \in \mathbb{R}^{n+1}$, $x_1 \in \mathbb{R}^n$, $x_2 \in \mathbb{R}$ and $x := (x_1^T, x_2)^T$. In this context, the uncertainty vector $w := (w_1^T, w_2)^T \in \mathbb{R}^{n+1}$ is assumed to satisfy $\|w\| = \|w_1\|_2^2 + w_2^2 \leq 1$.

Definition 3.1: *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function. Now, we define a curvature function $\kappa : \mathbb{R}^n \to \mathbb{R}_+$ of the function f for all $x_1 \in \mathbb{R}^n$ as the unique solution of the LMI*

$$\kappa(x) := \operatorname*{argmin}_{\kappa \geq 0} \kappa \quad \text{s.t.} \quad f''(x) \preceq \kappa \sqrt{1 + \| f'(x) \|_2^2} \left[I + f'(x) f'(x)^T \right] .$$

Here, $f'(x) \in \mathbb{R}^n$ and $f''(x) \in \mathbb{S}^n$ denote the first and second order derivative of f, respectively.

The reason why this curvature function is useful in our context can be stated as follows:

Lemma 3.1: *Let $X \subseteq \mathbb{R}^n$ be an open and convex set and $Y := X + \mathcal{E}(I)$ the Minkowski sum of X and the unit ball $\mathcal{E}(I)$. Moreover, let $f : Y \to \mathbb{R}$ be a twice continuously differentiable function and $\kappa : Y \to \mathbb{R}_+$ the associated curvature function given by Definition 3.1. If we have $\kappa(y) < 1$ for all $y \in Y$, then the optimization problem*

$$\max_{\|w\|_2 \leq 1} f(x_1 + w_1) - (x_2 + w_2) \tag{3.1.3}$$

admits for all $x_1 \in X$ (and all $x_2 \in \mathbb{R}$) a unique local maximizer $w^*(x)$, i.e., there cannot be local maxima which are not global.

Proof: First of all, the constraint $\|w\|_2 \leq 1$ must be active in the optimal solution of the problem (3.1.3) – otherwise, we could keep w_1 and make w_2 smaller. In other words, the optimization problem (3.1.3) is equivalent to an unconstrained optimization problem of the form

$$\max_{w_1} \; f(x_1 + w_1) - \left(x_2 - \sqrt{1 - w_1^2} \right) .$$

Here, we can exclude maxima at the extreme points $w_1 = 1$ or $w_1 = -1$, as may use $f'(x_1 + w_1) < \infty$ for any fixed $x_1 \in X$, i.e., we only have to look for local maxima w_1^* which are inside the open interval $(-1, 1)$. Consequently, we analyze the stationarity condition

$$f'(x_1 + w_1) - \frac{w_1}{\sqrt{1 - w_1^2}} \; = \; 0 \, ,$$

which can equivalently be written as

$$w_1 \; = \; \frac{f'(x_1 + w_1)}{\sqrt{1 + \|f'(x_1 + w_1)\|_2^2}} \, . \tag{3.1.4}$$

The main idea of the proof is to verify that the equation

$$g(w_1) \; := \; \frac{f'(x_1 + w_1)}{\sqrt{1 + \|f'(x_1 + w_1)\|_2^2}} - w_1 \; = \; 0$$

admits at most one solution for w_1. In order to show this, we compute the derivative g' of the auxiliary function g with respect to w_1 finding

$$\begin{aligned} g'(w_1) &= \frac{f''(y)}{\sqrt{1+\|f'(y)\|_2^2}} - \frac{f'(y)f'(y)^T f''(y)}{(1+\|f'(y)\|_2^2)^{\frac{3}{2}}} - I \\ &= \left[I - \frac{f'(y)f'(y)^T}{1+\|f'(y)\|_2^2} \right] \frac{f''(y)}{\sqrt{1+\|f'(y)\|_2^2}} - I \end{aligned}$$

Here, we have introduced the short hand $y := x_1 + w_1 \in Y$ in order to simplify the notation. Note that an application of the Sherman-Morrison formula yields

$$\begin{aligned} g'(w_1) &= \left[I + f'(y)f'(y)^T \right]^{-1} \frac{f''(y)}{\sqrt{1+\|f'(y)\|_2^2}} - I \\ &\preceq \kappa(y) I - I \prec 0 \,. \end{aligned} \tag{3.1.5}$$

In the last step, the definition of κ as well as the requirement $\kappa(y) < 1$ has been used. At this point, the proof is complete, as the function g has a negative definite derivative on the domain of our interest and consequently this function can have at most one root. □

Coming back to the special case from Example 3.1, we can compute the curvature function κ for the e-function in order to illustrate the above Lemma. In this special case, we find

$$\forall x \in \mathbb{R}: \qquad \kappa(x) = \frac{e^x}{(1+e^{2x})^{\frac{3}{2}}} \leq \frac{2}{\sqrt{27}} < 1 \,.$$

In other words, the curvature of the uncertainty ball, which is 1 in this example, is larger than the curvature of the function f_1, which is at most $\frac{2}{\sqrt{27}}$. Thus, the result of Lemma 3.1 can be applied: we have proven that the non-convex lower-level maximization problem (3.1.2) has exactly one unique local maximizer.

Example 3.2: Let us consider the min-max problem

$$\min_x \quad \left(x_1 - \frac{1}{2}\right)^2 + x_2^2 \quad \text{s.t.} \quad \begin{cases} 0 \geq \max\limits_{\|w\|_2 \leq \frac{1}{3}} & 1 - (x_1+w_1)^2 - (x_2+w_2)^2 \\ 0 \geq \max\limits_{\|w\|_2 \leq \frac{1}{3}} & \log(x_1+w_1) - (x_2+w_2) \\ 0 \geq \max\limits_{\|w\|_2 \leq \frac{1}{3}} & -(x_1+w_1) \end{cases}$$

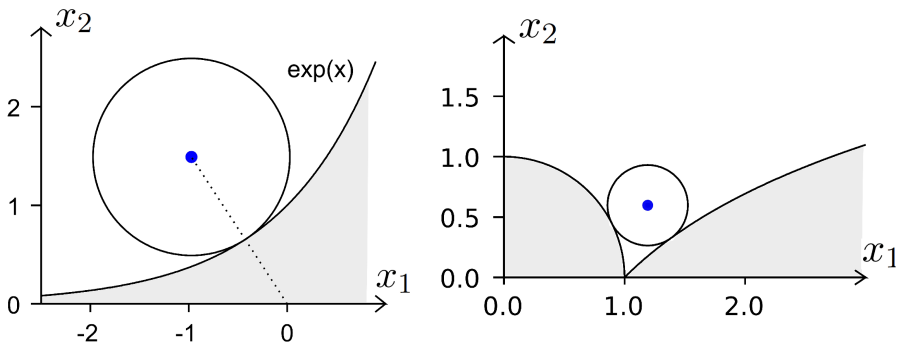


Figure 3.1: Left: a visualization of the optimal solution to the semi-infinite optimization problem from Example 3.1. Right: a visualization of the optimal solution to the semi-infinite optimization problem from Example 3.2.

In contrast to the previous example, we have now lower level convexity while the upper level problem turns out to be non-convex. The graphical interpretation is analogous to the previous example. The optimal solution visualized in the right part of Figure 3.1, where the uncertainty ball has the radius $\frac{1}{3}$. Finally, we note that despite the lower level convexity we cannot explicitly apply the dual reformulation strategy as the dual function cannot explicitly be written down, i.e., we would need functions which can theoretically be constructed but which are usually not available in standard programming libraries. In the optimal solution only the first two constraints are active.

Example 3.3: We consider a quadratic min-max problem of the form (with $\Sigma \in \mathbb{S}^2_{++}$)

$$\min_x \; x_2 \quad \text{s.t.} \quad \max_{w^T \Sigma^{-1} w \leq 1} \; (x_1 + w_1)^2 - (x_2 + w_2) \; \leq \; 0 \; . \tag{3.1.6}$$

This problem is upper level convex but not lower level convex. In the left part of Figure 3.2 the example is visualized for the case that the uncertainty set is ellipsoidal. Here, we use

$$\Sigma^{-1} \; := \; \begin{pmatrix} 0.8 & -0.6 \\ -0.6 & 0.8 \end{pmatrix} \; .$$

Note that the ellipsoid touches the boundary of the nominally feasible set twice illustrating that there are two local maxima in the non-convex lower level problem. Recall that despite the non-convexity of the lower level problem, the above min-max problem can be solved by convex optimization applying the S-procedure as discussed in Section 2.2. If we introduce

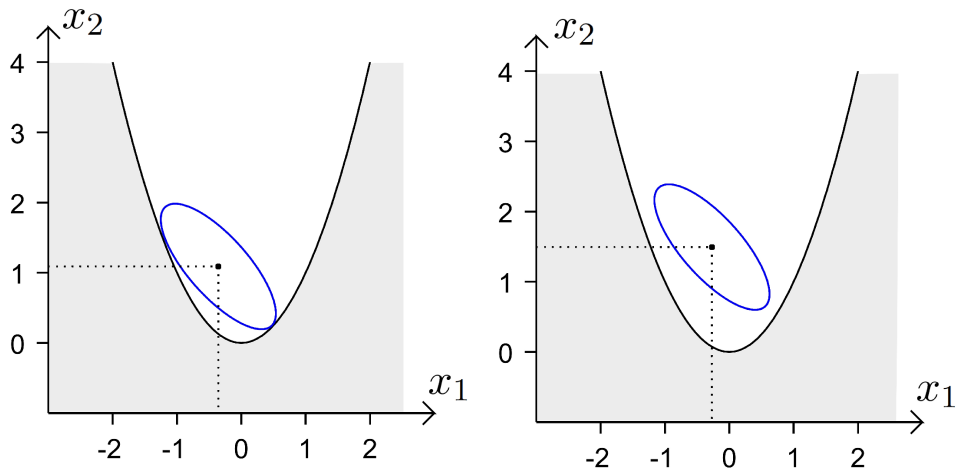


Figure 3.2: Left: a visualization of the optimal solution of the semi-infinite optimization problem (3.1.6) from Example 3.3, which has been found by solving a semi-infinite programming problem of the form (3.1.7). Right: a visualization of a sub-optimal, conservative solution of the semi-infinite optimization problem (3.1.6), which has been found by the linear approximation strategy as discussed within Example 3.5.

the short-hands

$$Q := \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad q(x_1) := \begin{pmatrix} 2x_1 \\ -1 \end{pmatrix}$$

we can compute the unique optimal solution of the min-max problem (3.1.6) numerically by solving an equivalent semi-definite programming problem of the form

$$\min_{x,\lambda} \; x_2 \;\; \text{s.t.} \;\; \begin{pmatrix} x_2 - \lambda & q(x_1)^T & x_1 \\ q(x_1) & \lambda\Sigma - Q & 0 \\ x_1 & 0 & 1 \end{pmatrix} \succeq 0 \,. \tag{3.1.7}$$

The associated result $x^* \approx (-0.35\,,\, 1.08)^T$ corresponds to the center of the ellipsoidal uncertainty region which is shown in the left part of Figure 3.2.

Example 3.4: Let us consider an example which is in contrast to the previous cases non-convex in both the lower-level maximization as well as the upper level minimization problem:

$$\min_{x} \; -x \quad \text{s.t.} \quad \max_{w^2 \leq 1} \; \sin(x\,w) \leq \frac{1}{2}$$

It is quite trivial to solve this min-max optimization explicitly: the optimal solution is at $(x^*, w^*) = \left(\arcsin\left(\frac{1}{2}\right), 1\right)$. Nevertheless, it might be helpful to keep the above problem as a simple guiding example in mind in order to verify and understand the following considerations in a better way.

The four examples above are certainly not representable for the class of problems we are addressing. These four examples are all rather simple. None of them looks hopeless or even numerically unsolvable. On the other hand, we know only for the min-max problem from Example 3.3 how to reformulate it explicitly into a "plain" convex optimization problem, while the other three examples outline a class of problems which require both theoretical and numerical techniques which are beyond the traditional framework of standard convex formulations. Thus, we first ask the question whether we can replace the convexity requirement with less restrictive assumptions on the functions F_i such that we can still develop efficient and reliable optimization algorithms for the considered class of problems. The main difficulty is that we always have to find global maximizers of the lower level maximization problems as we cannot guarantee feasibility otherwise. This implies that we cannot directly employ local search routines for the lower-level problem, which are typically employed in the field of nonconvex optimization.

For the special type of semi-infinite optimization problems from Example 3.1, we have seen that the curvature of the constraint function and objective can in some cases be used to guarantee that a non-convex maximization problem has only one unique local maximizer. However, we have also discussed within Example 3.3 that we might even be able to find tractable reformulations of a robust optimization problem for which the non-convex lower-level problem has two or more global maxima in the optimal solution. We propose to require an assumption on the second derivatives of the functions F_i:

Assumption 3.1: *Let us assume that we have for each $i \in \{0, \ldots, n\}$ a twice continuously differentiable and non-negative function $\overline{\lambda}_i : \mathbb{R}^{n_x} \to \mathbb{R}_+$ which satisfies the inequality*

$$\forall w \in W: \quad \lambda_{\max}\left(\frac{\partial^2}{\partial w^2} F_i(x, w)\right) \leq 2\,\overline{\lambda}_i(x)\,, \tag{3.1.8}$$

i.e., the maximum eigenvalue of the Hessian of F_i with respect to w is for all $w \in W$ bounded by the function $2\,\overline{\lambda}_i$.

Note that there exist numerical techniques from the field of global optimization [27, 99, 181], which are able to provide interval bounds on the eigenvalues of the Hessian matrix of a

given function as required in the above assumption. Nevertheless, the above assumption is still questionable, as it is in practice often not clear how we can obtain such functions $\overline{\lambda}_i$ if the suggested global numerical interval methods are too expensive to be applied. However, once we accept this assumption, we are able to develop efficient, derivative based robust optimization algorithms for the case $n_w \gg 1$. This is the aim of this and the following Chapter 4. In the following considerations the case of lower level convexity will always trivially be included, as we can employ the trivial choice $\overline{\lambda}_i(x) = 0$ if all functions F_i are concave in w for all x.

Finally, we note that the above assumption can be generalized by requiring the existence of a twice continuously and matrix valued positive semi-definite function $\overline{\Lambda}_i : \mathbb{R}^{n_x} \to \mathbb{S}^n_+$ which satisfies the inequality

$$\forall w \in W : \quad \frac{\partial^2}{\partial w^2} F_i(x,w) \preceq 2\,\overline{\Lambda}_i(x)\,. \tag{3.1.9}$$

The following consideration can be extended to this case by rescaling the variable w if necessary.

3.2 Convexification of Robust Counterparts

Due to the fact that the lower level maximization problems must be solved globally, the exact robust counterpart functions $V_i(x)$ within the problem formulation (3.1.1) can often only approximately be evaluated, as it is typically very expensive to solve non-convex optimization problems globally. In the following, we concentrate on convexification methods which can be employed in order to replace the functions V_i with a conservative approximation. In the following section, we review linearization techniques which have been developed in [71, 123, 133, 174]. After this, more advanced convexification techniques which are based on Lagrangian duality are discussed.

Approximate Robust Counterparts based on Linearization

Note that Assumption 3.1 enables us to construct conservative approximations of the robust counterpart functions V_i in the optimization problem (3.1.1). One method to

obtain such an approximation is linearization. Employing a Taylor expansion we find

$$\begin{aligned} V_i(x) &= \max_{w\in W} F_i(x,w) \\ &\leq \max_{v,w\in W} F_i(x,0) + \frac{\partial F_i(x,0)}{\partial w} w + \frac{1}{2} w^T \left(\frac{\partial^2}{\partial w^2} F_i(x,v) \right) w \\ &\leq \max_{w\in W} F_i(x,0) + \frac{\partial F_i(x,0)}{\partial w} w + \overline{\lambda}_i(x)\, w^T w \,. \end{aligned} \tag{3.2.1}$$

Note that the uncertainty set W can for example be modeled as an ellipsoidal set. In order to briefly discuss this case, we assume here for simplicity that the uncertainty set is a unit ball:

$$w \in \mathcal{E}(I) := \left\{ v \in \mathbb{R}^{n_w} \mid v^T v \leq 1 \right\} .$$

For theoretical considerations, this assumption is not excessively restrictive. For example, if we have a set W for which we can find a twice continuously differentiable and surjective map $\varphi(\cdot) : \mathcal{E}(I) \to W$, we can always reformulate the problem replacing in the functions $F_0, F_1, \ldots, F_n$ the variable w by $\varphi(w)$. However, it is also possible to model W as an intersection of ellipsoids – we will later come back to this case.

Now, for the case $W = \mathcal{E}(I)$ we can explicitly solve the convex problem (3.2.1) finding the overestimate (compare also with Example 2.1):

$$V_i(x) \leq F_i(x,0) + \left\| \frac{\partial F_i(x,0)}{\partial w} \right\|_2 + \overline{\lambda}_i(x) \,. \tag{3.2.2}$$

Recall that $\|\cdot\|_2 : \mathbb{R}^{n_w} \to \mathbb{R}$ denotes the Euclidean norm.

Definition 3.2: *We define the best conservative first order approximation $J_i : \mathbb{R}^{n_x} \to \mathbb{R}$ associated with the i-th lower level maximization problem by*

$$\forall x \in \mathbb{R}^{n_x} : \quad J_i(x) := F_i(x,0) + \left\| \frac{\partial F_i(x,0)}{\partial w} \right\|_2 + \overline{\lambda}_i(x) \,. \tag{3.2.3}$$

The above definition is motivated by the observation that once we linearize the function F_i at $w = 0$ allowing neither to compute the gradient of F_i at any other point nor to compute any second order term, J_i is the smallest conservative approximation that we can obtain by using Assumption 3.1 only without having any further information on the function F_i.

Note that e.g. in [71, 123, 133, 174] the functions J_i have been used suggesting to solve the approximate robust counterpart problem

$$\begin{aligned} \min_{x \in \mathbb{R}^{n_x}} \quad & J_0(x) \\ \text{subject to} \quad & J_i(x) \leq 0 \quad \text{for all } i \in \{1, \ldots, n\} \end{aligned} \tag{3.2.4}$$

instead of the original exact robust counterpart problem (3.1.1). In this context, we should be aware of the fact that we need to compute at least first order derivatives of the functions F_i in order to evaluate the associated functions J_i. As originally proposed in [71], it is desirable to use automatic differentiation [108, 112] for that purpose. One of the motivations for automatic differentiation is that the accuracy of numerical differentiation is often not sufficient, as for example an exact Hessian sequential quadratic programming method applied to problem (3.2.4) would already require to evaluate third order derivatives. Moreover, we have to distinguish two cases:

- The first case is that we have only a few uncertain variables n_w, but many constraint functions $m \gg n_w$. In this situation, it is usually better to use the forward mode of automatic differentiation in order to compute the terms $\frac{\partial F(x,0)}{\partial w}$ which are needed for the evaluation of the functions J_i.
- In the second case, if we have $m \ll n_w$, it is typically more efficient to use the backward mode of automatic differentiation, such that only m backward sweeps are needed to compute the required partial derivative $\frac{\partial F(x,0)}{\partial w}$.

The following example discusses an application of the linear approximation strategy but also outlines its conservatism and limits of accuracy:

Example 3.5: Let us once more consider the problem from Example 3.3 where the robust worst-case constraint is of the form

$$\max_{w^T Q^{-1} w \leq 1} \quad (x_1 + w_1)^2 - (x_2 + w_2) \;\leq\; 0\,. \tag{3.2.5}$$

Denoting the Cholesky decomposition of the matrix Q^{-1} as $RR^T = Q^{-1}$, the approximate counterpart formulation based on linearization can be written as

$$\min_x \quad x_2 \quad \text{s.t.} \quad J_1(x) \;=\; x_1^2 - x_2 + \left\| \begin{pmatrix} 2R_{1,1}x_1 - R_{1,2} \\ -R_{2,2} \end{pmatrix} \right\|_2 + R_{1,1}^2 \;\leq\; 0\,.$$

Using the same values as in Example 3.3, we find the solution $\hat{x} \approx (-0.27, 1.49)$. As visualized in the right part of Figure 3.2, the solution is robustly feasible, as the ellipsoid is completely contained in the nominally feasible paraboloid. However, recall that the exact solution of the robust counterpart problem is at $x^* \approx (-0.35, 1.08)$ (cf. the left part of Figure 3.2). Thus, we have to pay 38% of optimality if we apply the linear approximation comparing the optimal value with the exact robust solution from Example 3.3.

The above example shows that the linear approximation strategy can lead to a tractable way of formulating an approximate robust counterpart problem. If we would know that we cannot do better, we could now accept this linear approximation strategy as a practical way to solve the robust optimization problem conservatively. However, the min-max optimization problem from Example 3.3 can be treated in a much more elegant way by reformulating it into a simple convex semi-definite programming (SDP) problem. Consequently, we have to ask whether we can exploit Assumption 3.1 in a more efficient way than the linearization strategy does. One way would be to consider second order Taylor expansions of the functions F_i in the uncertainty w such that we can cover the specific case in Example 3.3. However, in general second order derivatives are expensive to compute. In particular, if we want to solve the robust counterpart problem with derivative based optimization algorithms, we need derivatives of order three or even higher. In the following section we will discuss a strategy which avoids the limitations of the Taylor expansion based approaches, which leads to a better conservative approximation of the worst case, and which is exact for the case that we deal with quadratic forms as in Example 3.3.

A Worst Case Approximation based on the Dual Lagrange Function

In this section, we pick any $i \in \{0, \ldots, n\}$ and ask once more the question how we can compute an upper bound on the function $V_i(x)$ which is needed in robust counterpart formulations. As in the previous consideration, we still assume that $W = \mathcal{E}(I)$ is the unit ball. Recall that our only information about the function F_i is that Assumption 3.1 holds.

Let us consider the dual Lagrange function $D_i : \mathbb{R}^{n_x} \times \mathbb{R}_+ \to \mathbb{R}$, which is associated with the lower level maximization problem:

$$\begin{aligned} D_i(x, \lambda_i) &:= \max_{w_i} G_i(x, \lambda_i, w_i) \\ \text{with} \quad G_i(x, \lambda_i, w_i) &:= F_i(x, w_i) - \lambda_i w_i^T w_i + \lambda_i \, . \end{aligned} \tag{3.2.6}$$

Note that $D_i(x, \lambda_i)$ is an upper bound on $V_i(x)$ for all $\lambda_i \in \mathbb{R}_{++}$, i.e., we have

$$\forall x \in \mathbb{R}^{n_x} : \quad V_i(x) \ \leq \ \inf_{\lambda_i > 0} \ D_i(x, \lambda_i) \ .$$

For the case that the above inequality holds with equality we say that the strong duality condition is satisfied. This is for example the case if F_i is strictly concave in w.

So far, we have not solved the problem: we still need to solve the optimization problem (3.2.6) globally. However, an interesting observation is that we have

$$\forall x \in \mathbb{R}^{n_x} : \quad M_i(x) := \inf_{\lambda_i > \overline{\lambda}_i(x)} \ D_i(x, \lambda_i) \ \geq \ \inf_{\lambda_i > 0} \ D_i(x, \lambda_i) \ , \tag{3.2.7}$$

since we assume that $\overline{\lambda}_i$ is a non-negative function. Note that $D_i(x, \lambda_i)$ is for $\lambda_i \geq \overline{\lambda}_i(x)$ easier to evaluate in the sense that the function $G_i(x, \lambda_i, \cdot)$ is in this case concave. Thus, we know that every local maximum of the function $G_i(x, \lambda_i, \cdot)$ is also a global maximum as long as the condition $\lambda_i \geq \overline{\lambda}_i(x)$ is satisfied.

In order to get used to the notation, we note that the function M_i can also be written as

$$M_i(x) \ = \ \max_{w_i} H_i(x, w_i) \quad \text{s.t.} \quad \|w_i\|_2^2 \leq 1 \ , \tag{3.2.8}$$

where the functions $H_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}$ are defined as

$$H_i(x, w_i) \quad := \quad G_i(x, \overline{\lambda}(x), w_i) \quad \text{for all} \quad i \in \{0, \ldots, n\}.$$

Note that equation (3.2.8) is equivalent to the definition (3.2.7) of M_i. This can directly be seen by shifting the multiplier λ_i by $\overline{\lambda}_i(x)$. Recall that the main motivation for this construction is that the function H_i is concave in w_i.

Lemma 3.2: *The function M_i is an upper bound on V_i which can never be more conservative than the best linear approximation J_i. Hence, we have*

$$\forall x \in \mathbb{R}^{n_x} : \quad V_i(x) \ \leq \ M_i(x) \ \leq \ J_i(x) \ .$$

Proof: Note that by a Taylor expansion of the function G_i there exists a $v \in \mathbb{R}^{n_w}$ such that

$$G_i(x, \lambda_i, w_i) \quad = \quad F_i(x, 0) + \frac{\partial}{\partial w} F_i(x, 0) w_i + \frac{1}{2} w_i^T \left(\frac{\partial^2}{\partial w^2} F_i(x, v) - 2\lambda_i I \right) w_i + \lambda_i \ .$$

For the case $\lambda_i > \overline{\lambda}(x)$ the function G_i is strictly concave and we can maximize over w_i finding that for all $x \in \mathbb{R}^{n_x}$ and all $\lambda_i > \overline{\lambda}(x)$ the estimate

$$\begin{aligned}\max_{w_i} G_i(x,\lambda_i,w_i) &\leq \max_{v} F_i(x,0) \\ &\quad + \frac{1}{2}\left\| \left(2\lambda_i I - \frac{\partial^2}{\partial w^2} F_i(x,v)\right)^{-\frac{1}{2}} \frac{\partial F_i(x,0)}{\partial w}^T \right\|_2^2 + \lambda_i \\ &\leq F_i(x,0) + \frac{1}{4}\frac{1}{\left(\lambda_i - \overline{\lambda}(x)\right)} \left\| \frac{\partial F_i(x,0)}{\partial w} \right\|_2^2 + \lambda_i \qquad (3.2.9)\end{aligned}$$

is satisfied. Now, it follows that

$$\begin{aligned}M_i(x) &= \inf_{\lambda_i > \overline{\lambda}_i(x)} D_i(x,\lambda_i) \\ &\overset{(3.2.9)}{\leq} \inf_{\lambda_i > \overline{\lambda}_i(x)} F_i(x,0) + \frac{1}{4}\frac{1}{\left(\lambda_i - \overline{\lambda}(x)\right)} \left\| \frac{\partial F_i(x,0)}{\partial w} \right\|_2^2 + \lambda_i \\ &\overset{\text{AM-GM}}{=} F_i(x,0) + \left\| \frac{\partial F_i(x,0)}{\partial w} \right\|_2 + \overline{\lambda}_i(x) \\ &= J_i(x)\,.\end{aligned}$$

As the above consideration holds for all $x \in \mathbb{R}^{n_x}$ it follows with (3.2.7) that we have

$$\forall x \in \mathbb{R}^{n_x}: \quad V_i(x) \;\leq\; M_i(x) \;\leq\; J_i(x)\,,$$

which is the statement of the Lemma. □

Note that for the case that F_i is already concave in w, we have $M_i = V_i$, i.e., there is no duality gap and consequently no conservatism introduced. In order to see that the function M_i might also beyond concavity coincide with the exact function V_i, we formulate once more the tight version of the S-procedure for quadratic forms (cf. Theorem 2.1):

Lemma 3.3: *If the function F_i is a not necessarily concave quadratic in w given in the form*

$$F_i(x,w) \;=\; w^T Q(x) w + q(x)^T w + s(x)$$

with $\overline{\lambda}_i(x) := \max\{0, \lambda_{\max}(Q(x))\}$ *and* $Q(x)$ *symmetric, then the approximation function* M_i *is exact, i.e., we have*

$$\forall x \in \mathbb{R}^{n_x}: \quad V_i(x) = M_i(x) .$$

Proof: As we have only one single ellipsoidal uncertainty constraint, in this case the ball $W = \mathcal{E}(I)$, the Lemma follows immediately from the tight version of S-procedure. In other words, we can directly apply Theorem 2.1. □

Example 3.6: Let us once more regard Example 3.4. Here, the uncertain constraint function is given by $F_1(x,w) = \sin(x\,w)$. Thus, the exact robust counterpart function is given by

$$V_1(x) = \begin{cases} \sin(|x|) & \text{if } |x| \leq \frac{\pi}{2} \\ 1 & \text{otherwise.} \end{cases}$$

In order to apply our technique, we first need a Hessian upper bound, which is given by

$$\overline{\lambda}_1(x) := \begin{cases} \frac{x^2}{2}\sin(|x|) & \text{if } |x| \leq \frac{\pi}{2} \\ \frac{x^2}{2} & \text{otherwise.} \end{cases}$$

Now, the associated best linear approximation can be written as

$$J_1(x) = \begin{cases} |x| + \frac{x^2}{2}\sin(|x|) & \text{if } |x| \leq \frac{\pi}{2} \\ |x| + \frac{x^2}{2} & \text{otherwise.} \end{cases}$$

In Figure 3.3 the functions V_1 and J_1 are visualized as a dashed and a dotted line, respectively. The corresponding Lagrangian based overestimation M_1 is shown as a solid line. As we can see from Figure 3.3, we have $V_1(x) = M_1(x)$ for all x with $|x| \lessapprox 0.86$.

The above example illustrates that the relation $V(x) = M(x)$ does in general not globally hold. Nevertheless, in some cases equality can be shown within a local region. Let us formalize this idea within the following Lemma:

Lemma 3.4: *Let* $X \subseteq \mathbb{R}^{n_x}$ *be a subset of* $\mathbb{R}^{n_x}$ *as well as* $W := \{w \mid w^T w \leq \gamma^2\}$ *and*

$$G_i(x, \overline{\lambda}(x), w) := F_i(x,w) - \overline{\lambda}_i(x) w^T w + \gamma^2 \overline{\lambda}_i(x) .$$

If for all $x \in X$ *the function* $G_i(x, \overline{\lambda}(x), \cdot)$ *does not take a maximum on the open set* ***int***(W) *denoting the interior of* W*, then there is no duality gap. In other words, we have in this case*

$$\forall x \in X: \quad V_i(x) = M_i(x) := \inf_{\lambda_i > \overline{\lambda}_i(x)} \max_{w_i} G_i(x, \lambda_i, w_i) .$$

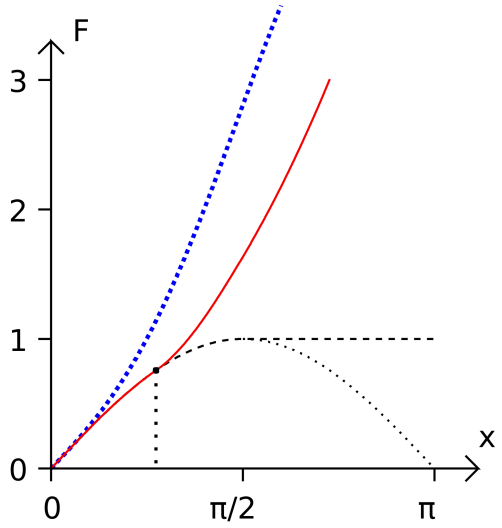


Figure 3.3: A visualization of the robust counterpart function V_1 (black dashed line) from Example 3.6, the associated best linear approximation function L_1 (blue dotted line), as well as the dual Lagrangian based approximation function M_1 (red solid line). Note that we have $V_1(x) \leq M_1(x) \leq J_1(x)$ for all $x \in \mathbb{R}$ and $V_1(x) = M_1(x)$ for all x with $|x| \lessapprox 0.86$.

Moreover, if X is compact and $\frac{\partial}{\partial w} F(x,0) \neq 0$ for all $x \in X$ then there exists a $\gamma^2 > 0$ such that the above condition is satisfied.

Proof: If the function $G_i(x, \overline{\lambda}(x), \cdot)$ does not take a maximum in the interior **int**(W) of the set W, it must take a maximum w^* on the boundary, where we have

$$G_i(x, \overline{\lambda}(x), w^*) \;=\; F_i(x, w^*)$$

for all $x \in X$. Thus, we can conclude that the function $M_i(x)$ is in this case coinciding with $V_i(x)$. The second statement of the Theorem follows from the stationarity condition

$$\frac{\partial}{\partial w} G_i(x, \overline{\lambda}_i(x), w^*) \;=\; \frac{\partial}{\partial w} F_i(x, w^*) - 2\overline{\lambda}_i(x) w^* \;,$$

which must be satisfied at every maximizer of $G_i(x, \overline{\lambda}_i(x), \cdot)$ which is in the interior of W. However, if γ^2 is sufficiently small and $\frac{\partial}{\partial w} F_i(x,0) \neq 0$ for all $x \in X$ then it follows from

the continuity of the function $\frac{\partial}{\partial w} F_i(x, \cdot)$ that we cannot satisfy the above stationarity condition. □

Summarizing the above result in words, we can state that under some mild regularity conditions, the approximation M_i is locally exact, i.e., if the uncertainty set is sufficiently small. Note that we can also find a global upper bound on the duality gap:

Theorem 3.1 (An Upper Bound on the Duality Gap): *With the same assumptions as in Lemma 3.4, i.e., $W := \{w \mid w^T w \leq \gamma^2\}$, we define a radius $\overline{\gamma}_i(x)$ as*

$$\overline{\gamma}_i(x) \ := \ \min_{w \in W} \left\| \left(\frac{\partial^2 F_i(x,w)}{\partial w^2} - 2\overline{\lambda}_i(x) I \right)^{-1} \frac{\partial F_i(x,0)}{\partial w} \right\|_2 .$$

The approximation function M_i always satisfies an inequality of the form

$$| M_i(x) - V_i(x) | \ \leq \ \overline{\lambda}_i(x) \max \left\{ 0, \gamma^2 - \overline{\gamma}_i(x)^2 \right\} .$$

In particular, if we have $\overline{\gamma}_i(x) \geq \gamma$, then there is no duality gap. Moreover, the term $\overline{\lambda}_i(x)\, \gamma^2$ is always an upper bound on the approximation error $| M_i(x) - V_i(x) |$.

Proof: Note that we can directly assume that there exists a stationary point $w^* \in W$ which satisfies

$$\frac{\partial}{\partial w} G_i(x, \overline{\lambda}_i(x), w^*) \ = \ \frac{\partial}{\partial w} F_i(x, w^*) - 2\overline{\lambda}_i(x) w^* .$$

If there is no such point, we can use Lemma 3.4 to show that $M_i(x) - V_i(x) \ = \ 0$. Otherwise, we can employ a Tailor expansion of the above stationarity condition as well as the relation

$$| M_i(x) - V_i(x) | \ = \ \overline{\lambda}_i(x) \left(\gamma^2 - (w^*)^2 \right) ,$$

which yields the statement of the Theorem. □

The above theorem is important in the sense that it yields an upper bound of the conservatism which is introduced by employing the approximate robust counterpart $M_i(x)$ instead of the exact value $V_i(x)$. For special classes of functions tighter sub-optimality estimates are possible.

Remark 3.1 (Limits of the Approximation): *In the worst case, Theorem 3.1 yields the upper bound $\overline{\lambda}_i(x)\,\gamma^2$. There are examples where this worst case occurs: let us consider the case that the function $F_i(x,w)$ with scalar uncertainty w has the unfortunate form $F_i(x,w) = w^4 - w^2$ with $\gamma = 1$. In this case, we find that the Hessian of F_i is given by*

$$\frac{\partial^2}{\partial w^2} F_i(x,v) = 12w^2 - 1 \,.$$

Consequently, the smallest upper bound on the Hessian matrix is given by $\overline{\lambda}_i(x) := 11$ finding $V_i(x) = 0$ and $M_i(x) = J_i(x) = 11$. For the case that functions F_i are - as above - polynomial in w there exist convexification techniques which do not suffer from conservatism but require to reformulate the polynomial maximization problem via linear matrix inequalities [116, 148, 215]. As such techniques are typically very expensive for larger dimensions n_w, we do not review them here, but refer to the work of Lasserre [149] and Parrilo [184] for more details.

Finally, we are interested in solving an approximate robust counterpart problem of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^{n_x}} \quad & M_0(x) \\ \text{subject to} \quad & M_i(x) \leq 0 \quad \text{for all } i \in \{1,\dots,n\} \,. \end{aligned} \tag{3.2.10}$$

Here, it cannot be recommended to solve the above problem with a standard nonlinear program (NLP-) solver, as evaluations of the functions M_i are expensive. Recall equation (3.2.8) where these functions are given as the optimal values of the maximization problems

$$M_i(x) = \max_{w_i} H_i(x,w_i) \quad \text{s.t.} \quad \|w_i\|_2^2 \leq 1 \,.$$

Moreover, due to possible active set changes, the functions M_i are in general not differentiable in x. In the following, we plan to develop an algorithm to solve problem (3.2.10) by taking the min-max structure explicitly into account. This algorithm will be worked out in this and the following chapter.

Generalizations for other Types of Uncertainty Sets

So far our analysis was based on the assumption that the uncertainty set is a simple ball in $\mathbb{R}^{n_w}$, which covers already many cases, for example ellipsoidal sets, if we allow to re-scale

the variable w. However, the Lagrangian dual relaxation strategy transfers in principle also for uncertainty sets which are modeled as intersections of ellipsoids, i.e., for the case that the uncertainty set is of the more general form

$$W \;=\; \left\{ w \in \mathbb{R}^{n_w} \;\middle|\; (w-\sigma_j)^T \, \Sigma_j \, (w-\sigma_j) \;\leq\; 1 \;\text{ with } j \in \{1,\ldots,N\} \right\} ,$$

where the matrices $\Sigma_1, \ldots, \Sigma_N \in \mathbb{S}_+^{n_w}$ and vectors $\sigma_1, \ldots, \sigma_N \in \mathbb{R}^{n_w}$ are assumed to be given. The argumentation is in principle analogous to the previous section, but the dual function $D_i : \mathbb{R}^{n_x} \times \mathbb{R}_+^N \to \mathbb{R}$ is now given by

$$D_i(x,\lambda_i) \;:=\; \max_{w_i} \; G_i(x,\lambda_i,w_i)$$

$$\text{with} \quad G_i(x,\lambda_i,w_i) \;:=\; F_i(x,w_i) - \sum_{j=1}^{N} \Big[\, \lambda_{i,j} \, (w_i-\sigma_j)^T \, \Sigma_j \, (w_i-\sigma_j) - \lambda_{i,j} \Big] \; .$$

In this case, it is most consistent to assume that we have a twice continuously and matrix valued positive semi-definite function $\overline{\Lambda}_i : \mathbb{R}^{n_x} \to \mathbb{S}_+^{n_w}$ which satisfies the inequality

$$\forall w \in W : \quad \frac{\partial^2}{\partial w^2} F_i(x,w) \;\preceq\; 2 \, \overline{\Lambda}_i(x) \; ,$$

such that the function G_i is known to be concave in w for all x and for all $\lambda_i \in \mathbb{R}_+^N$ which satisfy the semi-definite inequality

$$\overline{\Lambda}_i(x) \;\preceq\; \sum_{j=1}^{N} \lambda_{i,j} \Sigma_j \; .$$

Now, we define the approximate robust counterpart function $M_i : \mathbb{R}^{n_x} \to \mathbb{R}$ as

$$\forall x \in \mathbb{R}^{n_x} : \quad M_i(x) \;:=\; \inf_{\lambda_i} \; D_i(x,\lambda_i) \quad \text{s.t.} \quad \overline{\Lambda}_i(x) \;\preceq\; \sum_{j=1}^{N} \lambda_{i,j} \Sigma_j \; .$$

Note that this approximate robust counterpart function also satisfies

$$\forall x \in \mathbb{R}^{n_x} : \quad M_i(x) \;\geq\; \inf_{\lambda_i > 0} \; D_i(x,\lambda_i) \;\geq\; V_i(x) \; ,$$

which shows that M_i is an upper bound on the exact robust counterpart function V_i. A difference to the previous section is that it is now more difficult to find a priori bounds on the duality gap, i.e., the level of conservatism which is introduced by exchanging the exact

robust counterpart functions with their approximations M_i. However, for some special cases, where the function F_i is a quadratic form and the ellipsoids have a common center, i.e., $\sigma_1 = \ldots = \sigma_N$, upper bounds on the duality gap are known, as for example the argumentation of Nemirovski, Roos, and Terlaky [176] can be transferred.

If we employ the above framework, the original non-convex min-max problem can be approximated by a lower level convex min-max problem of the form

$$\begin{array}{ll} \min\limits_{x,\lambda \geq 0} & \max\limits_{w_0} \; G_0(x, \lambda_0, w_0) \\ \text{s.t.} & \left\{ \begin{array}{rcl} 0 & \geq & \max\limits_{w_i} \; G_i(x, \lambda_i, w_i) \\ 0 & \succeq & \overline{\Lambda}_i(x) - \sum_{j=1}^N \lambda_{i,j} \Sigma_j \quad \text{for all } i \in \{1, \ldots, n\} \; . \end{array} \right. \end{array} \tag{3.2.11}$$

Recall that the function G_i are concave in w, whenever $i \in \{0, \ldots, n\}$ while the pair (x, λ_i) is a feasible point of problem (3.2.11).

Remark 3.2 (Box Constraints): *Note that the above consideration includes the important case that the uncertainty set is a box constraint, as we may choose $N := n_w$, $\sigma_j := 0$, and $\Sigma_j := e_j e_j^T$, with e_j being the j-th unit vector in $\mathbb{R}^{n_w}$ with $j \in \{1, \ldots, N\}$. If we assume additionally that the given Hessian upper bound $\overline{\Lambda}_i(x)$ is diagonal, the semi-definite inequalities of the form*

$$0 \; \succeq \; \overline{\Lambda}_i(x) - \sum_{j=1}^N \lambda_{i,j} \Sigma_j$$

can equivalently be imposed as a standard inequality for the diagonal elements, as all non-diagonal entries of the matrices Σ_i and $\overline{\Lambda}_i(x)$ are equal to zero in this case. Note that such Lagrangian based relaxation strategies for box constrained uncertainties have been suggested in [100], where the so-called α-BB method is introduced. Note that in [100] the convexification method is combined with a branch-and-bound strategy and applied in the context of generalized semi-infinite programming for the case $n_w = 1$, i.e., for the case that W is a one dimensional interval. In this context, we also refer to [179], where the problem of maximizing a non-convex quadratic over a box is analyzed. More generally, we have the upper bound

$$\forall x \in \mathbb{R}^{n_x} : \qquad \| \, M_i(x) - V_i(x) \, \| \; \leq \; \mathrm{Tr} \left(\overline{\Lambda}_i(x) \right)$$

in the case that the uncertainty is a unit box. This can be proven by a transfer of the argumentation in Theorem 3.1.

3.3 Necessary and Sufficient Optimality Conditions

In this section we are interested in both necessary and sufficient optimality conditions for min-max optimization problems of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^{n_x}} \quad & \max_{w_0\in\mathcal{B}} H_0(x,w_0) \\ \text{subject to} \quad & \max_{w_i\in\mathcal{B}} H_i(x,w_i) \;\leq\; 0 \quad \text{for all } i\in\{1,\ldots,n\}\,. \end{aligned} \tag{3.3.1}$$

This problem has in principle the same form as the generalized robust counterpart problem (3.1.1), but we have switched our notation in order to make clear that we will from now on work with the following assumption:

Assumption 3.2 (Lower Level Convexity): *We assume that the functions $H_0,\ldots,H_n$ are not only twice continuously differentiable but also (for all $x\in\mathbb{R}^{n_x}$) concave in w. Moreover, we assume that the set $\mathcal{B}$ is a convex set of the form*

$$\mathcal{B} \;:=\; \{\, w\in\mathbb{R}^{n_w} \;\mid\; B(w) \;\leq\; 0\} \;,$$

where the function $B:\mathbb{R}^{n_w}\to\mathbb{R}^{n_B}$ is twice continuously differentiable and componentwise convex in w.

In other words, we assume lower level convexity. Recall from the last section that such a convex set $\mathcal{B}$ can be obtained by taking the convex hull of the original uncertainty set W while the function $H_0,\ldots,H_n$ are concave over-estimators of the original functions $F_0,\ldots,F_n$. In this context, we might have the examples

$$B_{\text{ball}}(w) \;=\; \|w\|_2^2 - 1 \qquad \text{and} \qquad B_{\text{box}}(w) \;=\; \begin{pmatrix} w-\overline{w} \\ \underline{w}-w \end{pmatrix} \tag{3.3.2}$$

from the previous section in mind.

Definition 3.3: *A point (x^*,w^*) is said to be a local min-max point if the components of the variable $w^* := (w_0^*,\ldots,w_n^*)$ are global maximizers of the given concave functions $H_0(x^*,\cdot),\ldots,H_n(x^*,\cdot)$ subject to $w_0,\ldots,w_n\in\mathcal{B}$ while x^* is a local minimizer of problem* (3.3.1).

First Order Necessary Conditions

Let us start with a trivial but important observation: if we consider an unconstrained min-max problem of the form

$$\min_x \max_w H(x,w)$$

with a function $H(x,\cdot)$ that is assumed to be strictly concave in w for all x, then a local min-max point (x^*,w^*) satisfies necessarily the stationarity conditions

$$\frac{\partial}{\partial x} H(x^*,w^*) \;=\; 0 \quad \text{as well as} \quad \frac{\partial}{\partial w} H(x^*,w^*) \;=\; 0\,. \tag{3.3.3}$$

Here, the second equation, i.e., the stationarity with respect to w, is easy to verify. In order to prove also the other equation we denote with $w^*(x) := \operatorname{argmax}_w H(x,w)$ the parameterized solution of the lower level maximization problem which is differentiable in x, as we assume strict concavity (implicit function theorem). Consequently, the stationarity condition for the upper level problem can be written as

$$\begin{aligned} 0 \;&=\; \frac{\mathrm{d}}{\mathrm{d}x} H(x,w^*(x)) \\ &=\; \frac{\partial}{\partial x} H(x^*,w^*) + \underbrace{\frac{\partial}{\partial w} H(x^*,w^*)}_{=0} \frac{\partial}{\partial x} w^*(x) \;=\; \frac{\partial}{\partial x} H(x^*,w^*)\,. \end{aligned}$$

Note that the first order necessary optimality conditions (3.3.3) for the unconstrained case would read exactly the same if we would regard min-min problems. In standard unconstrained optimization problems we are searching for stationary points which are minima (or maxima) while for min-max problems we are searching for stationary points which are saddle points. Thus, the first order necessary optimality conditions for a min-max problem are exactly the same as if we had a min-min problem. We can only see a difference if we formulate second order sufficient conditions. Thus, at least in the unconstrained case, we can solve a min-max problem locally in the same way as we would solve a standard minimization problems: we can apply Newton-type methods in order to solve the necessary stationarity equations (3.3.3) numerically.

In Chapter 2 we have discussed strategies to re-formulate a min-max or robust counterpart problem as a standard minimization problem. For example, the robust counterpart of a linear program (LP) with affine polytopic uncertainty can be reformulated as a standard LP, or an LP whose uncertain coefficients are bounded within an ellipsoid, can be formulated

as an SOCP. This type of reformulation strategies were based on the idea to replace the lower level maximization problem by its dual assuming that the dual optimization problem can be worked out explicitly. The reformulation has the advantage that existing optimization algorithms can directly be applied. However, we might also ask whether we can solve a min-max problem directly developing a tailored min-max optimization algorithm, which does not require an explicit reformulation of the problem. The aim of the following consideration is to discuss first order necessary optimality conditions for constrained min-max problem, which will later be exploited by an algorithm.

Note that Assumption 3.2 enables us to equivalently replace the condition "$w \in \mathcal{B}$ maximizes $H(x^*, w)$" (with x^* being a local minimizer of (3.3.1)) by the first order KKT conditions of the form

$$\begin{aligned} 0 &= \frac{\partial}{\partial w} L_j(x^*, w_j^*, \lambda_j^*) \\ 0 &\geq B(w_j^*) \\ 0 &\leq \lambda_j^* \\ 0 &= \sum_{k=0}^{n} \lambda_k^T B_k(w_k^*) \end{aligned} \tag{3.3.4}$$

for all $j \in \{0, \ldots, n\}$. Here, we have used the notation

$$L_j(x, w, \lambda) := H_j(x, w) - \lambda^T B(w)$$

to denote the Lagrangian $L_j : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \times \mathbb{R}^{n_B} \to \mathbb{R}$ which is associated with the j-th lower level concave maximization problem. Note that this argumentation is still applicable, if we only know that every local maximizer must be global while the function H_i is not necessarily concave in w (cf. Example 3.1).

In this context, we make the assumption that at least the Mangasarian-Fromovitz constraint qualification (MFCQ) for the lower level maximization problems holds such that the existence of the multipliers λ_j^* can be guaranteed. In this case, the KKT conditions (3.3.4) are both necessary and sufficient to guarantee that w^* denotes the maximizers of the concave lower level problems. Under the stronger linear independence constraint qualification (LICQ) λ^* is also unique. Following the classical framework [222, 223], we introduce two other assumptions on the maximizers w_j^* of the lower level problems: first we assume that the strict complementarity condition (SCC) is satisfied, i.e., we assume

$(i \in \{0, \ldots, n\})$

$$B(w_i^*) - \lambda_i^* \; < \; 0 \tag{3.3.5}$$

at the local min-max point (x^*, w^*) of our interest. And second, we assume that the second order sufficient condition (SOSC)

$$\forall p_i \in \mathcal{T}_i \setminus \{0\} : \quad p_i^T \left(\frac{\partial^2}{\partial w_i^2} L_i(x^*, w_i^*, \lambda_i^*) \right) p_i < 0 \tag{3.3.6}$$

is satisfied, where the set $\mathcal{T}_i$ is defined as

$$\mathcal{T}_i \;\; := \;\; \left\{ p \in R^{n_w} \;\mid\; \frac{\partial}{\partial w} B^{i,\mathsf{act}}(w^*) p \; = \; 0 \right\}. \tag{3.3.7}$$

Here, $B^{i,\mathsf{act}}$ denotes the active constraint components of the function B in the i-th lower level maximization problem.

Now, we use the language from the semi-infinite programming literature [223]:

Definition 3.4: *A point w^* is nondegenerate if it satisfies the linear independence constraint qualification (LICQ), the strict complementarity condition (SCC), as well as the second order sufficient optimality condition (SOSC) for all lower level maximization problems in* (3.3.1).

The corresponding assumption that a point w^* is nondegenerate is in the context of semi-infinite programming also known under the name reduction ansatz [223, 118]. It can be used to guarantee that the primal and dual solution $\hat{w}_j(x)$ and $\hat{\lambda}_j(x)$ of the j-th parameterized lower level problems of the form

$$\min_{w_j \in \mathcal{B}} H_j(x, w_j) \tag{3.3.8}$$

can be regarded as differentiable functions in x. In fact, if $w_j^* = \hat{w}_j(x^*)$ is a non-degenerate maximizer, the functions $\hat{w}_j$ and $\hat{\lambda}_j$ exist in an open neighborhood $D_x \subset R^{n_x}$ of x^* and are differentiable in this neighborhood D_x . This is a well-known result [199, 200] which follows immediately from the implicit function theorem.

In the next step, we need to take care of the upper level minimization problem. In order to analyze KKT-points, we introduce the following definitions:

Definition 3.5: *We say that a point (x, w) satisfies the upper level (or extended) Mangasarian-Fromovitz constraint qualification (EMFCQ) if there exists a vector $\xi \in \mathbb{R}^{n_x}$ with*

$$\frac{\partial}{\partial x} H_k(x, w)\, \xi \;<\; 0 \qquad \text{for all } \; k \in \mathcal{A}\,. \tag{3.3.9}$$

Here, $\mathcal{A} := \{\, k \mid H_k(x, w) \,=\, 0 \,\}$ denotes the active set of the higher level minimization problem. Moreover, we say that (x, w) satisfies the upper level (or extended) linear independence constraint qualification (ELICQ) if the vectors

$$\frac{\partial}{\partial x} H_i(x, w) \qquad \text{with } \; i \in \mathcal{A} \tag{3.3.10}$$

are linearly independent from each other.

The result of the following theorem has been proven in [135] (without even using the above notation of nondegenerate maximizers) in a more general form, where first order optimality conditions for generalized semi-infinite programming problems are discussed. Concerning optimality conditions in semi-infinite programming we also refer to the earlier work in [118, 119]. For our numerical purposes, we summarize these existing results in a much less general form. This enables us to provide a quite concise proof of the following result:

Theorem 3.2 (First Order Optimality Conditions): *Let (x^*, w^*, λ^*) be a local min-max solution of the problem* (3.3.1) *with w^* being a nondegenerate maximizer of the lower level concave maximization problems at x^* and λ^* the associated dual solution. Now, the following statements hold:*

1. *If (x^*, w^*, λ^*) satisfies the upper level MFCQ condition, then there exists a multiplier $\chi^* \in \mathbb{R}^n$ such that the first order KKT-type conditions*

$$\begin{array}{rclcrcl} 0 &=& \frac{\partial}{\partial x} K(x^*, w^*, \chi^*) & & 0 &=& \frac{\partial}{\partial w} L_j(x^*, w_j^*, \lambda_j^*) \\ 0 &\geq& H_i(x^*, w_i^*) & & 0 &\geq& B(w_j^*) \\ 0 &\geq& \chi_i^* & & 0 &\leq& \lambda_j^* \\ 0 &=& \sum_{k=1}^n \chi_k^* H_k(x^*, w_k^*) & & 0 &=& \sum_{k=0}^n \lambda_k^{*T} B(w_k^*) \end{array} \tag{3.3.11}$$

   are satisfied for all $i \in \{1, \ldots, n\}$ and all $j \in \{0, \ldots, n\}$. Here, we use the notation

$$K(x, w, \chi) \;:=\; H_0(x, w_0) - \sum_{k=1}^{n} \chi_k H_k(x, w_k)$$

in order to define the upper level Lagrange function.

2. *If* (x^*, w^*, λ^*) *satisfies also the upper level LICQ condition, then the associated multiplier* χ^* *in the necessary conditions* (3.3.11) *is unique.*

Proof: Due to the complementarity relation for the lower level maximization problems we have

$$\forall x \in D_x : \quad H_j(x, \hat{w}_j(x)) \;=\; L_j(x, \hat{w}_j(x), \hat{\lambda}_j(x)) \;,$$

where $\hat{w}_j$ and $\hat{\lambda}_j$ denote the parameterized dual solution of the lower level maximization problems as a function in $x \in D_x$ as introduced above. Thus, the min-max problem (3.3.1) is locally equivalent to the following auxiliary problem

$$\begin{array}{ll} \min\limits_{x \in D_x} & L_0(x, \hat{w}_0(x), \hat{\lambda}_0(x)) \\ \text{s.t.} & L_i(x, \hat{w}_i(x), \hat{\lambda}_i(x)) \;\leq\; 0 \quad \text{for all} \quad i \in \{1, \ldots, n\} \;. \end{array} \tag{3.3.12}$$

Using the optimality and feasibility condition for the lower level maximizer $\hat{w}_j(x^*)$, we find

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x} L_j(x^*, \hat{w}_j(x^*), \hat{\lambda}_j(x^*)) \;&=\; \frac{\partial}{\partial x} L_j(x^*, w_j^*, \lambda_j^*) \\ &\quad + \frac{\partial}{\partial w} L_j(x^*, w_j^*, \lambda_j^*) \frac{\partial \hat{w}_j(x^*)}{\partial x} - B^{j,\mathsf{act}}(w_j^*) \frac{\partial \hat{\lambda}_j(x^*)}{\partial x} \\ &=\; \frac{\partial}{\partial x} H_j(x^*, w_j^*) \;. \end{aligned}$$

for all $j \in \{0, \ldots, n\}$. In the last step we have used the stationarity conditions

$$\frac{\partial}{\partial w} L_j(x^*, w_j^*, \lambda_j^*) \;=\; 0 \quad \text{as well as the relation} \quad B^{j,\mathsf{act}}(w_j^*) \;=\; 0$$

which holds by the definition for the active constraints. Thus, the upper level MFCQ (or upper level LICQ) condition from Definition 3.5 boils down to the MFCQ (or LICQ) condition for the auxiliary problem (3.3.12). The statements of the theorem are now equivalent to the standard KKT theorem for problem (3.3.12) under the MFCQ and LICQ condition, respectively. □

Remark 3.3: *The above proof can be generalized for the case that the lower level problems comprise not only convex inequalities but also linear equalities. Furthermore, we could consider the case that problem* (3.3.1) *has additional equality and/or inequality constraints which only depend on* x *etc. Please note that such generalizations are straightforward and omitted here for the ease of notation.*

In the above form, the first order necessary optimality conditions for min-max problems should be easy to remember: we first write down the first order KKT conditions for the maxima by fixing x^*, which yields the right-hand column in conditions (3.3.11), and second we write down the first order KKT conditions for a local minimum of the higher level problem neglecting the implicit dependence of the maximizer w^* on x. Note that the multiplier χ^* of the upper level problem satisfies $0 \geq \chi_i^*$ while the multipliers of the lower level maximization problems satisfy $0 \leq \lambda_j^*$. In this sense, the only difference between standard minimization problems and min-max optimization problems are a couple of "minus-signs" in the optimality conditions.

Finally, we conclude our discussion of first order necessary optimality conditions with the following definition:

Definition 3.6: *We say that a point (x^*, w^*, λ^*) is a KKT point of the min-max problem* (3.3.1) *if and only if (w_j^*, λ_j^*) is a nondegenerate maximizer of the j-th lower level maximization problem for all $j \in \{0, \ldots, m\}$ and x^* is a KKT point of the auxiliary problem* (3.3.12).

Second Order Sufficient Conditions

Let us discuss second order sufficient conditions for a local solution of the min-max problem (3.3.1) reviewing the results in [119] on semi-infinite programming problems. Note that the following result on second order sufficient conditions is not a new result but summarized in a form in which it will later be needed for the discussion of numerical algorithms.

Let us come back to the implicitly defined functions $\hat{w}_j(x)$ and $\hat{\lambda}_j(x)$ which have been used within the proof of Theorem 3.2. We denote the active set of the j-th lower level problem in a KKT point (x^*, w^*, λ^*) by

$$\mathcal{A}_j^* \;:=\; \Big\{ k \mid B_k(w_j^*) = 0 \Big\} \;=\; \{ k \mid \lambda_k^* > 0 \} \;.$$

Note that we assume here that the maximizers are non-degenerate in the sense of Definition 3.4. Hence, we have in particular strict complementarity as indicated in the

above equation. Moreover, we define the matrices:

$$\Omega_j^* \;:=\; \begin{pmatrix} \frac{\partial^2}{\partial w^2} L_j(x^*, w_j^*, \lambda_j^*) & \left(\frac{\partial}{\partial w} B^{j,\mathsf{act}}(w_j^*)\right)^T \\ \frac{\partial}{\partial w} B^{j,\mathsf{act}}(w_j^*) & \end{pmatrix}$$

$$\text{and } R_j^* \;:=\; \begin{pmatrix} \frac{\partial^2}{\partial w \partial x} H_j(x^*, w_j^*, \lambda_j^*) \\ 0_{|\mathcal{A}_j^*|} \end{pmatrix} .$$

Here, $B^{j,\mathsf{act}}$ is a function which consists of the components B_k of the function B for which $k \in \mathcal{A}_j$. The syntax $0_{|\mathcal{A}_j^*|}$ denotes a $|\mathcal{A}_j^*|$ dimensional vector filled with zeros.

Proposition 3.1: *Let (x^*, w^*, λ^*) be a KKT point of the min-max problem* (3.3.1) *in the sense of Definition 3.6 with w^* being a nondegenerate maximizer. Now, we have for all $j \in \{0, \ldots, n\}$:*

$$\frac{\partial}{\partial x} \begin{pmatrix} \hat{w}_j(x^*) \\ -\hat{\lambda}_j^{\mathsf{act}}(x^*) \end{pmatrix} \;=\; -\Omega_j^{*-1} R_j \; .$$

Here, $\hat{\lambda}_j^{\mathsf{act}}$ denotes the components of the function $\hat{\lambda}_j$, whose index is in the active set $\mathcal{A}_j^*$.

Proof: Note that under the non-degeneracy assumption for the lower level problems, the matrix Ω_j^* is invertible and the active set remains constant in a neighborhood of (x^*, w^*, λ^*) - thus, we can simply compute the derivative by solving the associated parameterized KKT system with respect to the active constraints. □

Theorem 3.3: *Let the conditions of Theorem 3.2 be satisfied with $(x^*, w^*, \lambda^*, \chi^*)$ being the KKT point of the min-max problem* (3.3.1) *satisfying the conditions* (3.3.11). *Furthermore, we introduce for each $j \in \{0, \ldots, n\}$ a Schur matrix $S_j^* \in \mathbb{R}^{n_x \times n_x}$ defined by*

$$S_j^* \;:=\; \frac{\partial^2}{\partial x^2} H_j(x^*, w_j^*) - R_j^{*T} \Omega_j^{*-1} R_j^*$$

as well as a tangent space $\mathcal{T} \subseteq \mathbb{R}^{n_x}$ defined by

$$\mathcal{T} \;:=\; \left\{ \xi \in \mathbb{R}^{n_x} \;\middle|\; \forall k \in \mathcal{A} : \; \frac{\partial}{\partial x} H_k(x^*, w^*)\, \xi \;=\; 0 \right\} .$$

If the second order sufficiency condition

$$\forall\, \xi \in \mathcal{T}: \quad \xi^T \left(S_0^* - \sum_{k=1}^{n} \chi_k^* S_j^* \right) \xi \;>\; 0$$

is satisfied, then (x^*, w^*, λ^*) *is a locally optimal of the min-max problem* (3.3.1).

Proof: In the proof of Theorem 3.2 we have introduced the auxiliary problem (3.3.12) which is locally equivalent to the min-max problem (3.3.1). Using Proposition 3.1 we find that

$$\forall\, j \in \{0, \ldots, n\}: \qquad \frac{\mathrm{d}^2}{\mathrm{d}x^2}\, L_j(x^*, \hat{w}_j(x^*), \hat{\lambda}_j(x^*)) \;\;=\;\; S_j^* \,.$$

Consequently, the statement of the above theorem is equivalent to the standard second order sufficient condition (SOSC) for the auxiliary problem (3.3.12). □

3.4 Mathematical Programming with Complementarity Constraints

The key idea of the above proof was to introduce the auxiliary problem (3.3.12) which is equivalent to the original min-max problem (3.3.1). However, in order to formulate problem (3.3.12) we had to introduce the functions $\hat{w}_j$ and $\hat{\lambda}_j$ by using the implicit function theorem. These functions are only locally defined and might become non-differentiable at points x that are far from x^*. We are interested in the question whether we can also transform problem (3.3.1) into an equivalent minimization problem without using the implicit function theorem. This is possible by writing the KKT conditions into the constraints, considering a mathematical program with complementarity constraints (MPCC) of the form

$$\begin{array}{ll} \underset{x,w,\lambda}{\text{minimize}} & H_0(x, w_0) \\ \text{subject to} & \left\{ \begin{array}{lcllcl} 0 & \geq & H_i(x, w_i) & 0 & = & \nabla_w L_j(x, w_j, \lambda_j) \\ 0 & \geq & B(w_j) & 0 & \leq & \lambda_j \\ 0 & = & \sum_{k=0}^{n} \lambda_k^T B_k(w_k) & & & \end{array} \right. \end{array} \tag{3.4.1}$$

for all $i \in \{1, \ldots, n\}$ and all $j \in \{0, \ldots, n\}$. Note that MPCC formulations of the above form are well-known and discussed in the literature [223]. However, mathematical problems

with complementarity have certain disadvantages in our context. Before we discuss the details of this statement, we outline the two main drawbacks of formulation (3.4.1) as follows:

- It is without further precaution not trivial to discuss KKT points of an MPCC. For example the Mangasarian Fromovitz constraint qualification (MFCQ) for the minimization problem (3.4.1) is violated at all feasible points of an MPCC. This can directly be seen by looking at the complementarity conditions but we also refer to [205] for a discussion of the details of this statement. As the LICQ condition implies the MFCQ condition, both constraint qualifications do not help in the context of mathematical programs with complementarity constraints.
- If the functions H_j are convex in x and concave in w for all $j \in \{0, \ldots, n\}$, the original robust counterpart problem (3.3.1) is perfectly convex. However, formulation (3.4.1) must in this form be regarded as a non-convex optimization problem.

Due to the above observations, the MPCC (3.4.1) must have a certain structure which we have to exploit in order to re-cover a non-degenerate formulation.

Remark 3.4: *The degeneracy of the MPCC* (3.4.1) *seems to be the main motivation for the development of smoothing techniques for numerical approaches. In [223] or also in [88] such smoothing techniques for MPCCs have been discussed. Here, the main concept is to replace the complementarity conditions by an NCP function* $\Psi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *which satisfies by definition*

$$\Psi(a, b) = 0 \quad \textit{if and only if} \quad a \geq 0,\ b \geq 0,\ ab = 0\ . \tag{3.4.2}$$

Note that for example the (smoothed) Fischer-Burmeister function [92], which is a function of the form $\Psi_\tau(a, b) := a + b - \sqrt{a^2 + b^2 + 2\tau^2}$*), or also the Chen-Harker-Kanzow-Smale function [56], which is defined as* $\Psi_\tau(a, b) := \frac{1}{2}\left(a + b - \sqrt{(a-b)^2 + 4\tau^2}\right)$*, satisfy for* $\tau = 0$ *the above property. Using such an NCP function* Ψ*, the MPCC* (3.4.1) *can equivalently be written as*

$$\begin{aligned}
\underset{x,w,\lambda}{\textit{minimize}} \quad & H_0(x, w_0) \\
\textit{subject to} \quad & 0 \ \geq\ H_i(x, w_i) \\
& 0 \ =\ \nabla_w L_j(x, w_j, \lambda_j) \\
& 0 \ =\ \Psi(-B_l(w_j), (\lambda_j)_l)
\end{aligned}$$

for all components $l \in \{1, \ldots, n_B\}$ and all $i \in \{1, \ldots, n\}$, $j \in \{0, \ldots, n\}$. As NCP functions are typically non-smooth, the function Ψ must be regularized before the above minimization problem can be solved with standard derivative based NLP solvers - for example by using $\tau > 0$ in the above examples for smoothed NCP functions. This leads to a kind of interior point approach where a sequence of regularized NLPs must be solved and which can be shown to converge to Fritz-John points of the original semi-infinite programming problem (3.3.1)*. For the details of this smoothing approach we refer to [223].*

Note that there exist also general purpose SQP methods for MPCCs (3.4.1)*. For a recommendable article about such SQP methods for mathematical programming problems with complementarity constraints, we refer to the work of Fletcher, Leyffer, Ralph, and Scholtes [98]. Note, that the local convergence properties of such methods are typically challenging to analyze, as the KKT points of the MPCC* (3.4.1) *do not satisfy the MFCQ condition. Additionally, globalization results for general purpose SQP methods applied to MPCCs are - due to the unbounded multiplier solution set of an MPCC - difficult to obtain [98], but they are subject of current research [8, 239].*

Remark 3.5: *Note that there exists extensive literature on optimality conditions and constraint qualifications for general MPCCs. Here, we refer to the work of Flegel and Kanzow [94] and the references therin. Although, the standard constraint qualifications like MFCQ and LICQ cannot be applied for MPCC's, Flegel and Kanzow have shown that Guignard's constraint qualification [113] can be used to discuss KKT points.*

In the following, we develop an alternative strategy to deal with the MPCC (3.4.1), which does not employ the smoothing techniques which have been outlined in the remark above. Rather, we plan to use the particular structure which occurs in problem (3.4.1) as this MPCC arises within the context of min-max formulations.

Elimination of the Complementarity Constraints

How can we avoid the complementarity constraints in the MPCC (3.4.1)? The first idea would be to simply skip the complementarity constraint in the formulation. As the primal and dual feasibility constraints of the lower level maximization problems are still required, we could then still guarantee that the term $\lambda_i^T B(x, w_i)$ is non-positive in the optimal solution, but not necessarily zero. In order to fix this, we plan to penalize the terms

$-\lambda_i^T B(x, w_i)$. Thus, we add them in the constraints and objective of our formulation:

$$\begin{array}{ll} \underset{x,w,\lambda}{\text{minimize}} & H_0(x, w_0) - \lambda_0^T B(w_0) \\ \text{subject to} & \left\{ \begin{array}{lcllcl} 0 & \geq & H_i(x, w_i) - \lambda_i^T B(w_i) & 0 & = & \nabla_w L_j(x, w_j, \lambda_j) \\ 0 & \geq & B(w_j) & 0 & \leq & \lambda_j \end{array} \right. \end{array} \tag{3.4.3}$$

with $i \in \{1, \ldots, n\}$ and all $j \in \{0, \ldots, n\}$. This way of reformulating the MPCC turns out to be an equivalence transformation:

Lemma 3.5: *Problem* (3.4.3) *is equivalent to original MPCC* (3.4.1) *in the sense that every solution* x^* *of problem* (3.4.3) *corresponds to a solution of the original problem* (3.4.1).

Proof: Note that problem (3.4.3) can also in a more compact form be written as

$$\begin{array}{ll} \underset{x,w,\lambda}{\text{minimize}} & L_0(x, w_0, \lambda_0) \\ \text{subject to} & \left\{ \begin{array}{lcllcl} 0 & \geq & L_i(x, w_i, \lambda_i) & 0 & = & \nabla_w L_j(x, w_j, \lambda_j) \\ 0 & \geq & B(w_j) & 0 & \leq & \lambda_j \ . \end{array} \right. \end{array} \tag{3.4.4}$$

Due to duality, the inequalities $0 \geq B(w_j)$ and $0 \leq \lambda_j$ imply that we must have

$$H_i(x, w_i) \ \leq \ L_i(x, w_i, \lambda_i) \ \leq \ 0 \, .$$

Thus, the upper level constraints of the original minimization problem are satisfied in the optimal solution. This implies that every solution x^* of problem (3.4.3) corresponds to a solution of the original problem (3.4.1), as we can achieve $\lambda_j^T B(w_j) = 0$ for all $j \in \{0, \ldots, n\}$. □

A Closer Look at the Optimality Conditions

In the following, we plan to analyze the optimality conditions of problem (3.4.4) in more detail which are expected to coincide with the conditions from Theorem 3.2. For this aim, we first define the Lagrangian function for this minimization problem as

$$\begin{aligned} \mathcal{K} \ := \ & L_0(x, w_0, \lambda_0) - \sum_{i=1}^{n} \chi_i L_i(x, w_i, \lambda_i) \\ & - \sum_{j=0}^{n} \left[\rho_j^T \nabla_w L_j(x, w_j, \lambda_j) + \mu_j^T B(w_j) - \kappa_j^T \lambda_j \right] \, . \end{aligned} \tag{3.4.5}$$

Here, the variables $\chi \in \mathbb{R}^n$, $\rho_j \in \mathbb{R}^{n_w}$, $\mu_j \in \mathbb{R}^{n_B}$, and $\kappa \in \mathbb{R}^{n_B}$ with $j \in \{0, \ldots, n\}$ are multipliers which are associated with the constraints in the minimization problem (3.4.4).

In the following, we will show that the multipliers ρ_j and μ_j vanish in the optimal solution of the problem (3.4.4). This is a technical result which will in the next chapter be used for constructing efficient algorithms.

Lemma 3.6: *Let (x^*, w^*, λ^*) be a KKT point of problem (3.4.4) such that w^* is a nondegenerate maximizer of the lower level problem. Then the multipliers ρ_j which are associated with the stationarity constraint*

$$\nabla_w L_j(x, w_j, \lambda_j) \; = \; 0$$

are all equal to zero, i.e., we have $\rho_j = 0$ for all $j \in \{0, \ldots, n\}$. Similarly, the multipliers which are assoicated with the constraints of the form $B(w_j) \leq 0$ vanish, too. In other words, we have $\mu_j = 0$ for all $j \in \{0, \ldots, n\}$.

Proof: The first step of the proof is to work out the stationarity conditions for problem (3.4.4) which must hold at any solution point (x^*, w^*, λ^*). Differentiating the Lagrangian function $\mathcal{K}$, which has been defined in equation (3.4.5), with respect to w and λ yields the relations

$$\begin{aligned} \forall j \in \{0, \ldots, n\}: \quad 0 \;&=\; \rho_j^T \frac{\partial^2 L_j(x^*, w_j^*, \lambda_j^*)}{\partial w^2} + \mu_j^T \frac{\partial B(w_j^*)}{\partial w} \qquad (3.4.6) \\ \text{and} \quad 0 \;&=\; \chi_j B(w_j^*) + \frac{\partial B(w_j^*)}{\partial w} \rho_j + \kappa_j \; , \end{aligned}$$

respectively. Here, we have already simplified the stationarity condition with respect to w by using the relation $\frac{\partial L_j(x^*, w_j^*, \lambda_j^*)}{\partial w} = 0$. Moreover, we use the definition $\chi_0 := -1$ in order to simplify notation. Now, the main idea is to multiply equation (3.4.6) from the right with ρ_j. This leads to

$$\forall j \in \{0, \ldots, n\}: \quad 0 \;=\; \rho_j^T \frac{\partial^2 L_j(x^*, w_j^*, \lambda_j^*)}{\partial w^2} \rho_j - \chi_j \mu_j^T B(w_j^*) - \mu_j^T \kappa_j$$

As we can use the complementarity condition $\mu_j^T B(w_j^*) = 0$, this equation further simplifies to

$$\forall j \in \{0, \ldots, n\}: \quad \rho_j^T \frac{\partial^2 L_j(x^*, w_j^*, \lambda_j^*)}{\partial w^2} \rho_j \;=\; \mu_j^T \kappa_j \;\geq\; 0 \; . \qquad (3.4.7)$$

If we would assume that the matrix $\frac{\partial^2 L_j(x^*,w_j^*,\lambda_j^*)}{\partial w^2} \prec 0$ is negative definite, we could now already conclude that we must have $\rho_j = 0$. However, as we assume that w^* is a nondegenerate maximizer of the lower level problem, we only know that the Hessian matrix $\frac{\partial^2 L_j(x^*,w_j^*,\lambda_j^*)}{\partial w^2}$ is negative definite in the subspace which is spanned by the active lower level constraints. Thus, we introduce the short hand $\frac{\partial B^{\mathsf{act}}(w_j^*)}{\partial w}$ to denote those rows of the matrix $\frac{\partial B(w_j^*)}{\partial w}$ whose index corresponds to an active lower level constraint in the j-th lower level maximization problem. Here, we also write $B^{\mathsf{act}}(w_j^*) = 0$. Similarly, κ_j^{act} collects the components of the vector κ which are in this active set. As we assume that w^* is a nondegenerate lower level maximizer, we can use the strict complementarity condition to show that we have $\lambda_j^{\mathsf{act}} > 0$. Note that the complementarity condition for problem (3.4.4) can be written as $(\kappa^{\mathsf{act}})^T \lambda_j^{\mathsf{act}} = 0$ which implies $\kappa^{\mathsf{act}} = 0$. Thus, we conclude

$$\begin{aligned} \forall j \in \{0,\ldots,n\}: \quad 0 \;&\leq\; \rho_j^T \frac{\partial^2 L_j(x^*,w_j^*,\lambda_j^*)}{\partial w^2} \rho_j \;\geq\; 0 \\ \text{and} \quad 0 \;&=\; \chi_j B^{\mathsf{act}}(w_j^*) + \frac{\partial B^{\mathsf{act}}(w_j^*)}{\partial w}\rho_j + \kappa_j^{\mathsf{act}} \;=\; \frac{\partial B^{\mathsf{act}}(w_j^*)}{\partial w}\rho_j \;=\; 0\,. \end{aligned}$$

Indeed, as $\frac{\partial^2 L_j(x^*,w_j^*,\lambda_j^*)}{\partial w^2}$ is negative definite in the tangential subspace which is spanned by the active lower level constraints, we may conclude $\rho_j = 0$ for all indices $j \in \{0,\ldots,n\}$. Finally, as we can use $\rho_j = 0$ in equation (3.4.6), we may also conclude $\mu_j = 0$ for all $j \in \{0,\ldots,n\}$, as we can use that the lower level LICQ condition is satisfied. □

Chapter 4

Sequential Algorithms for Robust Optimization

The aim of this chapter is to develop numerical algorithms which can solve min-max problems of the form (3.3.1). Recall that this problem can be written as

$$\begin{array}{lll} \min\limits_{x\in\mathbb{R}^{n_x}} & \max\limits_{w_0\in\mathcal{B}} & H_0(x,w_0) \\ \text{s.t.} & \max\limits_{w_i\in\mathcal{B}} & H_i(x,w_i) \;\leq\; 0 \qquad \text{with } \; i\in\{1,\ldots,n\}\;, \end{array} \tag{4.0.1}$$

where the functions H_i are assumed to be concave in w. Moreover, the uncertainty set $\mathcal{B}$ is – as in the previous chapter – assumed to be given in the explicit form

$$\mathcal{B} \;:=\; \{\, w\in\mathbb{R}^{n_w} \mid B(w) \;\leq\; 0 \,\}\;,$$

where the function B is component-wise convex in w. In this context, our aim is not to find global solutions of the above min-max problem, but we are looking for an algorithm which converges globally to local minimizers.

The question of how such an algorithm should be designed depends heavily on the functions H_i and B. For example the dimensions of these functions, the dimensions n_x and n_w of the optimization variables x and w, respectively, the costs for a function evaluation, as well as the cost of computing derivatives of the functions H_i and B will mainly influence our choice of numerical techniques. If the function evaluation is cheap while the difficulty is in determining the active sets, an application of interior point techniques might come to our mind. However, in this thesis, we are interested in the opposite situation, i.e., in

the case that the evaluation of the functions and their derivatives is the most expensive part. For standard nonlinear programs, SQP methods have turned out to perform very well in such situations [182].

We first discuss in Section 4.1 the advantages and disadvantages of tailored sequential quadratic programming algorithms, which are one option to solve the min-max problem (4.0.1). In Section 4.2 an alternative strategy is suggested which we call sequential convex bilevel programming. The local and global convergence properties of this method are discussed in Sections 4.3 and 4.4, respectively.

4.1 Tailored Sequential Quadratic Programming Methods

The aim of this section is to discuss a tailored sequential quadratic programming algorithm applied to a structured optimization problem of the form

$$\begin{array}{ll} \underset{x,w,\lambda}{\text{minimize}} & L_0(x, w_0, \lambda_0) \\ \text{subject to} & \left\{ \begin{array}{lcllcl} 0 & \geq & L_i(x, w_i, \lambda_i) & 0 & = & \nabla_w L_j(x, w_j, \lambda_j) \\ 0 & \geq & B(w_j) & 0 & \leq & \lambda_j \end{array} \right. \end{array} \tag{4.1.1}$$

with $i \in \{1, \ldots, n\}$ and $j \in \{0, \ldots, n\}$. Here, we adopt the notation from the previous chapter planning to solve the original min-max problem (4.0.1) numerically. Recall that the Lagrangian functions L_j can explicitly be written as

$$L_j(x, w_j, \lambda_j) \;=\; H_j(x, w_j) - \lambda_j^T B(w_j)\,.$$

As it has extensively been discussed within Section 3.4, problem (4.1.1) is equivalent to the original min-max problem (4.0.1).

As the optimization problem (4.1.1) is a standard minimization problem, existing optimization algorithms can be applied. In this section, our focus is on sequential quadratic programming (SQP) methods. In order to transfer the main idea of such SQP methods to our situation, we assume that we have an initial guess z^0 for a local minimizer $z^* := (x^*, w^*, \lambda^*)$ of problem (4.1.1). Now, we plan to perform iterates of the form

$$z^+ \;=\; z + \alpha\,\Delta z := \begin{pmatrix} x + \alpha \Delta x \\ w + \alpha \Delta w \\ \lambda + \alpha\,\Delta\lambda \end{pmatrix}$$

with $\alpha \in (0,1]$ being a damping parameter. The aim is to choose the steps Δz such that the corresponding sequence of iterates z converges to a local solution z^* of problem (4.1.1). Here, it should be explained that we suppress the iteration index within our notation, i.e., z^+ denotes always the "next" iterate in the sequence.

Applying the standard version of SQP, the step Δz is obtained by solving a quadratic programming problem of the form

$$\begin{aligned} \min_{\Delta z} \quad & L^0 + L^0_x \Delta x + L^0_w \Delta w_0 - B_0^T \Delta\lambda_0 + \tfrac{1}{2}\Delta z^T K \Delta z \\ \text{s.t.} \quad & 0 \;\geq\; L^i + L^i_x \Delta x + L^i_w \Delta w_i - B_i^T \Delta\lambda_i \\ & 0 \;=\; L^j_w + \Delta x^T L^j_{xw} + \Delta w_j^T L^j_{ww} - \Delta\lambda_j^T B^j_w \\ & 0 \;\geq\; B^j + B^j_w \Delta w \\ & 0 \;\leq\; \lambda_j + \Delta\lambda_j \;, \end{aligned} \tag{4.1.2}$$

where the constraints are imposed for all indices $i \in \{1,\dots,n\}$ and all $j \in \{0,\dots,n\}$. Here, we use the following short hands:

$$\begin{aligned} L^j_{ww} &:= \tfrac{\partial^2}{\partial w^2} L_j(x, w_j, \lambda_j)\,, & L^j_{wx} &:= \tfrac{\partial^2}{\partial w \partial x} L_j(x, w_j, \lambda_j)\,, & L^j_{xw} &:= \left(L^j_{wx}\right)^T\,, \\ L^j_w &:= \tfrac{\partial}{\partial w} L_j(x, w_j, \lambda_j)\,, & L^j_x &:= \tfrac{\partial}{\partial x} L_j(x, w_j, \lambda_j)\,, & L^j &:= L_j(x, w_j, \lambda_j)\,, \\ B^j_w &:= \tfrac{\partial}{\partial w} B(w_j)\,, & & & B^j &:= B(w_j)\,. \end{aligned}$$

Most of the commonly used variants of SQP methods distinguish only in the way how the Hessian matrix is approximated. In our case, the Hessian approximation is denoted by the symmetric matrix $K \in \mathbb{S}^{n_x + (n+1)\,n_w + (n+1)\,n_B}$. Recall that the Lagrangian function $\mathcal{K}$, which is associated with the optimization problem (4.1.1), has already been worked out within equation (3.4.5). The exact Hessian matrix $\frac{\partial^2 \mathcal{K}}{\partial z^2}$ has a particular structure, which we plan to exploit for the construction of an approximation of the form

$$K \;\approx\; \frac{\partial^2 \mathcal{K}}{\partial z^2} \;=\; \left(\begin{array}{c|ccc|ccc} \mathcal{K}_{xx} & \mathcal{K}^0_{xw} & \dots & \mathcal{K}^n_{xw} & 0 & \dots & 0 \\ \hline \mathcal{K}^0_{wx} & \mathcal{K}^0_{ww} & & & \mathcal{K}^0_{w\lambda} & & \\ \vdots & & \ddots & & & \ddots & \\ \mathcal{K}^n_{wx} & & & \mathcal{K}^n_{ww} & & & \mathcal{K}^n_{w\lambda} \\ \hline 0 & \mathcal{K}^0_{\lambda w} & & & & & \\ \vdots & & \ddots & & & 0 & \\ 0 & & & \mathcal{K}^n_{\lambda w} & & & \end{array}\right)\,. \tag{4.1.3}$$

In this context, we are using the following short hands (with $\chi_0 := -1$):

$$\begin{aligned}
\mathcal{K}_{xx} &:= \sum_{j=0}^{n} \mathcal{K}_{xx}^j \,, \\
\mathcal{K}_{xx}^j &:= -\chi_j L_{xx}^j - \rho_j^T L_{wxx}^j \,, \\
\mathcal{K}_{wx}^j &:= \left(\mathcal{K}_{xw}^j\right)^T = -\chi_j L_{wx}^j - \rho_j^T L_{wwx}^j \,, \\
\mathcal{K}_{ww}^j &:= -\chi_j L_{ww}^j - \rho_j^T L_{www}^j - \mu^T B_{ww}^j \,, \\
\mathcal{K}_{w\lambda}^j &:= \left(\mathcal{K}_{w\lambda}^j\right)^T = -\chi_j L_{w\lambda}^j - \rho_j^T L_{ww\lambda}^j \,.
\end{aligned}$$

At this point, there are several options on how we can deal with the particular structure of the Hessian matrix (4.1.3). In the following, we compare four different options discussing in each case the corresponding advantages and disadvantages.

Standard BFGS-SQP Algorithms

First, we discuss the possibility to solve problem (4.1.1) by applying a standard SQP method without exploiting any structure. Most of the existing SQP solvers use BFGS-updates [48, 95, 104, 211] for approximating the Hessian matrix. Here, the Hessian matrix K is in each step updated as

$$K^+ = K + \frac{yy^T}{y^T s} - \frac{Kss^T K}{s^T K s} \,, \tag{4.1.4}$$

where we define $s := \alpha \Delta z$ and $y := \frac{\partial \mathcal{K}(z^+)}{\partial z} - \frac{\partial \mathcal{K}(z)}{\partial z}$. Again, there are several variants, as for example Powell's modification [192] which maintains a symmetric and positive definite approximation matrix K. These BFGS-SQP methods are - under some mild non-degeneracy assumptions [66, 199, 200] - q-superlinearly convergent in a neighborhood of a minimizer z^* (cf. also [182]). This variant would have the advantage that it is easy to realize, as we can simply take an existing implementation, but has the disadvantage that we do not exploit the problem structure efficiently: first, a standard BFGS-SQP method would not exploit the block structure of the Hessian matrix $\frac{\partial^2 \mathcal{K}}{\partial z^2}$ and second, we would not use the fact that some of the blocks in the Hessian matrix – for example the terms L_{ww}^j and L_{wx}^j – have to be computed anyhow.

Block BFGS-SQP Algorithms

The second possibility is to slightly modify the standard BFGS-SQP updates in order to exploit the block structure of the Hessian matrix. For this aim, we first introduce slack variables $x_0, x_1, \ldots, x_n \in \mathbb{R}^{n_x}$ writing the problem (4.1.1) in the equivalent form

$$\begin{array}{ll} \underset{x,w,\lambda}{\text{minimize}} & L_0(x_0, w_0, \lambda_0) \\ \text{subject to} & \left\{ \begin{array}{llll} 0 \;\geq\; L_i(x_i, w_i, \lambda_i) & 0 \;=\; \nabla_w L_j(x_j, w_j, \lambda_j) \\ 0 \;\geq\; B(w_j) & 0 \;\leq\; \lambda_j \\ x_0 \;=\; x_1 \;=\; \ldots \;=\; x_n \;. \end{array} \right. \end{array} \tag{4.1.5}$$

The advantage of this reformulation is that the optimization problem is "almost" decoupled – the only coupling is collected in the linear constraint of the form

$$x_0 \;=\; x_1 \;=\; \ldots \;=\; x_n \;.$$

The optimization variables can in this case be arranged in the following order:

$$\hat{z} \;:=\; \left(\hat{z}_0^T, \ldots, \hat{z}_n^T \right)^T \quad \text{with} \quad \hat{z}_j^T := \left(x_j^T, w_j^T, \lambda_j^T \right)^T \;.$$

The main point is now that the linear constraint does not have a contribution to the Hessian matrix $\frac{\partial^2 \hat{K}}{\partial \hat{z}^2}$ which is associated with the problem (4.1.5) and which can be written as

$$\frac{\partial^2 \hat{K}}{\partial \hat{z}^2} \;=\; \begin{pmatrix} \hat{K}_{zz}^0 & & & 0 \\ & \hat{K}_{zz}^1 & & \\ & & \ddots & \\ 0 & & & \hat{K}_{zz}^n \end{pmatrix} \quad \text{with} \quad \hat{K}_{zz}^j \;:=\; \begin{pmatrix} \mathcal{K}_{xx}^j & \mathcal{K}_{xw}^j & \mathcal{K}_{w\lambda}^j \\ \mathcal{K}_{wx}^j & \mathcal{K}_{ww}^j & 0 \\ \mathcal{K}_{\lambda w}^j & 0 & 0 \end{pmatrix} \;.$$

The idea is now to use the BFGS updates not for approximating the whole Hessian matrix but only to approximate the diagonal sub-blocks $\hat{K}_{zz}^j$ for all $j \in \{0, \ldots, n\}$. Note that sparse or block structured matrix updates for optimization have intensively been studied by Toint [227] and Griewank [111], as well by Bock and Plitt [43]. As for the unstructured BFGS-SQP method, we can typically establish q-superlinear convergence of the method, but the hope is that the block structured high-rank updates lead to a better performance than the unstructured updates, as the high-rank updates take additional information into account. Here, the block structured updates also reduce the memory requirements of the method. Moreover, there exist QP-solvers which exploit sparse Hessian matrices as well. For the details and a complete overview of such sparse QP solvers we refer to a recent work by Maes [165] – and the references therein.

Exact Hessian SQP Algorithms

Let us also discuss the possibility to apply an exact Hessian SQP methods, as for example analyzed by Fletcher [96]. The advantage of computing the Hessian matrix exactly is that we can – under some mild regularity assumptions [96] – expect that the method converges locally q-quadratic. Another advantage is also here that existing implementations can be used, i.e., we do not have to write a specialized algorithm. Another motivation for using exact Hessians is that we have to compute the matrices L^j_{ww}, B^j_{ww} and L^j_{wx} anyhow. The reason for this is that we need these terms in order construct the linearization of the stationarity constraints

$$0 \;=\; L^j_w + \Delta x^T L^j_{xw} + \Delta w_j^T L^j_{ww} - \Delta \lambda_j^T B^j_w \;.$$

However, we should also be aware of the fact that the sub-blocks $\mathcal{K}_{xx}$ and $\mathcal{K}^j_{wx}$ within the exact Hessian matrix require the computation of third order derivatives of the model functions

$$\begin{aligned} L^j_{www} &:= \frac{\partial^3}{\partial w^3} L_j(x, w_j, \lambda_j) \\ L^j_{wwx} &:= \frac{\partial^3}{\partial w^2 \partial x} L_j(x, w_j, \lambda_j) \\ \text{and} \quad L^j_{ww\lambda} &:= \frac{\partial^3}{\partial w^2 \partial \lambda} L_j(x, w_j, \lambda_j) \end{aligned}$$

with $j \in \{0, \ldots, n\}$. Thus, if we apply an exact Hessian SQP method blindly, we need to compute these expensive third order terms. Another disadvantage of exact Hessian SQP is that the matrix K is in general not positive semi-definite. This implies for example that we can be in the situation that we solve a sequence of non-convex QPs, even if the original min-max problem is upper-level convex. Although state-of-the-art SQP solvers can in principle deal with indefinite Hessian matrices, too, the non-convexity is often a source of practical problems, as non-standard solvers for the non-convex QPs have to be used and also globalization techniques become more difficult [60, 182].

Asymptotically Exact Hessian SQP Algorithms

We assume for a moment that we can accept possibly indefinite Hessians in the QP. In this case, there arises the questions, whether we can modify the above exact Hessian SQP

algorithm in such a way that we can still get q-quadratic convergence without computing any third order derivatives. In order to motivate this, note that the formulation of the first order necessary optimality conditions within Theorem 3.2 requires first order derivatives only, which suggest that we can achieve quadratic convergence without computing any third order derivatives of the Lagrangian functions L^j. If we want to exploit this, we have to develop a tailored SQP method which uses the fact that the multipliers ρ_j are equal to zero at the optimal solution z^* (cf. Lemma 3.6).

The idea is now to construct a Hessian approximation K by taking basically the exact Hessian, but leaving away all terms in which the multiplier ρ_j occur. More precisely, we propose to use in each step of the SQP algorithm the following Hessian approximation:

$$K := \left(\begin{array}{c|ccc|ccc} \mathcal{K}_{xx} & -\chi_0 L^0_{xw} & \dots & -\chi_n L^n_{xw} & 0 & \dots & 0 \\ \hline -\chi_0 L^0_{wx} & -\chi_0 L^0_{ww} - \mu_0^T B^0_{ww} & & & -(B^0_w)^T & & \\ \vdots & & \ddots & & & \ddots & \\ -\chi_n L^n_{wx} & & & -\chi_n L^n_{ww} - \mu^T B^n_{ww} & & & -(B^n_w)^T \\ \hline 0 & -B^0_w & & & & & \\ \vdots & & \ddots & & & 0 & \\ 0 & & & -B^n_w & & & \end{array}\right)$$

This choice for K has the advantage that the method can - under mild non-degeneracy assumptions [96] - be expected to converge locally q-quadratic. This can be proven by using the fact that the above Hessian approximation K converges locally at least linearly to the exact Hessian matrix, if we are using that the iterates for ρ_j can be expected to converge at least linearly to 0. This motivates the name asymptotically exact Hessian SQP algorithm. Note that the above convergence argumentation is so far only a sketch of a proof. A more consistent mathematical argumentation uses the Dennis-More theorem [66, 67, 182]. As such a proof is completely analogous to the argumentation in the proof of Theorem 9.2, which will later be discussed in different context in Chapter 9, we do not expand all details here.

Concluding this discussion, standard as well as tailored SQP methods can be applied to solve problems of the form (4.1.1). We have outlined the advantages and disadvantages of these algorithms. In the following section, we will consider an alternative algorithmic strategy which avoids some of the disadvantages of the above methods by exploiting the structure of min-max problems in a more natural way.

4.2 Sequential Convex Bilevel Programming

In this section, we develop an alternative to the SQP algorithm from the previous section, which exploits the structure of the problem in a more natural way. As for the tailored SQP method, the aim of our algorithm is to find a local minimizer

$$z^* := \begin{pmatrix} x^* \\ w^* \\ \lambda^* \end{pmatrix}$$

of problem (4.1.1) assuming that we can satisfy the necessary KKT-type conditions (3.3.11) with x^* being a minimizer of problem (4.0.1). Let us assume that we have an initial guess z^0 for the point z^*. Starting from this initial guess, we plan to perform iterates of the form

$$z^+ \;=\; z + \alpha\,\Delta z := \begin{pmatrix} x + \alpha\Delta x \\ w + \alpha\Delta w \\ \lambda + \alpha\,\Delta\lambda \end{pmatrix}$$

with $\alpha \in (0,1]$ being a damping parameter while the steps Δx, Δw, and $\Delta\lambda$ are assumed to be the primal dual local min-max point of the following convex bilevel quadratic program (min-max-QP):

$$\begin{aligned} \min_{\Delta x} \max_{\Delta w_0 \in \mathcal{B}_0^{\mathsf{lin}}} \quad & H^0 + L_x^0\Delta x + \left(\tfrac{1}{2}\Delta w_0^T L_{ww}^0 + \Delta x^T L_{xw}^0 + H_w^0\right)\Delta w_0 + \tfrac{1}{2}\Delta x^T K_{xx}\Delta x \\ \text{s.t.} \max_{\Delta w_i \in \mathcal{B}_i^{\mathsf{lin}}} \quad & H^i + L_x^i\Delta x + \left(\tfrac{1}{2}\Delta w_i^T L_{ww}^i + \Delta x^T L_{xw}^i + H_w^i\right)\Delta w_i \;\leq\; 0 \end{aligned} \tag{4.2.1}$$

with $i \in \{1,\ldots,n\}$ and

$$\mathcal{B}_j^{\mathsf{lin}} \;:=\; \left\{\, \Delta w_j \;\mid\; B_w^i\Delta w_i + B^i \;\leq\; 0 \,\right\} \tag{4.2.2}$$

for all $j \in \{0,\ldots,n\}$. Here, it should be explained that we use the notation $\Delta\lambda_j := \lambda_j^\dagger - \lambda_j$ to denote the steps to be taken in the multipliers of the lower level maximization problems, while $\Delta\chi := \chi^\dagger - \chi$ depends on the dual solution $\chi^\dagger$ which is associated with the inequality constraints in the minimization problem (4.2.1). Analogous to the conventions in the previous section, the iteration index is suppressed for ease of notation. Once a step has been performed we set the variable z to z^+ in order to continue with the next step. In particular, the symmetric and positive definite matrix

$$K_{xx} \in \mathbb{R}^{n_x \times n_x}$$

may change from iteration to iteration although this is in our notation not indicated by an iteration index. However, possible choices of this matrix K_{xx} will be discussed later, but we mention already at this point that K_{xx} should be a suitable approximation of the Hessian matrix

$$L^0_{xx} - \sum_{k=1}^{n} \chi_k L^k_{xx} \, .$$

Note that the sub-maximization problems within the min-max problem (4.2.1) can be regarded as convex quadratic programs (QPs) of the form

$$\begin{array}{rcll} V_i(\Delta x) & := & \max\limits_{\Delta w_i} & \frac{1}{2}\Delta w_i^T L^i_{ww} \Delta w_i + \left(\Delta x^T L^i_{xw} + H^i_w\right) \Delta w_i \\ & & \text{s.t.} & B^i_w \Delta w_i + B^i \ \leq \ 0 \, , \end{array} \tag{4.2.3}$$

as L^i_{ww} is assumed to be negative semi-definite (cf. Assumption 3.2). Moreover, the upper level minimization problem takes the form

$$\begin{array}{ll} \min\limits_{\Delta x} & H^0 + L^0_x \Delta x + V_0(\Delta x) + \frac{1}{2}\Delta x^T K_{xx} \Delta x \\ \text{s.t.} & H^i + L^i_x \Delta x + V_i(\Delta x) \ \leq \ 0 \, , \end{array} \tag{4.2.4}$$

which is a strictly convex optimization problem if K_{xx} is positive definite. Here, we have used the fact that the functions V_j are convex in Δx as the maximum over linear functions is convex. As for SQP methods, the existence of Δz is not guaranteed as the sub-problems might be infeasible. However, assuming that the sub-problems are feasible and that the convex quadratic programs (4.2.3) have unique solutions, we have a guarantee that the step Δz is unique. Moreover, the convexity has the practical advantage that the sub-problem can efficiently be solved with existing convex optimization tools.

In the case that L^i_{ww} is strictly negative definite, we can explicitly compute the dual of the convex QP problems (4.2.3). Provided that the QPs (4.2.3) admit strictly feasible points (Slater's condition) problem (4.2.4) is equivalent to a convex QCQP of the form

$$\begin{array}{ll} \min\limits_{\Delta x, \lambda^\dagger} & H^0 + L^0_x \Delta x - \frac{1}{2} g_0(\Delta x, \lambda^\dagger) \left(L^0_{ww}\right)^{-1} g_0(\Delta x, \lambda^\dagger)^T - B_0^T \lambda_0^\dagger + \frac{1}{2}\Delta x^T K_{xx} \Delta x \\ \text{s.t.} & H^i + L^i_x \Delta x - \frac{1}{2} g_i(\Delta x, \lambda^\dagger) \left(L^i_{ww}\right)^{-1} g_i(\Delta x, \lambda^\dagger)^T - B_i^T \lambda_i^\dagger \ \leq \ 0 \, . \end{array}$$

where we have used the short hand

$$g_j(\Delta x, \lambda^\dagger) \ := \ \Delta x^T L^j_{xw} - \left(\lambda^\dagger_j\right)^T B^j_w + H^j_w \, .$$

Note that this problem can solved with any suitable convex QCQP solver.

Remark 4.1: *Being at this point, it is helpful to state the differences between the outlined sequential convex bilevel programming methods and the SQP method from the previous section. For this aim, we observe that the equality constraint in the QP* (4.1.2) *can be used to eliminate the variables* Δw *in this QP, as long as we assume that* L_{ww} *is invertible. This leads to a relation of the form*

$$\begin{aligned} 0 &= L_w^j + \Delta x^T L_{xw}^j + \Delta w_j^T L_{ww}^j - \Delta \lambda_j^T B_w^j \qquad (4.2.5) \\ \iff \quad \Delta w_j &= -\left(L_{ww}^j\right)^{-1} g_j(\Delta x, \lambda^\dagger)^T \ . \end{aligned}$$

If we use this relation, the linearized upper level constraints in the QP (4.1.2) *have the form*

$$\begin{aligned} 0 &\geq L^i + L_x^i \Delta x + L_w^i \Delta w_i - B_i^T \Delta \lambda_i \qquad (4.2.6) \\ &= H^i + L_x^i \Delta x - L_w^i \left(L_{ww}^i\right)^{-1} g_i(\Delta x, \lambda^\dagger)^T - B_i^T \lambda_i^\dagger \ . \end{aligned}$$

This is a linear constraint in the variables Δw *and* $\Delta \lambda$ *while the corresponding constraint in the QCQP* (4.2.5) *is quadratic in* Δw *and* $\Delta \lambda$*. This must be interpreted as the main difference between the two methods. In another variant, we could also try to not linearize the upper level constraints, but expand them a little further taking some quadratic terms into account. More precisely, we can replace the linearized constraint of the form* (4.2.6) *with the quadratic inequality*

$$0 \geq L^i + L_x^i \Delta x + L_w^i \Delta w_i - B_i^T \Delta \lambda_i - \Delta \lambda_i^T B_w^i \Delta w_i + \Delta x^T L_{xw}^i \Delta w_i + \frac{1}{2} \Delta w_i L_{ww}^i \Delta w_i \ ,$$

which corresponds to a second order Tailor expansion, but leaves the possibly non-convex quadratic terms in Δx *away. If we now use relation* (4.2.5) *this quadratic expansion is equivalent to a constraint of the form*

$$0 \geq H^i + L_x^i \Delta x - \frac{1}{2} g_i(\Delta x, \lambda^\dagger) \left(L_{ww}^i\right)^{-1} g_i(\Delta x, \lambda^\dagger)^T - B_i^T \lambda_i^\dagger \ . \qquad (4.2.7)$$

This is exactly the constraint which occurs in the QCQP (4.2.5)*. Thus, we have now two ways to derive the QCQP sub-problem* (4.2.5)*, which must be solved in each step of the algorithm.*

Definition 4.1: *We define for each* $j \in \{0, \ldots, n\}$ *the lower level working set* $A_j(\lambda^\dagger)$ *by*

$$\mathcal{A}_j(\lambda^\dagger) := \left\{ k \in \{0, \ldots, n\} \;\middle|\; \left(\lambda_j^\dagger\right)_k > 0 \right\} \ . \qquad (4.2.8)$$

Moreover, we denote the number of elements in $\mathcal{A}_j(\lambda^\dagger)$ *by* $m_j := \left|\mathcal{A}_j(\lambda^\dagger)\right|$.

We use the above notation to introduce the lower level KKT matrices

$$\Omega_j \;:=\; \begin{pmatrix} L^j_{ww} & \left(B^{j,\mathsf{act}}_w\right)^T \\ B^{j,\mathsf{act}}_w & 0 \end{pmatrix}, \tag{4.2.9}$$

where $B^{j,\mathsf{act}}_w \in \mathbb{R}^{m_j \times n_w}$ is a matrix which consists of the rows of B^j_w, whose index is in the working set $\mathcal{A}_j(\lambda^\dagger)$.

Assumption 4.1: *We assume that for all iterates and for all $j \in \{0,\ldots,n\}$ the matrix L^j_{ww} is negative semi-definite while the matrix Ω_j is invertible.*

Note that the above assumption seems reasonable in our context as we are interested in the case that the lower level optimization problems are convex while a non-degeneracy assumption (or reduction ansatz) holds in the optimal solution. In this sense, the above assumption is not excessively restrictive requiring a kind of regularity condition to be satisfied during the iterations.

Proposition 4.1: *If Assumption 4.1 holds, the bilevel optimization problem* (4.2.1) *can equivalently be regarded as an MPCC. Here, the condition that the pairs $(\Delta w_j, \lambda^\dagger_j)$ are primal-dual maximizers can for all $j \in \{0,\ldots,n\}$ equivalently be replaced by the corresponding KKT conditions*

$$0 \;=\; \Delta x^T L^j_{xw} + \Delta w_j^T L^j_{ww} - \Delta\lambda_j^T B^j_w + L^j_w \tag{4.2.10}$$

$$0 \;\geq\; B^j_w \Delta w_j + B^j \tag{4.2.11}$$

$$0 \;\leq\; \lambda_j + \Delta\lambda_j \;=\; \lambda^\dagger_j \tag{4.2.12}$$

$$0 \;=\; \left(B^j_w \Delta w_j + B^j\right)^T \lambda^\dagger_j \tag{4.2.13}$$

using the notation $L^j_w \;:=\; H^j_w - \lambda_j^T B^j_w$.

Proof: The above Proposition should be self-explaining: the conditions (4.2.10)-(4.2.13) are simply the necessary KKT optimality conditions for the lower-level QPs (4.2.3). Here, Assumption 4.1 guarantees both: first, the invertibility of Ω_j implies a linear independence constraint qualification which justifies the application of the KKT theorem, and second it implies concavity of the QPs and thus the sufficiency of the first order KKT conditions. □

Remark 4.2: *The above Proposition shows that the bilevel optimization problem* (4.2.1) *can be regarded as a mathematical program with linear complementarity constraints (MPLCC), which are in their general form rather expensive and difficult to solve [58, 154]. Note that the special structure arising from the semi-infinite programming context as well as the convexity of the bilevel problem* (4.2.1) *are the foundation of the presented sequential convex bilevel programming method, which make it efficient. This aspect is also the main difference of the presented method in comparison to techniques like piecewise sequential quadratic programming methods for general MPCCs [162, 196, 243], where a quadratic program with linear complementarity constraints (QPLCC) must be solved in each step of the sequential method.*

In the next step we work out the optimality conditions for the bilevel QP (4.2.1). For this aim, we introduce the matrices $R_j \in \mathbb{R}^{n_x \times (n_w + m_j)}$ as well as the vectors $s_j \in \mathbb{R}^{n_w + m_j}$ (with $j \in \{0, \ldots, n\}$) which are defined as

$$R_j \; := \; \begin{pmatrix} L^j_{w,x} \\ 0 \end{pmatrix} \quad \text{and} \quad s_j \; := \; \begin{pmatrix} (H^j_w)^T \\ B^{j,\mathsf{act}} \end{pmatrix} \tag{4.2.14}$$

respectively. Here, the matrix $B^{j,\mathsf{act}}$ consists of all components of B^j, whose index is in the working set $\mathcal{A}_j(\lambda^\dagger)$. Moreover, we use the notation $T_j := R_j^T \Omega_j^{-1} R_j$.

Definition 4.2: *Requiring that Assumption 4.1 is satisfied, we say that the QP* (4.2.3) *is nondegenerate for a given Δx if the strict complementarity condition (SCC)*

$$B^j_w \Delta w_j + B^j - \lambda^\dagger_j \; < \; 0 \, . \tag{4.2.15}$$

holds at the primal dual solution $(\Delta w_j, \lambda^\dagger_j)$ of the QP (4.2.3) *.*

Assumption 4.2: *We assume that all lower level QPs of the form* (4.2.3) *are non-degenerate at the solution $(\Delta x, \Delta w, \lambda^\dagger)$ of problem* (4.2.1)*, i.e., the strict inequality* (4.2.15) *is satisfied at this point for all indices $j \in \{0, \ldots, n\}$.*

Note that the non-degeneracy of the j-th lower level QP at a given Δx implies that the variables Δw_j and $\lambda^\dagger_j$ can in a neighborhood of Δx be regarded as a locally linear function. This is due to the fact that Assumption 4.1 is equivalent to the LICQ and SOSC condition for the lower level QPs while the SCC condition is required by the above definition.

Definition 4.3: *Let the point* $(\Delta x, \Delta w, \lambda^\dagger)$ *be a feasible point of the bilevel problem* (4.2.1)*. Providing that Assumption 4.1 is satisfied, we say that the extended LICQ condition is satisfied at* $(\Delta x, \Delta w, \lambda^\dagger)$ *if the vectors*

$$L_x^k - s_k^T \Omega_k^{-1} R_k + \Delta x^T T_k^T \qquad \forall k \in \mathcal{W} \tag{4.2.16}$$

are linearly independent. Here,

$$\mathcal{W} \;:=\; \left\{ k \;\mid\; \left(L_x^i \Delta x + L_w^i \Delta w_i - \Delta \lambda_i^T B^i + L^i \right)_k \;=\; 0 \right\}$$

denotes the active set which is associated with the upper level constraints.

Lemma 4.1: *Let Assumptions 4.1 and 4.2 be satisfied. Furthermore, let the point* $(\Delta x, \Delta w, \lambda^\dagger)$ *be a minimizer of problem* (4.2.1) *for which the extended LICQ-condition holds. Now, we have necessarily*

$$\begin{aligned} 0 \;&=\; \left(K_{xx} - T_0 + \sum_{k=1}^{n} \chi_k^\dagger T_k \right) \Delta x + \left(L_x^0 - s_0^T \Omega_0^{-1} R_0 \right)^T \\ &\quad - \sum_{k=1}^{n} \chi_k^\dagger \left(L_x^k - s_k^T \Omega_k^{-1} R_k \right)^T \end{aligned} \tag{4.2.17}$$

$$0 \;\geq\; H^i + L_x^i \Delta x - \frac{1}{2} \left(R_i \Delta x + s_i \right)^T \Omega_i^{-1} \left(R_i \Delta x + s_i \right) \tag{4.2.18}$$

$$0 \;\geq\; \chi + \Delta \chi \;:=\; \chi^\dagger \tag{4.2.19}$$

$$0 \;=\; \left(H^i + L_x^i \Delta x - \frac{1}{2} \left(R_i \Delta x + s_i \right)^T \Omega_i^{-1} \left(R_i \Delta x + s_i \right) \right) \chi_i^\dagger \tag{4.2.20}$$

for all $i \in \{1, \ldots, n\}$*. Furthermore, the multiplier* $\chi^\dagger$ *is unique.*

Proof: Due to the non-degeneracy Assumption 4.2 for the lower level QPs (4.2.3) the bilevel problem (4.2.1) is locally equivalent to an auxiliary quadratically constrained quadratic program of the form

$$\begin{array}{ll} \min\limits_{\Delta x} & \frac{1}{2} \Delta x^T K_{xx} \Delta x + L_x^0 \Delta x - \frac{1}{2} \left(R_0 \Delta x + s_0 \right)^T \Omega_0^{-1} \left(R_0 \Delta x + s_0 \right) \\ \text{s.t.} & H^i + L_x^i \Delta x - \frac{1}{2} \left(R_i \Delta x + s_i \right)^T \Omega_i^{-1} \left(R_i \Delta x + s_i \right) \;\leq\; 0 \end{array} \quad . \tag{4.2.21}$$

This follows immediately from a local elimination of the variable Δw on dependence on Δx, i.e., we know that the active set of the lower level QPs remains locally constant in Δx such that we can exploit the relation

$$R_j \Delta x + \Omega_j \begin{pmatrix} \Delta w_j \\ -\lambda_j^{\dagger,\mathsf{act}} \end{pmatrix} + s_j \;=\; 0\,, \tag{4.2.22}$$

which summarizes the parameterized stationarity as well as the primal feasibility condition of the active constraints associated with the j-th sub-QP (4.2.3). In this notation, $\lambda_j^{\dagger,\mathsf{act}}$ is the vector which consists of the non-zero components of $\lambda_j^{\dagger}$. Now, the extended LICQ condition for the bilevel problem (4.2.1) reduces to a standard LICQ condition for the auxiliary problem (4.2.21). Consequently, an application of the KKT theorem yields the statement of the Lemma. □

4.3 Local Convergence Analysis

The local convergence properties of the presented sequential convex bilevel programming method are much easier to discuss than the global convergence. Basically, we can transfer the classical concepts for the local analysis of standard SQP theory. Thus, we will in this section present the local convergence theory on an adequate advanced level aiming at remarks on the details which are specific for sequential convex bilevel programming methods.

Let us directly constrain ourselves to the assumption that the active set during the local phase of the algorithm is already correctly detected and stable. For this aim, we recall that the min-max problem (4.0.1) is equivalent to the MPCC (3.4.1). Similarly, the extended LICQ condition (ELICQ) for the min-max problem (4.0.1) (cf. Definition (3.5)) is equivalent to the MPCC-LICQ condition[1] for the problem MPCC (3.4.1), as this follows immediately from Lemma 3.6. Here, the stability of the active set can in our context be guaranteed as follows:

Assumption 4.3 (Strong Regularity): *We assume that at the local MPCC minimizer* (x^*, w^*, λ^*) *of our interest the following strong regularity conditions are satisfied:*

[1] For a deeper discussion of MPCC-LICQ and MPCC-MFCQ conditions, we refer to [93], where these conditions are defined via tightened nonlinear programming problems (TNLP). Similar considerations can be found in [98].

1. *The solution w^* of the lower level maximization problems is nondegenerate.*
2. *The ELICQ (or equivalently the MPCC-LICQ) condition is satisfied at (x^*, w^*, λ^*).*
3. *The second order sufficient condition, as defined in Theorem 3.3, is satisfied.*
4. *The upper level strict complementarity condition*

$$L_i(x^*, w_i^*, \lambda_i^*) - \chi_i^* \quad < \quad 0 \tag{4.3.1}$$

holds for all $i \in \{1, \ldots, n\}$.

Lemma 4.2: *Let (x^*, w^*, λ^*) be a local minimizer of the MPCC* (3.4.1) *at which the regularity Assumption 4.3 is satisfied. Then there exists a neighborhood of (x^*, w^*, λ^*) in which the bilevel optimization admits a feasible solution Δz which has the same active set as the local minimizer (x^*, w^*, λ^*), i.e., we have $\mathcal{A}_j(\lambda^\dagger) = \mathcal{A}_j^*$ for all $j \in \{0, \ldots, n\}$ as well as $\mathcal{A}(\chi^\dagger) := \{ k \mid \chi_k > 0 \} = \mathcal{A}^*$ for all iterates in this neighborhood.*

Proof: The feasibility as well as the stability of the active set for the lower level QPs follows immediately from Robinson's theorem [199, 200]. Similarly, we can also apply Robinson's theorem to the upper level auxiliary problem (3.3.12). Thus, we obtain the feasibility and active set stability of the local QP-type necessary conditions from Lemma 4.1. Here, we use that the ELICQ condition boils down to an LICQ condition for problem (3.3.12) while the second order sufficient condition from Theorem 3.3 is equivalent to the SOCS condition for problem (3.3.12). As the fourth requirement of Assumption 4.3 guarantees the SCC condition for problem (3.3.12), we have all the necessary regularity conditions for problem (3.3.12) such that an application of Robinson's theorem is justified. Thus, we conclude the statement of the theorem. □

A question which we have not discussed so far is how we should choose the matrix K_{xx}. In the following section we will assume that K_{xx} is positive definite as this makes the discussion of the global convergence properties more convenient. However, such an assumption is in principle not necessary for the discussion of local convergence properties, although it is still desirable in the sense that it guarantees the convexity of the sub-problems. In the context of local convergence, we are rather interested in a Dennis-Moré condition of the form

$$\left\| \left(K_{xx}^m - \frac{\partial^2}{\partial x^2} L_0(x^*, w_0^*, \lambda_0^*) + \sum_{k=1}^{n} \chi_k^* \frac{\partial^2}{\partial x^2} L_k(x^*, w_k^*, \lambda_k^*) \right) \Delta x^m \right\| \leq c_m \left\| \Delta x^m \right\| \tag{4.3.2}$$

where $(c_m)_{m\in\mathbb{N}}$ is a non-negative real valued sequence. Note that - with quite some abuse of notation - the iteration index m has been recovered in this formulation recalling that the Hessian approximation $K_{xx} = K_{xx}^m$ may change from iteration to iteration.

Theorem 4.1: *Let Assumption* (4.3) *be satisfied while the Hessian approximation sequence K_{xx}^m satisfies the Dennis-Moré estimate* (4.3.2) *for a sequence $(c_m)_{m\in\mathbb{N}}$. Moreover, we assume that the sequential convex bilevel programming method takes - at least close to the solution - always full-steps while the functions H_i and B have Lipschitz continuous Hessians. Now, the following statements hold:*

- *If the sequence $(c_m)_{m\in\mathbb{N}}$ satisfies $\lim_{m\to\infty} c_m = 0$, then the local convergence of the sequential convex bilevel programming method is r-superlinear.*
- *If the sequence $(c_m)_{m\in\mathbb{N}}$ satisfies $c_{m+1} \leq \kappa\, c_m$ for some $\kappa < 1$, then the local convergence of the sequential convex bilevel programming method is r-quadratic.*

Proof: Using Lemma 4.2 our aim is to show that the sequential convex bilevel programming method is locally equivalent to a Newton type method applied to the necessary conditions (3.3.11) from Theorem 3.2 under the assumption that the active set is fixed. As Proposition 4.1 show already that the sequential convex bilevel programming method linearizes the primal feasibility condition of the lower level problem in every step exactly, we discuss directly the linearization of the active upper level constraint:

$$\begin{aligned} L_i + \frac{\partial}{\partial z} L_i\, \Delta z &= L^i + L_x^i \Delta x + L_w^i \Delta w - \Delta\lambda_i^T B_i \\ &= H^i + L_x^i \Delta x + \left(\frac{1}{2}\Delta w_i^T L_{ww}^i + \Delta x^T L_{xw}^i + H_w^i\right)\Delta w_i \\ &\quad + \frac{1}{2}\Delta w_i L_{ww}^i \Delta w_i + L_w^i \Delta w_i \\ &= H^i + L_x^i \Delta x + \left(\frac{1}{2}\Delta w_i^T L_{ww}^i + \Delta x^T L_{xw}^i + H_w^i\right)\Delta w_i \\ &\quad - \frac{1}{2}\Delta w_i L_{ww}^i \Delta w_i - \Delta x L_{xw}^i \Delta w_i + \Delta\lambda_i^T B_w^i \Delta w\,, \end{aligned} \tag{4.3.3}$$

which leads to

$$\left\| L_i + \frac{\partial}{\partial z} L_i\, \Delta z \right\| \leq O(\|\Delta z\|^2) \tag{4.3.4}$$

for all i in the active set, i.e., for all i with

$$H^i + L_x^i \Delta x + \left(\frac{1}{2} \Delta w_i^T L_{ww}^i + \Delta x^T L_{xw}^i + H_w^i \right) \Delta w_i \;\; = \;\; 0 \, .$$

It remains to discuss the Newton step with regard to the stationarity equation

$$0 \;\; = \;\; \frac{\partial}{\partial x} \mathcal{K}(x, w, \lambda) \;\; = \;\; \frac{\partial}{\partial x} L_0(x, w_0, \lambda_0) - \sum_{k=1}^{n} \chi_k \frac{\partial}{\partial x} L_k(x, w_k, \lambda_k) \, . \quad (4.3.5)$$

A linearization of the above expression for $\frac{\partial}{\partial x}\mathcal{K}$ leads to

$$\begin{aligned} \frac{\partial}{\partial x}\mathcal{K} + \frac{\partial}{\partial z}\left[\frac{\partial}{\partial x}\mathcal{K}\right] \Delta z \;\; &= \;\; \left(L_{xx}^0 \Delta x + L_{xw}^0 \Delta w_0 + L_x^0 \right) \\ &\quad - \sum_{k=1}^{n} \chi_k \left(L_{xx}^k \Delta x + L_{xw}^k \Delta w_k + L_x^k \right) - \sum_{k=1}^{n} \Delta \chi_k L_x^k \, . \quad (4.3.6) \end{aligned}$$

Note that we may assume $\Delta \lambda_j^{\mathsf{inact}} = 0$ during the local phase as we consider the case that the correct active set has already settled. Combining this knowledge with the relation

$$R_j \Delta x + \Omega_j \begin{pmatrix} \Delta w_j \\ -\lambda_j^{\dagger,\,\mathsf{act}} \end{pmatrix} + s_j = 0$$

we can further transform to

$$\begin{aligned} \frac{\partial}{\partial x}\mathcal{K} + \frac{\partial}{\partial z}\left[\frac{\partial}{\partial x}\mathcal{K}\right] \Delta z \;\; &= \;\; \left(L_{xx}^0 - \sum_{k=1}^{n} \chi_k L_{xx}^k \right) \Delta x - T_0 \Delta x + \sum_{k=1}^{n} \chi_k T_k \Delta x \\ &\quad - R_0^T \Omega_0^{-1} s_0 + \sum_{k=1}^{n} \chi_k R_0^T \Omega_0^{-1} s_0 + L_x^0 - \sum_{k=1}^{n} \chi_k^\dagger L_x^k \, . \quad (4.3.7) \end{aligned}$$

Using the result of Lemma 4.1 in combination with the Lipschitz continuity of the Hessian terms as well as the Dennis-Moré estimate (4.3.2) we obtain

$$\begin{aligned} \left\| \frac{\partial}{\partial x}\mathcal{K} + \frac{\partial}{\partial z}\left[\frac{\partial}{\partial x}\mathcal{K}\right] \Delta z \right\| \;\; &\leq \;\; \left(c_m + O(\|z - z^*\|) \right) \|\Delta x\| \\ &\quad + \left\| - \sum_k \Delta \chi_k T_k \Delta x - \sum_k \Delta \chi_k R_k \Omega_k^{-1} s_k \right\| \quad (4.3.8) \end{aligned}$$

Now, we use that

$$\left\| R_k \Omega_k^{-1} s_k \right\| \;=\; \| -T_k \Delta x - L_{xw} \Delta w \| \;\leq\; O(\|\Delta z\|) \tag{4.3.9}$$

to finally conclude

$$\left\| \frac{\partial}{\partial x}\mathcal{K} + \frac{\partial}{\partial z}\left[\frac{\partial}{\partial x}\mathcal{K}\right]\Delta z \right\| \;\leq\; (c_m + O(\|z - z^*\|))\,\|\Delta x\| + O(\|\Delta z\|^2)\,. \tag{4.3.10}$$

Note that this last estimate (4.3.10) together with the estimate (4.3.4) boil down to a standard Dennis-Moré convergence criterion for the Newton method applied to the optimality conditions with respect to the fixed active set. Both statements of the theorem are a direct consequence. □

Remark 4.3: *Note that the above theorem covers the case that K_{xx}^m is generated by BFGS updates, for which superlinear convergence is obtained. In the case of exact Hessian approximations we have even quadratic convergence. This is in analogy to standard SQP methods.*

Remark 4.4: *Note that the above local convergence result could be generalized for the case that the second order matrices L_{ww}, L_{wx}, and L_{xw} do not exactly coincide with their associated second order terms as long as they are suitable approximations. However, for such an "inexact" sequential convex bilevel programming method, the global convergence argumentation from the following section would fail, as an approximation of these second order terms would amount to an inexact linearization of the lower level stationarity conditions, which are in the MPCC* (3.4.1) *formulated as equality constraints.*

4.4 Global Convergence Analysis

A crucial point in the discussion of global convergence of any SQP type method is the availability of a merit function which measures the progress of the iterations $z^+ = z + \alpha \Delta z$ towards a local minimum. This can for example be achieved via line search techniques [182] adjusting the damping parameter α if necessary but also trust region methods [60] make use of merit functions. In standard SQP methods with suitable regularity assumptions Han's exact l_1-penalty function [115] is a traditional choice but there are also other choices [182].

Note that for general MPCCs it is not straightforward to transfer the idea of penalty functions as most of the techniques, as e.g. discussed in [182], are based on the assumption that a suitable constraint qualification holds. As MPCCs do often not satisfy these constraint qualifications, standard proof techniques typically fail. Global convergence of SQP methods for general MPCCs are an active field of research and we refer to [7, 23] for further reading on global convergence of methods and a discussion of penalty functions for general MPCCs.

Fortunately, the MPCC (3.4.1) arises from the context of semi-infinite programming and it has a special structure which is exploited in the method presented in this thesis. This helps us also to construct a suitable merit function for our needs. Let us start by defining an upper level merit function $\Phi_U : \mathbb{R}^{n_x} \times \mathbb{R}^{(n+1)n_w} \times \mathbb{R}^{(n+1)n_B} \to \mathbb{R}$ planning to measure the progress in terms of upper level feasibility and optimality in the form

$$\Phi_U(x, w, \lambda) \quad := \quad L_0(x, w_0, \lambda_0) + \sum_{k=1}^{n} \hat{\chi}_k \, \pi_k(L_k(x, w_k, \lambda_k)) \, , \tag{4.4.1}$$

where $\hat{\chi} \in \mathbb{R}^n_{++}$ is a constant vector. Here, it should be explained that the positive projection $\pi : \mathbb{R}^d \to \mathbb{R}^d_+$ is defined for arbitrary dimensions d while the components of π satisfy

$$\forall s \in \mathbb{R}^d, \forall k \in \{1, \ldots, d\} : \quad \pi_k(s) \; := \; \max\{\, 0, \; s_k \,\} \, .$$

Similarly, $|\cdot| : \mathbb{R}^d \to \mathbb{R}^d_+$ is also defined for arbitrary d where $|s|$ denotes the component-wise absolute value of a vector $s \in \mathbb{R}^d$.

Beside the upper-level feasibility, we also need to measure the violation of the stationarity and primal feasibility condition for the lower level optimization problems. In this context, we observe that the dual feasibility condition $\lambda^+ \geq 0$ is automatically satisfied for the iterates. Thus, a violation of dual feasibility in the lower level problems does not need to be detected motivating the introduction of primal lower level merit functions of the form $\Phi^j_L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \times \mathbb{R}^{n_B} \to \mathbb{R}$ which are defined as

$$\Phi^j_L(x, w_j, \lambda_j) \quad := \quad \left| \frac{\partial L_j(x, w_j, \lambda_j)}{\partial w} \right| \hat{\rho}_j + \hat{\lambda}_j^T \, \pi(B(x, w_j))$$

for all $j \in \{0, \ldots, n\}$. Here, $\hat{\rho}_j \in \mathbb{R}^{n_w}_{++}$ and $\hat{\lambda}_j \in \mathbb{R}^{n_B}_{++}$ are positive constants. The final step is to compose a merit function $\Phi : \mathbb{R}^{n_x} \times \mathbb{R}^{(n+1)n_w} \times \mathbb{R}^{(n+1)n_B} \to \mathbb{R}$ as

$$\Phi(x, w, \lambda) \quad := \quad \Phi_U(x, w, \lambda) + \Phi^0_L(x, w_0, \lambda_0) + \sum_{k=1}^{n} \hat{\chi}_k \Phi^k_L(x, w_k, \lambda_k) \, . \tag{4.4.2}$$

In the following, we prepare the proof of Theorem 4.2 where a condition for a descent direction of the merit function Φ will be discussed. In this context we make use of the following assumption:

Assumption 4.4: *The matrix L_{ww} is negative definite.*

Let us introduce the short-hand "∂_α" to denote one sided directional derivatives in the step direction, i.e., we define for example

$$\partial_\alpha L_0(x, w_0, \lambda_0) := \lim_{\alpha \to 0^+} \frac{L_0(x + \alpha\Delta x, w_0 + \alpha\Delta w_0, \lambda_0 + \alpha\Delta\lambda_0) - L_0(x, w_0, \lambda_0)}{\alpha}. \tag{4.4.3}$$

This abstract notation for one sided derivatives can analogously be transferred for the other terms in the merit function. Let us state the following technical result:

Proposition 4.2: *Transferring the notation* (4.4.3) *to denote one-sided directional derivatives, the following expressions exist (the corresponding limits for $\alpha \to 0^+$ exist) and satisfy*

$$\partial_\alpha L_0(x, w_0, \lambda_0) = L_x^0 \Delta x + L_w^0 \Delta w_0 - \Delta\lambda_0^T B^0 \tag{4.4.4}$$

$$\partial_\alpha \pi(L_i(x, w_i, \lambda_i)) \leq -\pi(L^i) - \frac{1}{2} L_w^i \left(L_{ww}^i\right)^{-1} {L_w^i}^T \tag{4.4.5}$$

$$\partial_\alpha \pi(B(x, w_j)) \leq -\pi(B^j) \tag{4.4.6}$$

$$\partial_\alpha \left| \frac{\partial}{\partial w} L_j(x, w_j, \lambda_j) \right| = -\left| L_w^j \right| \tag{4.4.7}$$

for all $i \in \{1, \ldots, n\}$ and all $j \in \{0, \ldots, n\}$. Here, formula (4.4.5) *requires Assumption 4.4 to be satisfied.*

Proof: The first formula (4.4.4) follows immediately from definition (4.4.3). Moreover, the conditions from the lower level QP optimality

$$B_w^j \Delta w_j \leq -B^j\,, \tag{4.4.8}$$

$$\text{and} \quad \Delta x^T L_{xw}^j + \Delta w_j^T L_{ww}^j - \Delta\lambda_j^T B_w^j = -L_w^j \tag{4.4.9}$$

can be used to estimate the remaining directional derivatives (4.4.6) and (4.4.7) respectively. It remains to verify estimate (4.4.5). For this aim, we first compute for all $i \in \{1, \dots, n\}$ the term

$$\begin{aligned}
\partial_\alpha L_i &= L_x^i \Delta x + L_w^i \Delta w_i - \Delta\lambda_i^T B^i \\
&\leq -H^i - \left(\frac{1}{2}\Delta w_i^T L_{ww}^i + \Delta x^T L_{xw}^i + H_w^i\right)\Delta w_i + L_w^i \Delta w_i - \Delta\lambda_i^T B^i \\
&= -L^i - \left(\frac{1}{2}\Delta w_i^T L_{ww}^i + \Delta x^T L_{xw}^i + H_w^i\right)\Delta w_i + L_w^i \Delta w_i - (\Delta\lambda_i^\dagger)^T B^i \\
&\overset{(4.2.13)}{=} -L^i - \left(\frac{1}{2}\Delta w_i^T L_{ww}^i + \Delta x^T L_{xw}^i + H_w^i - (\Delta\lambda_i^\dagger)^T B_w^i\right)\Delta w_i + L_w^i \Delta w_i \\
&\overset{(4.2.10)}{=} -L^i + \frac{1}{2}\Delta w_i^T L_{ww}^i \Delta w_i + L_w^i \Delta w_i \\
&= -L^i + \frac{1}{2}\left(L_{ww}^i \Delta w_i + (L_w^i)^T\right)^T \left(L_{ww}^i\right)^{-1}\left(L_{ww}^i \Delta w_i + (L_w^i)^T\right) \\
&\quad -\frac{1}{2} L_w \left(L_{ww}^i\right)^{-1} {L_w^i}^T \\
&\leq -L^i - \frac{1}{2} L_w^i \left(L_{ww}^i\right)^{-1} {L_w^i}^T . \qquad (4.4.10)
\end{aligned}$$

In the last step, we have used that L_{ww}^i is negative definite. Estimate (4.4.5) is now a direct consequence. □

Definition 4.4: *Provided Assumption 4.4 is satisfied, we introduce the notation*

$$\rho_j \;:=\; \left(L_{ww}^j\right)^{-1} {L_w^j}^T$$

for all $j \in \{0, \dots, n\}$.

Assumption 4.5: *We assume that the matrix* K_{xx} *is symmetric and positive definite.*

In the following Theorem we discuss that the presented sequential convex bilevel programming method generates descent directions of the function Φ:

Theorem 4.2 (Compatibility of the merit function): *Let us assume that z is a given iterate of the above sequential bilinear programming method for which the bilevel quadratic optimization problem* (4.2.1) *admits a feasible solution Δz while Assumptions 4.1, 4.2, 4.4 and 4.5 are satisfied. Furthermore, we assume that the weights in the merit function Φ are sufficiently large such that we have*

$$\forall j \in \{0,\ldots,n\}: \quad \hat{\chi} > \left|\chi^{\dagger}\right| \ , \ \hat{\rho}_k > \frac{3}{2}\left|\rho_k\right| \ , \ \hat{\lambda}_j > 0 \ . \tag{4.4.11}$$

Now, we have either

$$\Delta x = 0 \ , \ \pi(B^j) = 0 \ , \ \pi(L^i) = 0 \ , \ \left|L_w^j\right| = 0 \ ,$$

$$\rho_j = 0 \ , \ \Delta w_j = 0 \ , \ \textit{and} \ \ \lambda_j^T B^j = 0 \tag{4.4.12}$$

for all $i \in \{1,\ldots,n\}$ and all $j \in \{0,\ldots,n\}$ or Δz is a descent direction of the merit function Φ, i.e., we have

$$\partial_\alpha \Phi \ := \ \lim_{\alpha \to 0^+} \frac{\Phi(\, x + \alpha \Delta x,\, w + \alpha \Delta w,\, \lambda + \alpha \Delta \lambda\,) - \Phi(x,w,\lambda)}{\alpha} \ < \ 0 \tag{4.4.13}$$

Proof: In the first step of this proof, we use the formula (4.4.4) in combination with the linearized stationarity conditions (4.2.17) to compute

$$\begin{aligned}
\partial_\alpha L_0(x,w_0,\lambda_0) \quad &= \quad L_x^0 \Delta x + L_w^0 \Delta w_0 - \Delta \lambda_0^T B^0 \\
&\overset{(4.2.17)}{=} \quad -\Delta x^T K_{xx} \Delta x + \Delta x^T T_0 \Delta x + s_0^T \Omega_0^{-1} R_0 \Delta x + L_w^0 \Delta w_0 \\
&\qquad -\Delta \lambda_0^T B^0 - \sum_{k=1}^{n} \chi_k^{\dagger} \left(-L_x^k \Delta x + \Delta x^T T_k \Delta x + s_k^T \Omega_k^{-1} R_k \Delta x \right)
\end{aligned}$$

By collecting terms, the above equation can also be summarized in the form

$$\partial_\alpha L_0(x,w_0,\lambda_0) \ = \ -\Delta x^T K_{xx} \Delta x + X_0 - \sum_{k=1}^{n} \chi_k^{\dagger} X_k - \sum_{k=1}^{n} \chi_k^{\dagger} L^k \ , \tag{4.4.14}$$

where we use the short hands

$$X_0 \ := \ \Delta x^T T_0 \Delta x^T + s_0^T \Omega_0^{-1} R_0 \Delta x + L_w^0 \Delta w_0 - \Delta \lambda_0^T B^0 \tag{4.4.15}$$

and

$$X_k \;:=\; -L_k - L_x^k \Delta x + \Delta x^T T_k \Delta x^T + s_k^T \Omega_k^{-1} R_k \Delta x \,. \tag{4.4.16}$$

for $k \in \{1, \ldots, n\}$. Now, the basic strategy is to use the necessary optimality conditions to transform the expressions for X_0 and X_k and completing squares in such a way that we can find suitable estimates for them. We start with the term for X_0:

$$\begin{aligned} X_0 \;&:=\; \Delta x^T T_0 \Delta x + s_0^T \Omega_0^{-1} R_0 \Delta x + L_w^0 \Delta w_0 - \Delta \lambda_0^T B^0 \\ &=\; -\Delta x^T L_{xw}^0 \Delta w_0 + L_w^0 \Delta w_0 - \Delta \lambda_0^T B^0 \,. \end{aligned} \tag{4.4.17}$$

The latter equality can be verified by multiplying equation (4.2.22) with $\Delta x^T R_0^T \Omega_0^{-1}$ from the left. In the next step we use the stationarity condition for the lower QP to further transform to

$$\begin{aligned} X_0 \;&=\; \Delta w_0 L_{ww}^0 \Delta w_0 + 2 L_w^0 \Delta w_o - \Delta \lambda_0^T B_w^0 \Delta w - \Delta \lambda_0^T B^0 \\ &=\; \left(L_{ww}^0 \Delta w_0 + L_w^0 \right) \left(L_{ww}^0 \right)^{-1} \left(L_{ww}^0 \Delta w_0 + L_w^0 \right) \\ &\quad\; - L_w^0 \left(L_{ww}^0 \right)^{-1} {L_w^0}^T + \lambda_0^T \left(B_w^0 \Delta w_0 + B^0 \right) \,. \end{aligned} \tag{4.4.18}$$

The first term in the right side of the above transformation is negative as L_{ww} is negative definite. Similarly, we have $\lambda_0^T \left(B_w^0 \Delta w + B^0 \right) \leq 0$ as $\lambda_0 \geq 0$ and $B_w^0 \Delta w + B^0 \leq 0$. Thus, we find

$$X_0 \;\leq\; -L_w^0 \left(L_{ww}^0 \right)^{-1} {L_w^0}^T \,. \tag{4.4.19}$$

In order to obtain a similar estimate for X_k with $k \in \{1, \ldots, n\}$ we use the complementarity relation (4.2.20) to find

$$\begin{aligned} X_k \;&=\; -L_k - L_x^k \Delta x + \Delta x^T T_k \Delta x^T + s_k^T \Omega_k^{-1} R_k \Delta x \\ &\overset{(4.2.20)}{=}\; -\frac{1}{2} \Delta w_k^T L_{ww}^k \Delta w_k + \Delta w_k^T {B_w^k}^T \lambda^\dagger + \lambda_k^T B_k - \Delta x^T L_{xw} \Delta w \\ &=\; \frac{1}{2} \Delta w_k^T L_{ww}^k \Delta w_k + L_w^k \Delta w_k - \Delta \lambda_k B_w^k \Delta w_k + {\lambda^\dagger}^T B_w^k \Delta w_k + \lambda_k^T B_k \\ &=\; \frac{1}{2} \left(L_{ww}^k \Delta w_k + L_w^k \right) \left(L_{ww}^k \right)^{-1} \left(L_{ww}^k \Delta w_k + L_w^k \right) \\ &\quad\; - \frac{1}{2} L_w^k \left(L_{ww}^k \right)^{-1} {L_w^k}^T + \lambda_k^T \left(B_w^k \Delta w_k + B^k \right) \end{aligned} \tag{4.4.20}$$

Thus, we have

$$X_k \;\leq\; -\frac{1}{2} L_w^k \left(L_{ww}^k\right)^{-1} {L_w^k}^T . \tag{4.4.21}$$

In the next step, we are interested in computing the directional derivative of the upper-level merit function Φ_U. For this aim, we use equation (4.4.14) to find

$$\begin{aligned}\partial_\alpha \Phi_U \;&\leq\; -\Delta x^T K_{xx} \Delta x + X_0 - \sum_{k=1}^n \chi_k^\dagger X_k - \sum_{k=1}^n \left(\hat{\chi} + \chi_k^\dagger\right) \pi_k(L^k) \\ &\quad - \sum_{k=1}^n \hat{\chi} L_w^k \left(L_{ww}^k\right)^{-1} {L_w^k}^T \\ &\leq\; X_0 - \sum_{k=1}^n \chi_k^\dagger X_k - \sum_{k=1}^n \hat{\chi}_k L_w^k \left(L_{ww}^k\right)^{-1} {L_w^k}^T ,\end{aligned} \tag{4.4.22}$$

where the last inequality holds strictly if $\Delta x \neq 0$ as K_{xx} is assumed to be positive definite and $0 \leq \left|\chi^\dagger\right| < \hat{\chi}$. Similarly, we compute the directional derivative of the lower level merit functions using the formulas from Proposition 4.2 to find

$$X_0 + \partial_\alpha \Phi_L^0 \overset{(4.4.19)}{\leq} -\left|L_w^0\right| \left(\hat{\rho}_0 - |\rho_0|\right) - \hat{\lambda}_0 \pi(B^0) \;\leq\; 0 \tag{4.4.23}$$

as well as

$$X_k + \frac{1}{3} \partial_\alpha \Phi_L^k \overset{(4.4.21)}{\leq} -\left|L_w^k\right| \left(\frac{1}{3}\hat{\rho}_k - \frac{1}{2}\left|\rho_k\right|\right) - \frac{1}{3}\hat{\lambda}_k \pi(B^k) \;\leq\; 0 \,. \tag{4.4.24}$$

as we assume $\hat{\rho}_k > \frac{3}{2}\left|\rho_k\right|$. Both estimates together yield

$$\partial_\alpha \Phi \;\leq\; \sum_{k=1}^n \left(-\frac{2}{3}\hat{\chi}_k \left|L_w^k\right| \hat{\rho}_k + \hat{\chi}_k \left|L_w^k\right| |\rho_k|\right) \;\leq\; 0 \,, \tag{4.4.25}$$

where we use again the assumption $\hat{\rho}_k > \frac{3}{2}\left|\rho_k\right|$. For the case that we have $\partial_\alpha \Phi = 0$ all the above inequalities must be tight. Collecting the corresponding conditions, we find that this can only be the case if we have

$$\Delta x = 0 \,, \;\; \pi(B_j) = 0 \,, \;\; \pi(L^i) = 0 \,, \;\; \left|L_w^j\right| = 0 \tag{4.4.26}$$

$$\Delta w_j = 0 \,, \; \lambda_j^T B^j = 0 \,, \text{ and } \rho_j = 0 \,. \tag{4.4.27}$$

for all $i \in \{1, \ldots, n\}$ and all $j \in \{0, \ldots, n\}$. Thus, we conclude the statement of the Theorem. □

Note that the above Theorem shows that we get either a descent direction of the merit function Φ or $\lambda_j^T B_j = 0$ is implied. This is surprising in the sense that we did not penalize the complementarity condition in the function Φ. Indeed, this observation leads to the following corollary:

Corollary 4.1: *Let us assume that the penalty weights in the merit function Φ are sufficiently large. Then every local solution of the unconstrained optimization problem*

$$\min_{x,w,\lambda} \ \Phi(x, w, \lambda) \tag{4.4.28}$$

at which the regularity Assumptions 4.1, 4.2, and 4.4 are satisfied, is either an infeasible stationary point or a KKT-point of the MPCC (3.4.1)*. Moreover, if there exists a solution $(\hat{x}, \hat{w}, \hat{\lambda})$ of the unconstrained optimization problem* (4.4.28) *at which the regularity Assumptions 4.1, 4.2, and 4.4 hold, then every solution of the MPCC* (3.4.1) *is also a solution of the unconstrained optimization problem* (4.4.28)*, i.e., the merit function Φ is an exact penalty function.*

Proof: Let us assume that we have a solution (x^*, w^*, λ^*) of the unconstrained penalty problem (4.4.28) which is not a KKT point of the MPCC (3.4.1). Provided that (x^*, w^*, λ^*) not an infeasible point, an application of the above sequential convex bilevel programming method is well defined in the sense that a feasible step Δz must exist - independent on how we choose the positive definite matrix K_{xx}. As (x^*, w^*, λ^*) is assumed to be not a KKT point it can easily be seen that we can not possibly satisfy all the conditions (4.4.12), i.e., we get a descent direction of Φ, which is obviously a contradiction to our assumption that (x^*, w^*, λ^*) is a local solution of the unconstrained penalty problem (4.4.28). Thus, every local solution of the unconstrained optimization problem (4.4.28) must either be an infeasible stationary point or a KKT point of the MPCC (3.4.1).

The other way round, let us assume that (x^*, w^*, λ^*) is a solution of the MPCC (3.4.1) achieving the minimum objective value $H_0(x^*, w_0^*)$. If this point is not a solution of the unconstrained optimization problem (4.4.28) and not an infeasible stationary point, then the solution $(\hat{x}, \hat{w}, \hat{\lambda})$ of (4.4.28) satisfies $H_0(\hat{x}, \hat{w}_0) < H_0(x^*, w_0^*)$, i.e., we can use the above argumentation to show that $(\hat{x}, \hat{w}, \hat{\lambda})$ is a feasible KKT point of the MPCC (3.4.1) with a lower objective value than the assumed solution (x^*, w^*, λ^*). This is a contradiction

to the assumption that (x^*, w^*, λ^*) is a solution of the MPCC (3.4.1). Consequently, Φ is an exact penalty function. □

Note that Theorem 4.2 and the corresponding Corollary 4.1 enable us to transfer the traditional argumentation for the globalization of SQP methods [115, 182], i.e., we can require an Armijo-Goldstein condition of the form

$$\widetilde{\Phi}(\alpha) \;\leq\; \widetilde{\Phi}(0) + \epsilon\,\alpha\,\partial_\alpha \widetilde{\Phi}(0) \quad \text{with} \quad \Phi(\alpha) \;:=\; \Phi(x + \alpha\Delta x, w + \alpha\Delta w, \lambda + \alpha\Delta\lambda)$$

to be satisfied with some $\epsilon > 0$, adjusting α via a line search such that a descent of the iterations is guaranteed. Under some additional assumptions, i.e., feasibility of the sub-problems, uniform boundedness of the multipliers χ, ρ, and λ, and the uniform boundedness of K_{xx} and K_{xx}^{-1}, the traditional global convergence statements from the SQP theory transfer [115].

A Stopping Criterion

Note that within an implementation of the proposed method, we need a stopping criterion to decide numerically when convergence is achieved. For this aim, we define the KKT-tolerance ϵ of the sequential convex bilevel programming method analogous to SQP methods as

$$\begin{aligned}
\epsilon_{\mathsf{L}}^j \;&:=\; \Phi_{\mathsf{L}}^j(x, w_j, \lambda_j) \;=\; \left|\frac{\partial L_j(x, w_j, \lambda_j)}{\partial w}\right| \hat{\rho}_j + \hat{\lambda}_j^T\, \pi(B(x, w_j))\;, \\
\epsilon_{\mathsf{U}} \;&:=\; |\partial_\alpha L_0(x, w_0, \lambda_0)| + \sum_{k=1}^{n} \hat{\chi}_k\, \pi_k(L_k(x, w_k, \lambda_k))\;, \\
\text{and} \qquad \epsilon \;&:=\; \epsilon_{\mathsf{U}} + \epsilon_{\mathsf{L}}^0 + \sum_{k=1}^{n} \hat{\chi}_k\, \epsilon_{\mathsf{L}}^k\;. \qquad (4.4.29)
\end{aligned}$$

We can stop the method if $\epsilon <$ TOL is satisfied for a user-specified tolerance TOL, as the above definition of the KKT tolerance ϵ measures the violation of the KKT conditions for optimality. The sequential convex bilevel programming complete algorithm is visualized within Figure 4.1.

Sequential Convex Bilevel Programming:

Initialization:

1) Choose an initial guess z^0 close to the local solution z^* and specify a termination tolerance TOL.
2) Evaluate the functions H_j, B_j, and $L_j := H_j - \lambda_j^T B_j$ together their derivatives at z^0 and store the corresponding nominal values H^j and B^j, the first order sensitivities H_w^j, B_w^j, and L_x^j, as well as the second order terms L_{ww}^j and $L_{x,w}^j$.
3) Choose a positive semi-definite initial Hessian approximation K_{xx} and set $z := z^0$.

Repeat:

4) Solve the min-max QP (4.2.1). As this min-max QP can equivalently be written as a convex QCQP, we can either employ a tailored QCQP solver (e.g. `CVXGEN` [166]) or use a hot-started SQP method which can solve this convex sub-problem reliably.
5) Compute the KKT-tolerance ϵ from equation (4.4.29) and check the stopping criterion $\epsilon <$ TOL. If this stopping criterion is satisfied, we terminate the algorithm with z as the solution.
6) Update the iterate $z^+ = z + \alpha \Delta z$. Here, $\alpha \in (0, 1]$ can for example be determined with the line search procedure from the previous section using the merit function (4.4.2). Another option would be to transfer the concept of trust region algorithms [60] or even filter SQP methods [97]. However, the details of such approaches are beyond the scope of this thesis.
7) Evaluate the functions H_j, B_j, and $L_j := H_j - \lambda_j^T B_j$ together with their derivatives at the new point z^+.
8) Choose a new Hessian approximation K_{xx} at the point z^+.
9) Set $z := z^+$ and continue with step 4).

Figure 4.1: An illustration of the sequential convex bilevel programming method.

The Possible Loss of Superlinear Convergence for Non-Convex Problems and Positive Definite Hessian Approximation

Being at this point, we have discussed the local and global convergence of the method rather independently obtaining consistent results. However, the question which we have not addressed so far is whether we can always satisfy the Dennis-Moré condition for superlinear or quadratic convergence which is needed in Theorem 4.1. This is certainly possible if we work with exact Hessians. For the case that the upper level problems are convex these exact Hessian matrices will be positive definite and we can not encounter problems with convexity of the sub-problems. The question is now whether we can work with bounded and positive semi-definite Hessian approximations K_{xx} even if the exact Hessian

$$\frac{\partial^2}{\partial x^2} L_0(x^*, w_0^*, \lambda_0^*) + \sum_{k=1}^{n} \chi_k^* \frac{\partial^2}{\partial x^2} L_k(x^*, w_k^*, \lambda_k^*) \tag{4.4.30}$$

is indefinite or negative definite still obtaining superlinear convergence. Although this is for standard SQP methods the case [191], this will in general not be possible for our sequential convex bilevel programming method. The corresponding effect has been worked out for sequential linear conic programming methods by Diehl, Jarre, and Vogelbusch in [77]. It was shown that sequential linear conic programming methods with bounded positive definite Hessian cannot converge superlinearly for some non-convex problems.

In the following we will show that there exists an example for which the proposed sequential convex bilevel programming method suffers from this Diehl-Jarre-Vogelbusch effect. For this aim we consider the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & -x_1^2 - (x_2 - 1)^2 \\ \text{s.t.} \quad \max_{w \in \mathbb{R}^2} \quad & 2x^T w - 1 - w^T w \ \leq \ 0 \end{aligned} \tag{4.4.31}$$

Applying the presented sequential convex bilevel programming strategy with the exact Hessian $K_{xx} = -2I_{2\times 2}$, the method converges independent of the starting point in one step to the unique solution $x^* = w^* = (0, -1)^T$.

The closest positive semi-definite approximation of the exact Hessian $-2I_{2\times 2}$ would be given by $K_{xx} = 0$. If we use this approximation the method converges linear with convergence rate $\frac{1}{2}$. Note that this example is completely analogous to the one proposed in [77] and thus the corresponding argumentation transfers.

The Diehl-Jarre-Vogelbusch effect can never cause a problem if the original optimization problem is convex as the exact Hessian is positive semi-definite in this case. However, for general non-convex optimization problems we should be aware of the fact that there exist non-convex cases in which the superlinear convergence is lost if we want to work with positive semi-definite Hessian approximations.

4.5 A Numerical Test Example

In this section, we discuss a numerical application of the sequential convex bilevel programming algorithm which has been developed within the previous sections. For this aim, we consider once more the min-max problem from Example 3.2

$$\min_x \quad \left(x_1 - \frac{1}{2}\right)^2 + x_2^2 \quad \text{s.t.} \quad \begin{cases} 0 \;\geq\; \max\limits_{\|w\|_2 \leq \frac{1}{3}} \; 1 - (x_1 + w_1)^2 - (x_2 + w_2)^2 \\ 0 \;\geq\; \max\limits_{\|w\|_2 \leq \frac{1}{3}} \; \log(x_1 + w_1) - (x_2 + w_2) \\ 0 \;\geq\; \max\limits_{\|w\|_2 \leq \frac{1}{3}} \; -(x_1 + w_1) \end{cases} \tag{4.5.1}$$

Recall that this problem is lower level convex while the upper level problem turns out to be non-convex. In order to start the sequential convex bi-level algorithm, we need a suitable starting point. In our test this starting point is given by $x^0 := \left(\frac{1}{3}, \frac{3}{2}\right)^T$, while the starting points for the lower level maximizers are at

$$w_j^0 := \begin{pmatrix} \frac{1}{3}\,\mathrm{Re}\left(e^{\left(\frac{1}{2}+\frac{2}{3}j\right)i\pi}\right) \\ \frac{1}{3}\,\mathrm{Im}\left(e^{\left(\frac{1}{2}+\frac{2}{3}j\right)i\pi}\right) \end{pmatrix} \quad \text{with} \quad i := \sqrt{-1} \quad \text{and} \quad j \in \{1,2,3\}\,.$$

This initial configuration is shown within Figure 4.2. The center of the circle corresponds to the point x while the lower level maximizers w^j are visualized as "distance keepers", which are expected to converge to the touching points between the constraint functions and the uncertainty circle.

During the iteration, the constraints of the lower level maximization problem are violated, as the ball-constraints are in every step of the method linearized and not exactly imposed. In this example, the sequential convex bilevel programming algorithm converges with full steps, i.e., the globalization routine would in this example in principle not be needed. The following table shows the convergence behavior of the method:

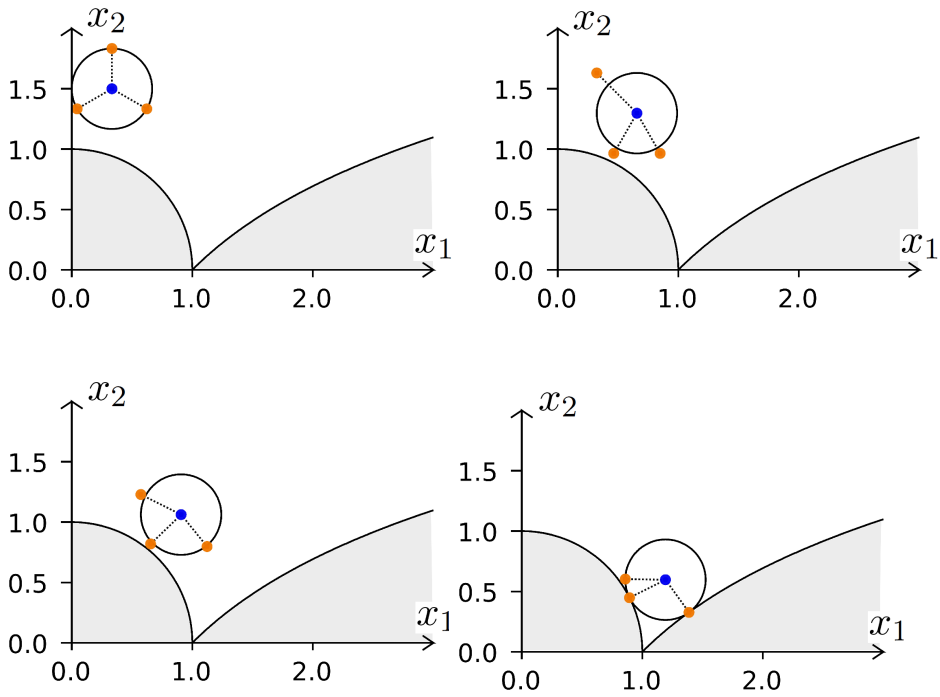


Figure 4.2: The upper left part of the figure visualizes the initialization of the sequential convex bilevel programming method, which is applied to problem (4.5.1). The upper right part and the lower left part of the figure indicate intermediate results after the first and second step of the method, respectively. After the 8-th step of the method convergence is achieved and the corresponding result is visualized within the lower right part of the figure.

Iteration Number	1	2	3	4	5	6	7	8
$-\log_{10}$(KKT-TOL)	0.3	0.5	0.7	1.0	1.5	3.4	7.0	12.1

Here, the evaluation of KKT-tolerance is based on definition (4.4.29). Note that the method converges reliably within 8 iteration and also the quadratic convergence behavior can be observed, as the exact expression for the Hessian K_{xx} has been used.

Finally, we compare the sequential convex bilevel programming algorithm with standard methods applied to the formulation (4.1.1): a standard SQP algorithm with BFGS updates needs 28 iterations to achieve an comparable accuracy while the exact Hessian SQP variant can solve the problem in 15 iterations. Thus, we may state that at least for this problem

the sequential convex bilevel programming algorithm led to a significant improvement, as it exploits the structure of min-max problems in a better way.

Remark 4.5: *In the above case study, we have applied the sequential convex bilevel programming algorithm exactly in the form in which it has been developed within the previous sections. However, due to the particular problem structure several variants are possible. For example, the convex quadratic lower level constraints of the form*

$$w_j \in \mathcal{B} \;=\; \left\{ w \in \mathbb{R}^2 \;\middle|\; \|w\|_2^2 \;\leq\; \frac{1}{9} \right\}$$

could also be kept in the sub-problems of the sequential convex algorithm. In this case we have to solve min-max sub-problems of the form

$$\begin{aligned} \min_{\Delta x} \quad & \max_{w_0+\Delta w_0 \in \mathcal{B}} \; H^0 + H_x^0 \Delta x + \left(\tfrac{1}{2} \Delta w_0^T H_{ww}^0 + \Delta x^T H_{xw}^0 + H_w^0 \right) \Delta w_0 + \tfrac{1}{2} \Delta x^T K_{xx} \Delta x \\ \textit{s.t.} \quad & \max_{w_i+\Delta w_i \in \mathcal{B}} \; H^i + H_x^i \Delta x + \left(\tfrac{1}{2} \Delta w_i^T H_{ww}^i + \Delta x^T H_{xw}^i + H_w^i \right) \Delta w_i \;\leq\; 0 \,, \end{aligned}$$

in order to obtain the steps Δz. Similar to the standard variant of the sequential convex bilevel programming algorithm these min-max sub-problems are convex and can be solved with existing convex solvers, if duality is used to transform them explicitly into an equivalent min-min problem. Numerical testing shows that the corresponding algorithm needs indeed one step less while still achieving a comparable accuracy (but the min-max sub-problems are also slightly more expensive). In this sense this variant can be considered as a reasonable alternative to the sequential linearization of the convex constraints in the lower level maximization problem.

Part II

Robust Optimal Control

Chapter 5

The Propagation of Uncertainty in Dynamic Systems

5.1 Uncertain Nonlinear Dynamic Systems

In this section, we introduce the basic notation and concepts for the discussion of solutions of uncertain linear and nonlinear dynamic systems. We assume that these dynamic systems can be written as

$$\forall \tau \in \mathbb{R}: \quad \dot{x}(\tau) \;=\; f(\,\tau,\, x(\tau),\, w(\tau)\,) \quad \text{with} \quad x(t_1) \;=\; x_1\,. \tag{5.1.1}$$

Here, $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ denotes the possibly nonlinear right-hand side function, $x : \mathbb{R} \to \mathbb{R}^{n_x}$ the state, and $w : \mathbb{R} \to \mathbb{R}^{n_w}$ an uncertain, possibly time-varying input. Throughout this section, we assume that the function f is uniformly Lipschitz continuous with respect to x and piecewise continuous in the other two arguments, such that we can rely on the following well-known result [6]: for any given initial value $x_1 \in \mathbb{R}^{n_x}$ at time $t_1 \in \mathbb{R}$ and for any given (Lebesgue-integrable) input $w : \mathbb{R} \to \mathbb{R}^{n_w}$, we can guarantee unique existence and prolongability of an associated solution x of the nonlinear differential equation (5.1.1).

Now, our assumption on x_1 and w is that they are bounded within a common uncertainty set $\mathcal{W}$, i.e., we only know that $(x_1, w) \in \mathcal{W}$. As a consequence, the solution x of the differential equation (5.1.1) will in general not be unique anymore. In contrast to standard dynamic systems, which typically allow one unique realization of the state, the solution of

an uncertain dynamic system is set valued. Here, we define the set of reachable states $X(t) \subseteq \mathbb{R}^{n_x}$ at any time $t \in [t_1, \infty)$ as follows:

$$X(t) \;:=\; \left\{ x(t) \in \mathbb{R}^{n_x} \;\middle|\; \begin{array}{l} \exists x(\cdot), w(\cdot) : \\ \dot{x}(\tau) \;=\; f(\,\tau,\, x(\tau),\, w(\tau)\,) \\ (x(t_1), w) \in \mathcal{W} \quad \text{for all } \tau \in [t_1, t] \end{array} \right\} . \tag{5.1.2}$$

Intuitively, the set $X(t)$ can be interpreted as the set of all states $x(t)$ which we can obtain by simulating the dynamic system on the time-horizon $[t_1, t]$ testing all possible choices for the initial state and the uncertain input $(x_1, w) \in \mathcal{W}$.

We shall see in the following that the set $X(t)$ is in general difficult to compute. However, for some special cases it is possible to find explicit representations of the set valued function X. Two such special cases are discussed within Examples 5.1 and 5.2.

Example 5.1: Let us consider a scalar but uncertain linear dynamic system of the form

$$\dot{x} = ax + bw \quad \text{with} \quad a, b \in \mathbb{R} .$$

We assume that the corresponding uncertainty set $\mathcal{W}$ is of the form

$$\mathcal{W} \;:=\; \{\, (x_1, w) \;|\; \forall t \in \mathbb{R} : \;\; -1 \;\leq\; w(t) \;\leq\; 1 \,,\; x_1 \;=\; c \}$$

for some $c \in \mathbb{R}$ and $t_1 := 0$. Now, it can easily be checked that the set $X(t)$ of reachable states at time $t \in \mathbb{R}_+$ is an interval. It can be written as

$$X(t) \;=\; \left[\; c\, e^{at} - b \left| 1 - e^{at} \right| \;\;,\;\; c\, e^{at} + b \left| 1 - e^{at} \right| \;\right] .$$

In Figure 5.1 we can find a visualization of this set for $a = -1$, $b = 1$, and $c = \frac{1}{2}$.

Example 5.2: Let us regard the case that the dynamic system is linear, i.e., we have $f(t, x, w) = A(t)x + B(t)w$ for some functions $A : \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ and $B : \mathbb{R} \to \mathbb{R}^{n_x \times n_w}$. Clearly, the state of the dynamic system depends in this case linearly on the uncertainties such that

$$x(t) \;=\; G(t, t_1)x_1 + \int_{t_1}^{t} H_t(\tau) w(\tau)\, \mathrm{d}\tau \;.$$

Here, $H_t(\cdot) := G(t, \cdot)B(\cdot)$ is the impulse response function and $G : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ denotes the fundamental solution, which is defined as:

$$\frac{\partial G(t, \tau)}{\partial t} \;:=\; A(t)G(t, \tau) \quad \text{with} \quad G(\tau, \tau) := I \tag{5.1.3}$$

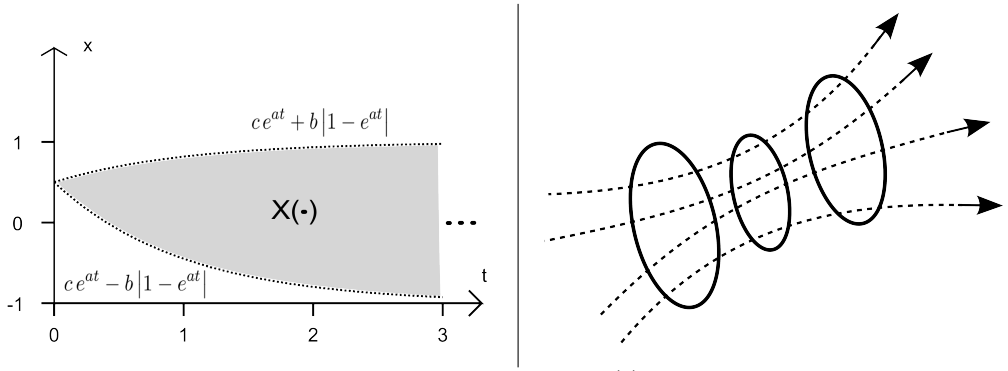


Figure 5.1: Left: A visualization of the solution $X(t)$ of the scalar uncertain dynamic system from Example 5.1. Note that the solution of an uncertain dynamic system is a set valued function (grey shaded area) rather than a single trajectory. Right: For the case that the dynamic system is linear and the uncertainty bounded by an L_2-norm, the reachable sets $X(t)$ are ellipsoids as discussed within Example 5.2. The set valued function $X(\cdot)$ can in this case be imagined as a tube whose cross sections are ellipsoids in which the state trajectories of the uncertain dynamic system can be guaranteed to be.

for all $t, \tau \in \mathbb{R}$. Now, if the set $\mathcal{W}$ is a (non-degenerate) ellipsoid, we may - after suitable scaling of the states and uncertainty - assume that $\mathcal{W}$ can be written as

$$\mathcal{W} \;:=\; \left\{ (x_1, w) \;\middle|\; x_1^2 + \int_{-\infty}^{\infty} \|w(\tau)\|_2^2 \,\mathrm{d}\tau \;\leq\; 1 \right\} .$$

In order to compute the set $X(t)$ for a given time t, we first compute its support function

$$V(c) \;:=\; \max_{x(t) \in X(t)} \; c^T x(t) \;=\; \sqrt{c^T \left(G(t,t_1)G(t,t_1)^T + \int_{t_1}^{t} H_t(\tau) H_t(\tau)^T \,\mathrm{d}\tau \right) c} \,.$$

In this context, we observe that the matrix

$$Q(t) \;:=\; G(t,t_1)G(t,t_1)^T + \int_{t_1}^{t} H_t(\tau) H_t(\tau)^T \,\mathrm{d}\tau$$

can also directly be obtained by solving a Lyapunov differential equation of the form

$$\forall \tau \in [t_1, t]: \quad \dot{Q}(\tau) \;=\; A(\tau)Q(\tau) + Q(\tau)A(\tau)^T + B(\tau)B(\tau)^T \quad \text{with} \quad Q(t_1) = 1 \,.$$

As the convex set $X(t)$ is uniquely characterized by its support function (cf. Corollary 2.11 from Chapter 2), we must have $X(t) \;=\; \mathcal{E}(\, Q(t)\,)$, i.e., the set of reachable states is at each time $t \geq t_1$ an ellipsoid. This observation has for example been used in [124, 126, 160, 173].

Unfortunately, it is difficult to make statements about the geometry of the set $X(t)$ in the general case. In Examples 5.1 and 5.2, we have seen that the set $X(t)$ turns out to be an interval or an ellipsoid, but both examples were based on the assumption that the dynamic system is linear. However, we have to be aware of the fact that even if the dynamic system is linear, the question of how to compute the set $X(t)$ can be non-trivial depending on our assumptions on the uncertainty set.

Definition of the Set-Propagation Operator

The aim of the following consideration is to formalize the construction of reachable sets of nonlinear dynamic systems. For this theoretical purpose, we first introduce a very basic assumption on the uncertainty set $\mathcal{W}$:

Assumption 5.1: *We assume that the uncertainty set $\mathcal{W}$ can be written in the following uncorrelated form*

$$\mathcal{W} \;=\; \{\; (x_1, w) \;\mid\; x_1 \in X_1 \quad \textit{and} \quad w(\tau) \in W(\tau) \quad \textit{for all } \tau \in \mathbb{R} \;\} \;.$$

Here, the sets $X_1 \subseteq \mathbb{R}^{n_x}$ and $W(\tau) \subseteq \mathbb{R}^{n_w}$ are assumed to be given for all times $\tau \in \mathbb{R}$.

Note that the above assumption is restricting as it excludes for example the case that the uncertainty set contains L_2-bounded inputs recalling our considerations from Example 5.2. Here, the L_2-bounded uncertainties are very interesting in the sense that we can for uncertain linear systems compute the reachable sets exactly by solving a simple Lyapunov differential equation. Nevertheless, we might argue that Assumption 5.1 is not too restrictive for most of the practical applications, as we are often able to formulate the dynamic system in such a way that the uncertainties do not correlate in time. In particular, the case that we have an uncertain time constant parameter can be re-formulated by introducing additional state variables - this will be discussed in more detail in Section 6.1.

Once we accept Assumption 5.1, it is easier to investigate how the uncertainty propagates through the dynamic evolution leading to a consistent mathematical notation which we will later employ to formulate robust optimal control problems. Here, we follow the classical concept of robust positive invariant tubes [35, 36, 195] as well as the theory on set valued (or multi-valued) differential equations [65].

Let us start with the definition of the set $\mathcal{F}(t_1, t_2)$ of feasible state and uncertainty realizations on the interval $[t_1, t_2] \subseteq \mathbb{R}$, which we define as

$$\mathcal{F}(t_1,t_2) \;:=\; \left\{ (x,w) \;\middle|\; \begin{array}{rcl} \dot{x}(\tau) & = & f(\,\tau,\, x(\tau),\, w(\tau)\,) \\ w(\tau) & \in & W(\tau) \quad \text{for all } \tau \in [t_1,t_2] \end{array} \right\} .$$

Now, we introduce the set propagation operator $T(t_2, t_1) : \Pi(\mathbb{R}^{n_x}) \to \Pi(\mathbb{R}^{n_x})$, which is associated with the uncertain differential equation. It is defined for all sets $X_1 \subseteq \mathbb{R}^{n_x}$ and for all $t_1, t_2 \in \mathbb{R}$ with $t_1 \leq t_2$:

$$T(t_2,t_1)[X_1] \;:=\; \{\, y \in \mathbb{R}^{n_x} \,|\, \exists (x,w) \in \mathcal{F}(t_1,t_2) : \;\; x(t_1) \in X_1 \;\text{ and }\; x(t_2) \,=\, y \,\} \;.$$

In this context, we use the set notation $\Pi(Y) := \{\, X \mid X \subseteq Y \,\}$ to denote a power set, i.e., the set of all subsets (including the empty set), of any given set Y. Note that the rather abstract set notation of the form

$$X_2 \,=\, T(t_2,t_1)[X_1]$$

is nothing but a concise way to say that X_2 is the reachable set of the dynamic system at time t_2 assuming that the initial value of the differential equation at time t_1 is known to be in the set X_1, while the uncertain input w satisfies $w(\tau) \in W(\tau)$ for all times $\tau \in [t_1, t_2]$.

Associativity of the Set-Propagation Operator

In the following, we will denote the set of all propagation operators, which are associated with the differential equation f, with the symbol $\mathcal{T}$. Now, if we regard two consecutive set propagation operators $T(t_3, t_2)$ and $T(t_2, t_1)$ for some $t_1, t_2, t_3 \in \mathbb{R}$ with $t_1 \leq t_2 \leq t_3$, their composition $\circ : \mathcal{T} \times \mathcal{T} \to \mathcal{T}$ can be defined by

$$T(t_3,t_2) \circ T(t_2,t_1) \;:=\; T(t_3,t_1) \;.$$

This composition satisfies the following fundamental property:

Proposition 5.1 (Associativity): *The pair $(\mathcal{T}, \circ)$ satisfies for any three consecutive operators $T(t_1, t_2), T(t_2, t_3), T(t_3, t_4) \in \mathcal{T}$ with $t_1 \leq t_2 \leq t_3 \leq t_4$ the associativity relation*

$$(\,T(t_4,t_3) \circ T(t_3,t_2)\,) \circ T(t_2,t_1) \;=\; T(t_4,t_3) \circ (\,T(t_3,t_2) \circ T(t_2,t_1)\,) \;.$$

In particular, for the case that the dynamic system is autonomous, the pair $(\mathcal{T}, \circ)$ can be interpreted as a semi-group, as summarized in the following remark.

Remark 5.1 (Set-Propagation Semi-Group): *If the right-hand side function f does not explicitly depend on the time τ while the set $W(\tau) = W$ is also autonomous, the operator $T(t_2, t_1)$ can be notated as $T(t_2 - t_1) \equiv T(t_2, t_1)$, i.e., the propagation depends on the difference $t_2 - t_1$ only. Thus, the definition of the composition takes the form of a homomorphism*

$$\forall \Delta t_1, \Delta t_2 \in \mathbb{R}_+ : \quad T(\Delta t_1) \circ T(\Delta t_2) \; = \; T(\Delta t_1 + \Delta t_2) \; .$$

I.e., $(\mathcal{T}, \circ) \cong (\mathbb{R}_+, +)$ turns out to be a semi-group, which is isomorphic to the additive semi-group of non-negative real numbers.

Note that the operators of the form $T(\tau, t_1)$ generate for every given set $X_1 \subseteq \mathbb{R}^{n_x}$ an associated orbit $X : [t_1, t_2] \to \Pi(\mathbb{R}^{n_x})$ which is on the interval $[t_1, t_2]$ defined as

$$\forall \tau \in [t_1, t_2] : \quad X(\tau) \; := \; T(\tau, t_1)[X_1] \; .$$

Due to the associativity of the pair $(\mathcal{T}, \circ)$ we can also construct the orbit X formally via a sequence of the form

$$X(\tau + \mathrm{d}\tau) \; = \; T(\tau + \mathrm{d}\tau, \tau)[X(\tau)] \; ,$$

which is started at $X(t_1) = X_1$ and which propagates with infinitesimal step-sizes $\mathrm{d}\tau$. Recall that this forward construction of the reachable set is possible due to Assumption 5.1, i.e., due to the fact that the uncertainty does not correlate in time. In order to express this infinitesimal forward generation within an intuitive notation, we employ the following formal definition:

Definition 5.1 (Infinitesimal Set-Generation): *We say that a set-valued function of the form $X : [t_1, t_2] \to \Pi(\mathbb{R}^{n_x})$ satisfies a formal differential equation of the form*

$$\forall \tau \in [t_1, t_2] : \quad X(\tau^+) \; = \; F(\tau, X(\tau), W(\tau)) \; ,$$

if and only if we have $X(\tau) \; := \; T(\tau, t_1)[X(t_1)]$ for all $\tau \in [t_1, t_2]$. Here, F is the infinitesimal generator of the pair $(\mathcal{T}, \circ)$, which can formally be written as

$$F(\tau, X(\tau), W(\tau)) \; := \; T(\tau + \mathrm{d}\tau, \tau)[X(\tau)] \; .$$

Our notation of the reachable set depends in the following considerations always on the context. The notation in form of the above formal differential equation has the advantage that it highlights the similarity between the propagation of a vector valued state x through a deterministic differential equation and the propagation of a set-valued state X through an uncertain differential equation. This notation is especially intuitive if we want to get clear about dependencies. For example, we might have a differential equation of the form

$$\forall \tau \in [t_1, t_2]: \quad \dot{x}(\tau) \;=\; f(\tau, u(\tau), x(\tau), w(\tau)) \;,$$

which depends additionally on a control input u. In this case, we will write the associated set-valued differential equation in the form

$$\forall \tau \in [t_1, t_2]: \quad X(\tau^+) \;=\; F(\tau, u(\tau), X(\tau), W(\tau))$$

in order to make clear that the propagation of the associated reachable sets X can be influenced by the control input u. It might also help to be aware of the fact that the above notation trivially transfers to discrete-time systems, for which the set propagation has the form $X_{k+1} \;=\; F(k, u_k, X_k, W_k)$. In this and the following chapter our focus is on continuous-time systems, but it is a general remark that most of the considerations transfer one-to-one to discrete-time systems as well.

Monotonicity of the Set-Propagation Operator

Besides the above fundamental associativity property, we also observe that the operator $T(t_2, t_1)$ satisfies a monotonicity relation which can be stated as follows:

Proposition 5.2 (Monotonicity): *Let $X \subseteq Y \subseteq \mathbb{R}^{n_x}$ be two sets, one contained in the other. Then we have an inclusion of the form*

$$T(t_2, t_1)[X] \quad \subseteq \quad T(t_2, t_1)[Y] \;.$$

This result holds for all $t_1, t_2 \in \mathbb{R}$ with $t_1 \leq t_2$.

Motivated by this monotonicity relation, we introduce the following notation of robust positive invariant tubes:

Definition 5.2 (Robust Positive Invariant Tubes): *A set-valued function of the form $\mathbb{X} : [t_1, t_2] \to \Pi(\mathbb{R}^{n_x})$ is called a robust positive invariant tube on the interval $[t_1, t_2]$, if the inclusion*

$$\mathbb{X}(t') \;\supseteq\; T(t', t)[\mathbb{X}(t)]$$

is satisfied for all $t, t' \in [t_1, t_2]$ with $t' \geq t$. This condition can alternatively also be written in form of a relaxed differential equation of the form

$$\forall \tau \in [t_1, t_2]: \quad \mathbb{X}(\tau^+) \supseteq F(\tau, \mathbb{X}(\tau), W(\tau)) ,$$

if we transfer our formal notation of infinitesimal set generation.

Note that if a function $\mathbb{X} : [t_1, t_2] \to \mathbb{R}^{n_x}$ is a robust positive invariant tube, this implies that once we know that the current state $x(t)$ is at a given time $t \in [t_1, t_2]$ inside this tube, i.e., $x(t) \in \mathbb{X}(t)$, we also know that it will be inside this tube for all future times $t' \in [t, t_2]$, i.e., $x(t') \in \mathbb{X}(t')$ - no matter how the uncertainty w is realized. In other words, if we find a computationally tractable way to generate robust positive invariant tubes, we also have a way to generate outer approximations of reachable sets.

5.2 Robust Positive Invariant Tubes for Linear Dynamic Systems

The main difficulty with uncertain dynamic systems of the form (5.1.1) is that we are not interested in a single vector-valued solution x for one particular realization of the uncertainties, but in a set-valued function X, as introduced in equation (5.1.2). An accurate numerical computation of the reachable set $X(t)$ is in principle possible by discretizing the function X in space and time and searching for a suitable numerical realization of the generating operator $T(t + \mathrm{d}t, t)$ for a small but positive step size $\mathrm{d}t$. Such a technique has for example been suggested in [170] where the time-dependent reachable sets are represented via level set functions which satisfy a Hamilton-Jacobi-Isaacs equation. Here discretization techniques inspired from the field of partial differential equations can be exploited. However, we should be aware of the fact that such techniques, which are exact up to a small numerical error, will be limited to small state dimensions n_x. For larger state dimensions, we encounter the natural tradeoff between accuracy and tractability, which appears in similar versions throughout almost all fields of robust optimization.

As we have already outlined in the previous section, the notation of robust positive invariant tubes is a very powerful concept to deal with outer approximations of reachable sets. The aim of this section is to outline computationally tractable techniques in order to compute parameterized robust positive invariant tubes, which shall later be employed for solving robust optimal control problems in a conservative approximation.

In order to develop these computational techniques, we have to specialize our assumptions on the uncertainty set. Based on the notation from Assumption 5.1, we propose to model the set $W(\tau)$ for the uncertain input $w(\tau)$ at any time $\tau \in \mathbb{R}$ as follows:

Assumption 5.2: *Let us employ the notation*

$$\Delta^n \ := \ \left\{ \ \lambda \in \mathbb{R}^n_{++} \ \middle| \ \sum_{i=1}^{n} \lambda_i \ = \ 1 \ \right\}$$

to denote the half-open unit simplex as introduced within Chapter 2. We assume that the uncertainty set $W(\tau) \subseteq \mathbb{R}^{n_w}$ has for all $\tau \in \mathbb{R}$ the form

$$W(\tau) \ := \ \{ \, w \in \mathbb{R}^{n_w} \mid \forall \, \lambda \in \Delta^n : \ w \ \in \ \mathcal{E}(\, \Omega_\tau(\lambda) \,) \ \} \tag{5.2.1}$$

Here, $\Omega_\tau : \mathbb{R}^n_{++} \to \mathbb{S}^{n_w}_+$ is assumed to be an anti-homogeneous matrix valued map, i.e., we assume that we have for every $\alpha > 0$ and every $\lambda \in \mathbb{R}^n_{++}$ the relation $\Omega_\tau(\alpha\lambda) \ = \ \frac{1}{\alpha}\Omega_\tau(\lambda)$.

Note that the above modeling assumption is sufficient to approximate all compact, convex, and point-symmetric uncertainty sets $W(\tau)$. Thus, the assumption that $W(\tau)$ has these three properties is not a main restriction for most practical applications. In order to illustrate how we can work with Assumption 5.2, we regard Examples 5.3, 5.4, and 5.5, where possible choices for the function Ω_τ are discussed.

Example 5.3: If we employ the choice $\Omega_\tau(\lambda) := \frac{1}{\lambda} I$ and $n = 1$ in equation (5.2.1), we obtain the set

$$W(\tau) \ = \ \{ \, w \ | \ w(\tau)^T w(\tau) \ \leq \ 1 \, \} \, .$$

More generally, the choice $\Omega_\tau(\lambda) \ := \ \frac{1}{\lambda}\Sigma(\tau) \in \mathbb{S}^{n_w}_+$ allows us to require that $w(\tau)$ is at every time τ bounded by an ellipsoidal uncertainty set of the form $\mathcal{E}(\Sigma(\tau))$.

Example 5.4: Having Theorem 2.4 from Chapter 2 in mind, we can employ an anti-homogeneous function of the form

$$\Omega_\tau(\lambda) \ := \ \sum_{i=1}^{n} \frac{1}{\lambda_i} \Sigma_i$$

for some positive semi-definite matrices $\Sigma_i \in \mathbb{S}^{n_w}_+$. This allows us to model the case that $w(\tau)$ is at each time τ known to be in the set $\sum_{i=1}^n \mathcal{E}(\Sigma_i)$, which is a finite sum of ellipsoids. In particular, we can model the set

$$W(\tau) \ = \ \{ \, w \ | \ \| \, w(\tau) \, \|_\infty \ \leq \ 1 \, \}$$

by choosing $\Omega_\tau(\lambda) := \mathrm{diag}(\lambda)^{-1}$ and $n = n_w$, as the unit cube in $\mathbb{R}^{n_w}$ is a sum of n_w degenerate ellipsoids (c.f. Example 2.14 from Chapter 2). In other words, the above assumption allows us to formulate simple component-wise bounds on the uncertainty. More generally, if the uncertainty set is a zonotope, it can be modeled with Assumption 5.2.

Example 5.5: Note that we can employ the anti-homogeneous choice

$$\Omega_\tau(\lambda) := \left(\sum_{i=1}^{n} \lambda_i \Sigma_i^{-1} \right)^{-1}$$

for some positive definite matrices $\Sigma_i \in \mathbb{S}_{++}^{n_w}$. This allows us to model the case that $w(\tau)$ is at each time τ known to be in the intersection of n centered ellipsoids $\mathcal{E}(\Sigma_i)$. This can easily be shown by using the Theorem 2.2 from Chapter 2. As mentioned above, we can in principle approximate every compact, convex, and point-symmetric set by an intersection of ellipsoids - in this sense, we may regard Assumption 5.2 as a quite powerful modeling tool.

In the following consideration, we plan to employ Assumption 5.2 as a basis for the construction of parameterized robust positive invariant tubes. Here, we first concentrate on constructive methods for linear dynamic systems, which will later be generalized for nonlinear dynamics as well.

Ellipsoidal Methods for Linear Dynamic Systems

For the case that the right-hand side function f is linear, we can write the uncertain dynamic system in the form

$$\forall \tau \in \mathbb{R}: \qquad \dot{x}(\tau) = A(\tau)x(\tau) + B(\tau)w(\tau) \,, \tag{5.2.2}$$

Here, $A : \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ and $B : \mathbb{R} \to \mathbb{R}^{n_x \times n_w}$ are assumed to be L_1-integrable functions. Recall that the fundamental solution $G : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ of this linear system is given by

$$\forall t, \tau \in \mathbb{R}: \quad \frac{\partial G(t,\tau)}{\partial t} := A(t)G(t,\tau) \quad \text{with} \quad G(\tau,\tau) := I \,. \tag{5.2.3}$$

Our consideration will be based on Assumption 5.2, which specifies the uncertainty constraints for the input w. Recall that this assumption implies that the uncertainty

sets $W(\tau)$ are compact, convex, and point-symmetric with respect to the origin. One of the main advantages of the analysis of linear dynamic systems is that we can formulate conservation laws for these three properties, which are summarized within the following proposition:

Proposition 5.3 (Conservation Laws of Linear Uncertainty Propagation): *Let us assume that the functions A and B are L_1-integrable and let $X(t) := T(t, t_1)[X_1]$ be the reachable set at some time $t \geq t_1$, which is associated with some given initial uncertainty set $X_1 \subseteq \mathbb{R}^{n_x}$. Then the following conservation laws hold:*

1. *If the sets X_1 and $W(\tau)$ are for all $\tau \in [t_1, t]$ compact, then $X(t)$ is compact.*
2. *If the sets X_1 and $W(\tau)$ are for all $\tau \in [t_1, t]$ convex, then $X(t)$ is convex.*
3. *If the sets X_1 and $W(\tau)$ are for all $\tau \in [t_1, t]$ point-symmetric with respect to the origin, then $X(t)$ is point-symmetric with respect to the origin.*

In summary, if the sets $W(\tau)$ are compact, convex, and point-symmetric, then the operator $T(t, t_1)$ preserves compactness, convexity, and point-symmetry.

Proof: For the proof of the conservation of compactness, we have to use the assumption that the functions A and B are L_1-integrable. As the required techniques for the proof are rather technical and not the focus of this thesis, we refer at this point to [218], where a mathematical proof of the statement can be found. Concerning the second statement, we know that $X(t)$ is convex if the uncertainty sets X_1 and $W(\tau)$ are all convex, as a linear transformation preserves convexity. A similar argumentation can be applied for the last statement, as a linear map does not only preserve convexity but also point-symmetry. □

Note that if we employ Assumption 5.2, all the conservation laws for the operator $T(t, t_1)$ apply. This implies in particular, that we can apply the convex optimization techniques from Chapter 2, i.e., the set $X(t) := T(t, t_1)[X_1]$ can under the mentioned assumptions uniquely be characterized via its support function. For the following technical consideration, we assume for a moment that the set $X_1 := \mathcal{E}(Q_1)$ is a given ellipsoid with $Q_1 \in \mathbb{S}_+^{n_x}$, i.e., our knowledge about the initial state $x(t_1)$ is of the form $x(t_1) \in X_1$. However, this assumption is only introduced for some intermediate simplifications, but will later, in the "initial value free" formulation of the main result (see Theorem 5.1), not be needed anymore.

In order to construct the support function of the set $X(t)$, we have to compute the maximum excitation of the linear dynamic system

$$V(t,c) \;:=\; \max_{\xi \in X(t)} \; c^T \xi$$

at some time $t \geq t_1$ in a given direction $c \in \mathbb{R}^{n_x}$. Here, we may also represent $V(t,c)$ in a more expanded form, which is given by

$$V(t,c) \;=\; \max_{x(\cdot),w(\cdot)} \quad c^T x(t) \quad \text{s.t.} \quad \left\{ \begin{array}{l} \text{for all } \tau \in [t_1,t] : \\ \quad \dot{x}(\tau) \;=\; A(\tau)x(\tau) + B(\tau)w(\tau) \\ \quad x(t_1) \;\in\; X_1 \\ \quad w(\tau) \;\in\; W(\tau) \;\; . \end{array} \right. \tag{5.2.4}$$

The above maximization problem can be regarded as an infinite dimensional convex optimization problem. Note that the maximum exists, as $X(t)$ is compact. We assume for simplicity that there is a feasible point such that we can compute V via a minimization problem by passing to the dual problem. For this aim, we first express the state function x of the linear dynamic system explicitly as

$$\forall t \geq t_1 : \quad x(t) \;=\; G(t,t_1)x(t_1) \,+\, \int_{t_1}^{t} H_t(\tau)w(\tau)\,\mathrm{d}\tau \;\; , \tag{5.2.5}$$

recalling that $H_t(\cdot) := G(t,\cdot)B(\cdot)$ denotes the impulse response function and G the fundamental solution, as defined in (5.2.3). Now, the dual of the problem defining the function V can be written as

$$V(t,c) \;=\; \inf_{\lambda(\cdot)} \; \inf_{\substack{\mu>0 \\ \nu(\cdot)>0}} \; \frac{1}{\mu} Q_1 \,+\, \int_{t_1}^{t} \frac{c^T H_t(\tau)\Omega_\tau(\lambda(\tau))H_t(\tau)^T c}{4\,\nu(\tau)}\,\mathrm{d}\tau \,+\, \mu \,+\, \int_{t_1}^{t} \nu(\tau)\,\mathrm{d}\tau$$

$$\text{s.t.} \quad \lambda(\tau) \in \Delta^n \quad \text{for all } \tau \in [t_1,t] \;,$$

where the time-varying multiplier $\nu : \mathbb{R} \to \mathbb{R}_+$ has been introduced to account for the parameterized constraints on the uncertainty which have the form $w(\tau) \in \mathcal{E}(\,\Omega_\tau(\lambda)\,)$ while the scalar multiplier $\mu \in \mathbb{R}_+$ takes care of the initial value inclusion $x(t_1) \in \mathcal{E}(\,Q_1\,)$. Finally, we use the assumption that the function Ω_τ is anti-homogeneous, i.e., we have

$$\frac{1}{\nu(\tau)}\Omega_\tau(\lambda(\tau)) \;=\; \Omega_\tau(\nu(\tau)\lambda(\tau))$$

such that we can rescale both the function λ and the multiplier μ writing the support function V in the form

$$\begin{aligned} V(t,c) = & \inf_{Q(t),\mu>0,\lambda(\cdot)>0} \sqrt{c^T Q(t)\, c} \quad \text{s.t.} \\ & Q(t) = \left(\tfrac{1}{\mu} Q_1 + \int_{t_1}^{t} H_t(\tau)\Omega_\tau(\lambda(\tau))H_t(\tau)^T \mathrm{d}\tau\right)\left(\mu + \int_{t_1}^{t} \textstyle\sum_{i=0}^{n} \lambda_i(\tau)\mathrm{d}\tau\right). \end{aligned} \tag{5.2.6}$$

As the above equation for the support function $V(t,c)$ of the closed and convex set $X(t)$ holds for all directions c, we obtain an ellipsoidal outer approximation (compare with Example 2.11 from Chapter 2):

Lemma 5.1: *Let Assumption 5.2 for the input uncertainty set be satisfied and let $\mu > 0$ be a given positive constant and $\lambda : [t_1, t] \to \mathbb{R}_{++}$ a function such that the integrals in the definitions*

$$P(t) \;:=\; \frac{1}{\mu} Q_1 + \int_{t_1}^{t} H_t(\tau)\Omega_\tau(\lambda(\tau))H_t(\tau)^T \,\mathrm{d}\tau \quad \textit{and} \quad r(t) \;:=\; \mu + \int_{t_1}^{t} \sum_{i=0}^{n} \lambda_i(\tau)\mathrm{d}\tau$$

exist. Then we have an inclusion of the form

$$X(t) \;=\; T(t,t_1)[\,\mathcal{E}(\,Q_1\,)\,] \;\subseteq\; \mathcal{E}(\,Q(t)\,)\,,$$

i.e., the set of reachable states is contained in an ellipsoid of the form $\mathcal{E}(\,Q(t)\,)$, where we use the notation $Q(t) \;:=\; r(t)P(t)$.

Note that the above Lemma can be regarded as a useful tool which can be employed to generate ellipsoidal outer approximations numerically. Here, the matrix $P(t)$ can also be obtained by a forward simulation of a Lyapunov differential equation of the form

$$\begin{aligned} \forall \tau \in [t_1, t]: \quad \dot{P}(\tau) &= A(\tau)P(\tau) + P(\tau)A(\tau)^T + B(\tau)\Omega_\tau(\lambda(\tau))B(\tau)^T \\ \text{with} \qquad P(t_1) &= \tfrac{1}{\mu} Q_1\,. \end{aligned} \tag{5.2.7}$$

Note that Lyapunov differential equations are well-known since a long time [163, 34]. The fact that the function

$$P(\tau) \;=\; \frac{1}{\mu} Q_1 + \int_{t_1}^{\tau} H_t(\tau)\Omega_\tau(\lambda(\tau))H_t(\tau)^T \,\mathrm{d}\tau$$

satisfies the differential equation (5.2.7) can simply be checked by using the definition of H_t together with the differential equation (5.2.3).

In the next step, we discuss an alternative way to formulate Lemma 5.1. The aim of this alternative formulation is to avoid an explicit specification of the set X_1 of uncertain initial states such that the associativity of the propagation operators in the set $\mathcal{T}$ with respect to composition is reflected in the formulation of the approximation strategy:

Theorem 5.1: *Let Assumption 5.2 for the input uncertainty set be satisfied and let $Q : [t_1, t_2] \to \mathbb{S}^{n_x}_+$ and $\kappa : [t_1, t_2] \to \mathbb{R}^n_{++}$ be any functions which satisfy for all $\tau \in [t_1, t_2]$ a differential equation of the form*

$$\dot{Q}(\tau) \;=\; A(\tau)Q(\tau) + Q(\tau)A(\tau)^T + \sum_{i=1}^{n} \kappa_i(\tau) Q(\tau) + B(\tau)\Omega_\tau(\kappa(\tau))B(\tau)^T \;.$$

Then the function $\mathbb{X}(\cdot) := \mathcal{E}(\, Q(\cdot)\,)$ is a robust positive invariant tube on the given time interval $[t_1, t_2]$.

Proof: The main idea of the proof is to directly derive a differential equation for the function $Q(t) := r(t)P(t)$ for $t \in [t_1, t_2]$. For this aim, we employ the chain rule:

$$\begin{aligned}
\dot{Q}(\tau) \;&=\; \dot{r}(\tau)P(\tau) + r(\tau)\dot{P}(\tau) \\
&=\; \left[\sum_{i=1}^{n} \lambda_i(\tau)\right] P(\tau) + A(\tau)Q(\tau) + Q(\tau)A(\tau)^T + r(\tau)B(\tau)\Omega_\tau(\lambda(\tau))B(\tau)^T \\
&=\; \frac{\sum_{i=1}^{n} \lambda_i(\tau)}{r(\tau)} Q(\tau) + A(\tau)Q(\tau) + Q(\tau)A(\tau)^T + B(\tau)\Omega_\tau\left(\frac{\lambda(\tau)}{r(\tau)}\right) B(\tau)^T \;.
\end{aligned}$$

In the last step we have used that the function r is for all $\tau \in [t_1, t]$ strictly positive:

$$r(\tau) \;=\; \mu + \int_{t_1}^{\tau} \sum_{i=1}^{n} \lambda_i(\tau')\, \mathrm{d}\tau' \;\geq\; \mu \;>\; 0 \;.$$

The statement of the Theorem follows now by introducing the new re-scaled multiplier function $\kappa(\tau) := \frac{\lambda(\tau)}{r(\tau)}$, which is well-defined for all $\tau \in [t_1, t]$. □

Remark 5.2: *Theorem 5.1 as well as Theorem 5.2 have been proposed in similar versions in the work of Kurzhanski and Varaiya [146, 144], who were among the pioneers of ellipsoidal methods for linear dynamic systems. In this context, we also refer to the work of Schweppe and Glover [102, 209] as well as Brockman and Corless [47]. In this thesis, we present these existing ellipsoidal techniques for linear dynamic systems in a uniform framework which is based on Assumption 5.2.*

Remark 5.3: *Note that the new function κ is in the proof of Theorem 5.1 constructed from the function λ via the re-scaling relation*

$$\kappa(\tau) \;:=\; \frac{\lambda(\tau)}{r(\tau)} \;=\; \frac{\lambda(\tau)}{\lambda^0 + \int_0^\tau \sum_{i=1}^n \lambda_i(\tau')\,\mathrm{d}\tau'}\,.$$

For a fixed $\lambda^0 > 0$, this definition can be interpreted as a bijective change of variables. Here, the inverse construction of the function λ from the function κ is given by

$$\lambda(\tau) \;=\; \lambda^0 \kappa(\tau)\,\exp\left(\int_0^\tau \sum_{i=1}^n \kappa_i(\tau')\,\mathrm{d}\tau'\right)$$

for all $\tau \in [t_1, t_2]$. Clearly, we cannot recover the initial constant λ^0 together with the function λ as the original parameterization in the λ-variables was scaling invariant, while the parameterization in κ avoids this type of scaling indefiniteness.

Note that the differential equation for the matrix valued function Q is a Lyapunov differential equation for any given function κ. Thus, if we choose an initial value $Q(t_1)$ at time t_1, the solution $Q(t_2) \;=\; \Gamma(t_2, t_1)[Q(t_1)]$ of the differential equation at time t_2 depends affinely on $Q(t_1)$, i.e., the associated propagation operator $\Gamma(t_2, t_1)$ is affine. Moreover, we can define a composition of the form

$$\Gamma(t_3, t_2) \circ \Gamma(t_2, t_1) \;:=\; \Gamma(t_3, t_1)$$

for $t_1 \leq t_2 \leq t_3$. This implies that the propagation operators which generate the parameterized robust positive invariant ellipsoidal tube $\mathcal{E}(Q(\cdot))$ are associative with respect to composition.

On the one hand, Theorem 5.1 provides only one way to construct robust positive invariant tubes. However, on the other hand, due to the fact that the construction of the ellipsoidal outer approximation is parameterized in the function κ we might nevertheless obtain sufficient flexibility to adapt to the particular situation, we are optimizing for. In order to illustrate this aspect, we highlight once more that we can always achieve that the ellipsoidal approximation is exact in a given direction:

Theorem 5.2: *Let $c \in \mathbb{R}^{n_x} \setminus \{0\}$ be any given direction. Then the support function $V(t, c)$, which is defined in* (5.2.4)*, can for all $t \in [t_1, \infty)$ be written as*

$$\begin{aligned} V(t,c) = \inf_{Q(\cdot),\kappa(\cdot)>0} & \sqrt{c^T Q(t) c} \quad \text{s.t.} \\ \dot{Q}(\tau) \;&=\; A(\tau)Q(\tau) + Q(\tau)A(\tau)^T + \sum_{i=1}^n \kappa_i(\tau) Q(\tau) + B(\tau)\Omega_\tau(\kappa(\tau))B(\tau)^T \\ Q(t_1) \;&=\; Q_1 \end{aligned} \tag{5.2.8}$$

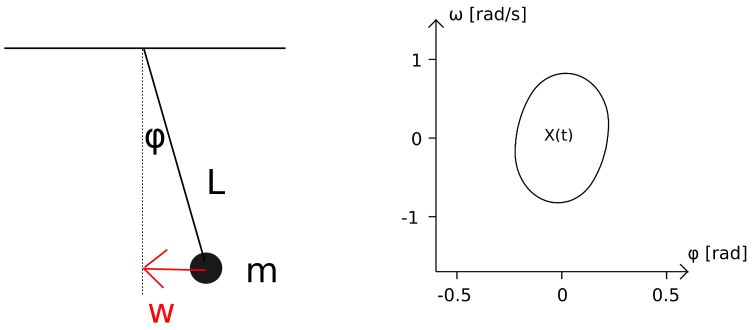


Figure 5.2: Left: a sketch of the pendulum. Right: a visualization of the set $X(t)$ of reachable states at the time $t = 1.2\,\mathrm{s}$ under the assumption that the uncertain force w satisfies $|w(\tau)| \leq 1\,\mathrm{N}$ for all $\tau \in [0,t]$. Note that this 2-dimensional set has been computed for visualization purposes. In order to obtain such a visualization, we computed the associated support function $V(t,c)$ for many directions $c \in \mathbb{R}^2$ with $\|c\|_2 = 1$ such that a sufficient resolution is obtained.

In other words, the ellipsoid $\mathcal{E}(\,Q(t)\,)$, given by the solution of the above optimal control problem, contains the set $X(t)$ of reachable states and touches it in the desired directions c and $-c$.

Proof: The statement of the theorem uses the fact that equation (5.2.6) holds with equality. Note that the optimization problems (5.2.6) and (5.2.8) are equivalent as a change of variables does not affect the objective value. □

In order to illustrate and visualize, how the above Theorem 5.2 can be used in practice, we consider Example 5.6.

Example 5.6: We regard the case that the matrix valued functions A and B are for all $\tau \in [0,t]$ explicitly given by

$$A(\tau) := \begin{pmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{pmatrix}, \quad \text{and} \quad B(\tau) := \begin{pmatrix} 0 \\ -\frac{1}{mL} \end{pmatrix}.$$

The corresponding dynamic system can be interpreted as the linearized dynamics of a pendulum with length L and mass m, where the state $x(t) := (\varphi(t), \omega(t))^T$ represents

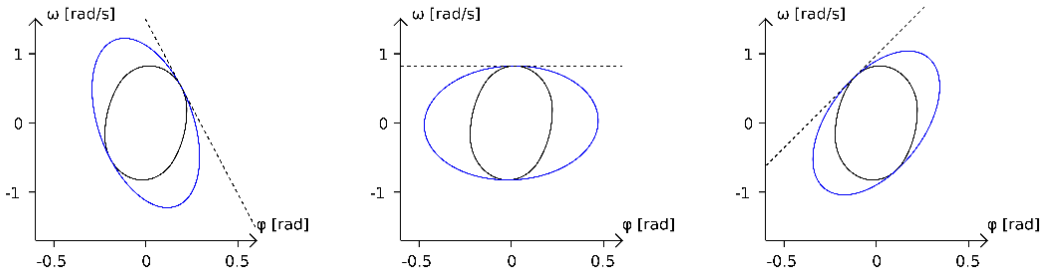


Figure 5.3: A visualization of some ellipsoidal outer approximations of the set $X(t)$ for the pendulum from Example 5.6. The approximations have been found by solving the optimal control problem (5.2.8) for various directions $c \in \mathbb{R}^2$. Note that the ellipsoids touch the set $X(t)$ in the desired direction.

the excitation and angular velocity of the pendulum, while w is an uncertain force with $|w(\tau)| \leq 1\,\mathrm{N}$ acting at the mass point.

Figure 5.2 shows a sketch of the pendulum as well as a visualization of the set $X(t)$ of reachable states at time $t = 1.2\,\mathrm{s}$ simulating the pendulum with the concrete values $L := 1\,\mathrm{m}$, $g := 9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}$, and $m := 1\,\mathrm{kg}$.

Theorem 5.2 gives us a practical tool to compute ellipsoidal outer approximations which are tight in the sense that they touch the set $X(t)$ in a given direction. Note that there are efficient tools available to solve optimal control problems of the form (5.2.8). Figure 5.3 shows some tight ellipsoidal outer approximations which have been computed by solving the optimal control problem (5.2.8) numerically choosing various directions $c \in \mathbb{R}^2$.

Note that the optimal control problem (5.2.8) from Theorem 5.2 is in general a non-convex optimization problem. In Example 5.6 we have used a standard nonlinear optimal control solver, which finds in general only local solutions of optimal control problems. Nevertheless, in Example 5.6 the numerical experience is that local search routines (based on Newton type methods) have absolutely no problem to converge to the global solution – independently of the initialization. For the example which is shown in Figure 5.3 we can clearly observe that the ellipsoids found by the local optimization routine touches the set $X(t)$ as expected. Is this always the case? In the following consideration we will show that under mild conditions on the function Ω_τ, we can indeed prove that every local minimizer of the optimal control problem (5.2.8) is also a global minimizer.

For this aim, we introduce an additional assumption on the function Ω_τ such that the optimal control problem (5.2.8) can equivalently be transformed into a convex optimization problem:

Assumption 5.3: *We assume that the function Ω_τ is not only anti-homogeneous but also chosen in such a way that the function $\chi_\tau : \mathbb{R}^{n_x} \times \mathbb{R}^n_{++} \to \mathbb{R}$ defined as*

$$\forall c \in \mathbb{R}^{n_x}, \forall \nu \in \mathbb{R}^n_{++} : \qquad \chi_\tau(c,\nu) \;:=\; c^T\,\Omega_\tau\,(\,e^\nu\,)\;c \tag{5.2.9}$$

is a convex function in ν for all $c \in \mathbb{R}^{n_x}$. Here, the exponential function is defined component-wise, i.e., such that $\exp(\nu) \;:=\; (\,\exp(\nu_1),\ldots,\exp(\nu_n)\,)^T$.

Example 5.7: If we model the uncertainty set as a sum of ellipsoids, i.e., if we employ the function $\Omega_\tau(\lambda) = \sum_{i=1}^n \lambda_i^{-1} W_i$ with positive semi-definite matrices $W_i \in \mathbb{S}^n_+$, the above assumption is satisfied, as the function

$$\chi_\tau(c,\nu) \;=\; \sum_{i=1}^n c^T\,W_i\,c\,e^{-\nu_i}$$

is obviously convex in ν.

Example 5.8: If we model the uncertainty set as an intersection of centered ellipsoids, i.e., if we employ the function $\Omega_\tau(\lambda) = (\sum_{i=1}^n \lambda_i W_i)^{-1}$ with positive definite matrices $W_i \in \mathbb{S}^n_{++}$, it can also be verified that the corresponding function χ is convex in ν.

Now, the idea is to use Assumption 5.3 together with Remark 2.8. If Assumption 5.3 holds, problem (5.2.8) can be interpreted as a (generalized) geometric optimal control problem. More precisely, we can perform a variable substitution of the form $\lambda_i(\tau) \;:=\; e^{\nu_i(\tau)}$ such that the optimal control problem (5.2.8) can equivalently be tranformed into an optimization problem of the form

$$\begin{aligned}
[V(t,c)]^2 \;&=\; \inf_{\nu(\cdot)} \;\int_0^t \Big(c^T H_t(\tau)\,\Omega_\tau\left(\,e^{\nu(\tau)}\,\right)H_t(\tau)c\Big)\;\left(\int_0^t \sum_{j=1}^n e^{\nu_j(\tau')}\,\mathrm{d}\tau'\right)\,\mathrm{d}\tau \\
&=\; \inf_{\nu(\cdot)} \;\int_0^t \int_0^t \sum_{j=1}^n \chi_\tau\left(\,H_t(\tau)c\,,\,\nu(\tau) - \nu_j(\tau')\mathbf{1}\,\right)\,\mathrm{d}\tau\,\mathrm{d}\tau'\,.
\end{aligned}$$

This is a convex optimization problem, as the function χ is convex in its second argument and the (infinite) sum over convex functions remains convex. It depends on the context

and our aim whether it is more suitable to write the optimization problem in the above convex form or in the form of problem (5.2.8) from Theorem 5.2. Both formulations are equivalent, which implies in particular that if Assumption 5.3 holds, then every local minimum (infimum) of the optimal control problem (5.2.8) is also a global minimum (infimum).

5.3 Uncertainty Propagation in Nonlinear Dynamic Systems

In this section, we discuss how to conservatively approximate the set of reachable states for an uncertain nonlinear dynamic system of the form

$$\forall \tau \in [t_1, t_2]: \quad \dot{x}(\tau) \;=\; f(\tau, x(\tau), w(\tau)) \; .$$

Here, the aim is - as in the consideration of linear dynamic systems - to construct a parameterized robust positive invariant tube, i.e., a set valued function of the form $\mathbb{X} : [t_1, t_2] \to \Pi(\mathbb{R}^{n_x})$ which satisfies

$$\forall \tau \in [t_1, t_2]: \quad \mathbb{X}(\tau^+) \;\supseteq\; F(\tau, \mathbb{X}(\tau), W(\tau)) \; .$$

Unfortunately, nonlinear dynamic systems are in general much more difficult to treat than linear systems. In order to understand the problem, we consider a nonlinear version of Example 5.6:

Example 5.9: Let us consider the following nonlinear pendulum model of the form

$$\begin{aligned} \dot{\varphi}(t) &= \omega(t) \\ \dot{\omega}(t) &= -\frac{g}{L}\sin(\varphi(t)) + \frac{\cos(\varphi(t))w(t) + \sin(\varphi(t))v(t)}{mL} \; . \end{aligned}$$

Here, w is an unknown horizontal force satisfying $|w(t)| \leq 1\,\mathrm{N}$. Moreover, v is an unknown vertical force satisfying $|v(t)| \leq \alpha$. Note that the force v is completely overseen if we linearize the model equations around the steady state. Here, the linearized version of the above model equations coincides with the linear example from the introduction. In Figure 5.4, we can find a sketch of the pendulum model as well as a visualization of the set of reachable states $X(t)$ at the time $t = 1.2\,\mathrm{s}$. In this example we have used $\alpha := 3\,\mathrm{N}$. Unfortunately, the nonlinear set $X(t)$ is larger than the corresponding set of reachable states for the linearized equations. Thus, our example illustrates that it is in

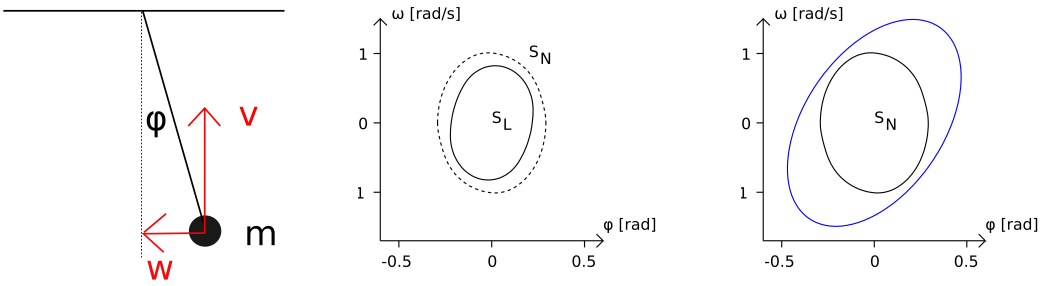


Figure 5.4: In the right part of the figure a sketch of the nonlinear pendulum model is shown while the visualization in the middle shows the set X_{N} of reachable states of this nonlinear model (dashed line) as well as the set X_{L} of reachable states of the associated linear approximation. In this example, X_{N} contains the set X_{L} illustrating that it is in general not enough to regard the linear approximation only. The right part of the figure shows a conservative and sub-optimal ellipsoidal outer approximation of the set X_{N}.

general too optimistic to consider linear approximations only. In particular, if we choose a very large upper bound α of the vertical force we can show that the linear approximation can be arbitrarily bad. Note that in the example which is visualized in Figure 5.4 the set $X(t)$ of the reachable states seems convex. However, this is by accident. For general nonlinear systems, we do not know anything about the structure of the set $X(t)$. The aim of the following considerations will be to compute at least sub-optimal ellipsoidal outer approximations of the set $X(t)$ as outlined in the right part of Figure 5.4.

Construction of Nonlinearity Estimates

In order do deal with a nonlinear right-hand side, we first introduce the central path, i.e., the trajectory which the state of the nonlinear dynamic system would follow if no uncertainties were present. Here, we assume that the point-symmetric uncertainty sets $W(\tau)$ for the input w are given by Assumption 5.2 such that the central input, i.e., $w(\tau) = 0$ for all $\tau \in \mathbb{R}$, corresponds to the case when there is no uncertainty. Moreover, we assume that our knowledge about the state $x(t_1)$ at the time t_1 is of the form $x(t_1) \in X_1 := \mathcal{E}(Q_1, q_1)$, where $q_1 \in \mathbb{R}^{n_x}$ is the central initial value. The central path (or reference function) $q : [t_1, \infty) \to \mathbb{R}^{n_x}$ is now defined to be the solution of the nominal differential equation

$$\forall \tau \in [t_1, \infty) : \qquad \dot{q}(\tau) = \varphi(\tau, q(\tau)) := f(\tau, q(\tau), 0) \quad \text{with} \quad q(t_1) = q_1 \, .$$

Recall our assumption that the right-hand side function f is uniformly Lipschitz continuous with respect to x which implies that the central path q exists. Our strategy is to decompose the dynamic system into a linear and a nonlinear part

$$\dot{x}(\tau) \;=\; d(\tau) + A(\tau)\,(\,x(\tau) - q(\tau)\,) + B(\tau)w(\tau) \;+\; f_{\text{nonlinear}}(\,\tau,\, q(\tau),\, x(\tau),\, w(\tau)\,)\;.$$

Here, the functions A, B and d are integrable functions with suitable dimensions while the function $f_{\text{nonlinear}} : [t_1, \infty) \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is simply defined in such a way that the above dynamic system is for all $\tau \in [t_1, \infty)$ equivalent to original dynamic equation. For the case that the right-hand side function f is differentiable in x and w, we may define the functions A, B, and d for all $\tau \in [t_1, \infty)$ by

$$A(\tau) \;:=\; \frac{\partial f(\tau, q(\tau), 0)}{\partial x}\;, \quad B(\tau) \;:=\; \frac{\partial f(\tau, q(\tau), 0)}{\partial w} \quad \text{and} \quad d(\tau) \;:=\; f(\tau, q(\tau), 0)\;.$$

However, the following consideration also applies, if f is not differentiable, as the decomposition in linear and nonlinear terms is so far redundant.

We plan to regard the nonlinear terms, collected in the function $f_{\text{nonlinear}}$, as an additional uncertainty, which can hopefully be bounded under suitable assumptions. This seems to be a crude plan and we have to be careful to not introduce an unnecessary amount of conservatism. However, we shall see in the following that the level of conservatism rather depends on how wisely we choose our estimate of the nonlinear terms. The main idea is to introduce the following assumption:

Assumption 5.4: *We assume that we have an explicit nonlinearity estimate for the right-hand side function f. I.e., we assume that we have*

$$\forall \lambda \in \Delta^{m-n} : \quad f_{\mathit{nonlinear}}\,(\,\tau,\, q(\tau),\, x(\tau),\, w(\tau)\,) \;\in\; \mathcal{E}\,(\,\Omega_N(\,\tau,\, q(\tau),\, Q,\, \lambda\,)\,) \quad (5.3.1)$$

for all $x(\tau) \in \mathcal{E}(Q, q(\tau))$, for all $w \in \mathcal{W}$, for all $\tau \in [t_1, \infty)$, and for all $Q \in S_+^{n_x}$. In this context, $m \geq n$ is a given integer and

$$\Omega_N : [t_1, \infty) \times \mathbb{R}^{n_x} \times \mathbb{S}_+^{n_x} \times \mathbb{R}_{++}^{m} \to \mathbb{S}_+^{n_x}$$

is assumed to be a positive semi-definite function, which is anti-homogeneous in λ.

From a mathematical point of view, the above assumption does not add a main restriction as we do not even require differentiability of f. However, in practice, it might of course be hard to find suitable functions Ω_N which satisfy the above assumption. Nevertheless, in the following we will demonstrate that there are many interesting cases in which such a function Ω_N can be constructed.

Example 5.10: Let us consider the case that we have a function f for which each component is convex quadratic in x and linear in w with $y(t) = 0$, i.e., we regard the case that we have

$$f_{\mathsf{nonlinear},i}(\tau, q, x, w) \;=\; x^T C x$$

for all components i and some positive semi-definite matrix C (which is for simplicity assumed to be the same for all components). As we can use the inequality $x^T C x \leq \mathrm{Tr}\,(QC)$ whenever $x \in \mathcal{E}(Q)$, we can employ the explicit nonlinearity estimate

$$\Omega_{\mathsf{N}}(\,\tau,\, q(\tau),\, Q,\, \lambda\,) \;:=\; [\,\mathrm{Tr}\,(QC)\,]^2\;\mathrm{diag}\,(\lambda)^{-1}$$

in order to satisfy the above assumption with $\lambda \in \mathbb{R}^{n_x}$ (compare with Example 5.4). Here, we have first overestimated the nonlinear function $f_{\mathsf{nonlinear}}$ by a box (i.e. component-wise) and second we used that a box can be written as a sum of ellipsoids. Note that there are also other nonlinearity estimates possible. For example the choice[1]

$$\Omega_{\mathsf{N}}(\,\tau,\, q(\tau),\, Q,\, \lambda\,) \;:=\; \left[\,\sigma_{\mathsf{max}}\,\left(Q^{\frac{1}{2}}\,C\,Q^{\frac{1}{2}}\right)\right]^2 \mathrm{diag}\,(\lambda)^{-1}$$

leads to a less conservative nonlinearity estimate requiring the computation of a maximum eigenvalue.

Example 5.11: Let us consider the case that we have an uncertain dynamic system of the form

$$\dot{x}(\tau) \;=\; (\,A(\tau) \,+\, C(\tau)E(\tau)D(\tau)\,)\;x(\tau) \,+\, B(\tau)\,v(\tau)\,. \tag{5.3.2}$$

Here, the functions A, B, C, and D are given matrix valued functions with appropriate dimensions while the matrix valued function E and the vector valued function v are regarded as uncertainties. The corresponding convex uncertainty sets for the input $w \;:=\; \left(\mathrm{vec}(E)^T\,,\, v^T\right)^T$ are assumed to be of the form

$$W(\tau) \;:=\; \left\{\, w(\tau) \;\mid\; v(\tau)^T v(\tau) \;\leq\; 1 \quad \text{and} \quad E(\tau)E(\tau)^T \;\preceq\; I \,\right\}\,.$$

In order to avoid confusion, we point out that the above system is in our context regarded as if it were a nonlinear dynamic system, although uncertain systems of this or a very similar form are typically introduced within the context of linear system theory and the H_∞-norm [82, 244], which is motivated by the fact that the right-hand side function is linear in the state x. However, from an optimization perspective, the problem of our

[1] We use the notation $\sigma_{\mathsf{max}}(S)$ to denote the maximum eigenvalue of a symmetric matrix S.

interest is nonlinear in the sense that the right and side is not jointly linear in the uncertain input $E(t)$ and the uncertain state $x(t)$. This is unfortunately a nonlinearity in our context, as we have to regard the functions x, w, and E as the optimization variables of the adverse player. The corresponding nonlinear term satisfies

$$f_{\text{nonlinear}}(t,q,x,w) \;=\; C(t)E(t)D(t)x(t) \;\in\; \mathcal{E}(\,\sigma_{\max}\Big(\,D(t)QD(t)^T\,\Big)\;C(t)C(t)^T\,)$$

whenever $x(t) \in \mathcal{E}(Q)$ as well as $E(t)E(t)^T \preceq I$. Consequently, we can employ a nonlinearity estimate of the form

$$\Omega_{\mathsf{N}}(\,t,\,q(t),\,Q,\,\lambda\,) \;:=\; \frac{\sigma_{\max}\Big(\,D(t)QD(t)^T\,\Big)}{\lambda}\,C(t)C(t)^T\,, \tag{5.3.3}$$

which satisfies all requirements from Assumption 5.4.

Example 5.12: Let us come back to the nonlinear pendulum model from Example 5.9. As the first component of the right-hand side function is linear, we only need a non-trivial nonlinearity estimate for the second component

$$f_2(x,w) = -\frac{g}{L}\sin(x_1) + \frac{1\mathsf{N}\,\cos(x_1)w_1 + 3\mathsf{N}\,\sin(x_1)w_2}{mL}\,.$$

We assume now that the pendulum is only operated on a feasible domain of the form $\mathcal{F}_x \;:=\; \Big\{\,(x_1,x_2)^T \;\mid\; |x_1| \,\leq\, \frac{\pi}{2}\,\Big\}$ on which we define the nonlinear term as

$$f_{\text{nonlinear},2}(x,w) \;:=\; f_2(x,w) - f_2(0,0) - \frac{\partial f_2(0,0)}{\partial x}x - \frac{\partial f_2(0,0)}{\partial w}w\,.$$

In this context, we can employ the inequality

$$\begin{aligned} f_{\text{nonlinear},2}(x,w) \quad &\leq \quad \chi(Q) \\ \text{with}\chi(Q) \quad &:= \quad \frac{g}{L}\left[\sqrt{Q_{11}} - \sin(\sqrt{Q_{11}})\right] + \frac{\left[1-\cos(\sqrt{Q_{11}}) + 3\sin(\sqrt{Q_{11}})\right]\mathsf{N}}{mL} \end{aligned}$$

for all $x \in \mathcal{E}(Q)$ with $\sqrt{Q_{11}} \,\leq\, \frac{\pi}{2}$ and $y = 0$. Using this definition of the short-hand χ, we can construct a nonlinearity estimate of the form

$$\Omega_{\mathsf{N}}(\,t,\,q(t),\,Q,\,\lambda\,) \;:=\; \frac{1}{\lambda}\begin{pmatrix} 0 & 0 \\ 0 & \chi(Q)^2 \end{pmatrix}$$

with $\lambda \in \mathbb{R}_{++}$. This nonlinearity estimate satisfies the requirement of Assumption 5.4 as long as we restrict the condition to hold on the specified domain $\mathcal{F}_x$ only.

Construction of Ellipsoidal Uncertainty Tubes

Let us try to transfer the construction principles for ellispoidal tubes which has been derived in Section 5.2. For this aim, we regard two functions $\nu_1 : \mathbb{R} \to \mathbb{R}^n_{++}$ and $\nu_2 : \mathbb{R} \to \mathbb{R}^{m-n}_{++}$, one taking care of the uncertainty w and one taking care of the nonlinear terms, as well as $\kappa := \left(\nu_1^T, \nu_2^T\right)^T$. After decomposing the nonlinear right-hand side function into linear and other terms, which have to be over-estimated, we collect the influence of the two associated functions Ω_τ and Ω_{N} again summarizing them within one function $\Omega_{\text{total}} : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{S}^{n_x}_+ \times \mathbb{R}^m \to \mathbb{S}^{n_x}_+$, which we define as

$$\Omega_{\text{total}}(\tau, q, Q, \kappa) := B(\tau)\Omega_\tau(\nu_1)B(\tau)^T + \Omega_{\mathsf{N}}(\tau, q, Q, \nu_2) .$$

Here, the motivation is to construct a matrix valued differential equation which can be used to generate robust positive invariant tubes for nonlinear dynamic systems.

Definition 5.3: *Using the above notation, we define for any function $\kappa : [t_1, \infty) \to \mathbb{R}^m_{++}$ a nonlinear matrix valued differential equation of the form*

$$\forall \tau \in [t_1, \infty) : \quad \dot{Q}(\tau) = \Phi(\tau, q(\tau), Q(\tau), \kappa(\tau)) .$$

In this context, we are using the following short hand for the right-hand side expression

$$\Phi(\tau, q, Q, \kappa) := A(\tau)Q + QA(\tau)^T + \sum_{i=1}^{m} \kappa_i Q + \Omega_{\text{total}}(\tau, q, Q, \kappa) ,$$

which is defined for all $\tau \in [t_1, \infty)$, $q \in \mathbb{R}^{n_x}$, $Q \in \mathbb{S}^{n_x}_+$, and $\kappa \in \mathbb{R}^m_{++}$.

Note that the above differential equation for the matrix valued function Q has many similarities with the Lyapunov differential equation which has been analyzed within Theorem 5.1. The only difference is that the function Φ is in general a nonlinear function in Q, while Lyapunov differential equations are by definition linear in their matrix valued state. However, the fact that the differential equation for Q is nonlinear does not prevent us from transferring the statement of Theorem 5.1:

Theorem 5.3: *Let Assumptions 5.2 and 5.4 be satisfied, let q denote the central path, and let the function Φ be given by Definition 5.3. Now, if $Q : [t_1, t_2] \to \mathbb{S}^{n_x}_+$ and $\kappa : [t_1, t_2] \to \mathbb{R}^m_{++}$ are any functions which satisfy for all $\tau \in [t_1, t_2]$ a differential equation of the form*

$$\dot{Q}(\tau) = \Phi(\tau, q(\tau), Q(\tau), \kappa(\tau)) ,$$

then the function $\mathbb{X}(\cdot) := \mathcal{E}(\, Q(\cdot)\,,\, q(\cdot)\,)$ *is a robust positive invariant tube on the given time interval* $[t_1, t_2]$.

Proof: The main idea of the proof of this theorem is already motivated above: first, we over-estimate the uncertain term $B(\tau)w(\tau)$ at every time τ by one parameterized ellipsoid, and second we overestimate the nonlinear terms $f_{\text{nonlinear}}\,(\,\tau,\, x(\tau),\, w(\tau)\,)$ by another parameterized ellipsoid. We know already from our considerations for linear dynamic systems how to over-estimate the sum of two ellipsoids with another ellipsoid. Consequently, we can combine everything by simply adding the influences of two separately over-estimated terms within one function Ω_{total}. Thus, we find that the statement of the Theorem holds by construction. □

Example 5.13: Let us once more regard the nonlinear pendulum model together with the nonlinearity estimate from Example (5.12). Using this nonlinearity estimate and choosing a suitable generating function κ we can apply Theorem 5.3 to generate an ellipsoidal outer approximation of $X(t)$. This strategy has been applied in order to obtain the ellipsoid which is shown in the right part of Figure 5.4. Note that the ellipsoid is not optimal as we can certainly find smaller ellipsoids which also contain $X(t)$. Nevertheless, we have at least a conservative approximation of the set $X(t)$ which might still be improved by optimizing the function κ with respect to one or the other criterion, but the main difference to linear dynamic systems is that the approximation will in general not be tight.

Chapter 6

Robust Open-Loop Control

There are many ways to formulate robust optimization problems. In the previous chapters, we have seen that the introduction of robust counterpart formulations turned out to be useful, while the notation of semi-infinite optimization problem is also helpful in some cases. When we refer to the field of robust optimization for dynamic systems, one option is to transfer the formulations which have already been developed for static, finite dimensional robust optimization problem. For example, we can formulate robust optimization problems for the discretized version of an uncertain dynamic system, which leads to a large but structured static robust optimization problem, as it has been outlined in the introduction within Section 1.2. Taking this way has the advantage that all our previous formulations and methods for static robust optimization, as e.g. the sequential convex bilevel programming method, can be transferred as long as the recursive structure of the discrete dynamic propagation is exploited in the corresponding numerical algorithms. However, our notation of reachable sets of dynamic systems, as discussed within the previous chapter, suggests an alternative way to look at robust optimization, which appears natural and is tailored for uncertain dynamic systems. Here, the aim is to compute the influence of the uncertainty via its propagation, i.e., by storing robust positive invariant tubes in the finite dimensional state space, rather than discretizing the whole uncertain optimal control problem. This strategy is motivated by the fact that the uncertain input w can vary in time and is thus an infinite dimensional quantity. Similarly, we have in the optimal control context usually infinitely many constraints, which have to be robustly satisfied, as we want to formulate bounds on the states of the system or general nonlinear path constraints.

The aim of the following section is to develop a formulation which expresses what we understand when referring to robust optimal control problems and to exploit the computational framework of uncertainty propagation from the previous chapter.

6.1 Robust Optimization of Open-Loop Controlled Systems

We are interested in the optimization of open-loop controlled uncertain dynamic system. Here, we only consider the dynamic system on finite time-horizon intervals $[0, T_\mathrm{e}]$, while infinite time horizons will later be discussed within Section 6.3. The uncertain dynamic systems we consider are in general nonlinear and of the form

$$\forall \tau \in [0, T_\mathrm{e}]: \quad \dot{x}(\tau) \;=\; f(\tau, u(\tau), p, x(\tau), w(\tau)) \;,$$

where the notation is analogous to the previous section. The only new variables are the control input $u : [0, T_\mathrm{e}] \to \mathbb{R}^{n_u}$ as well as the parameter $p \in \mathbb{R}^{n_p}$. As it has extensively been discussed in Section 5.1, we may associate a set valued differential propagation with the above uncertain dynamic system, which can in our case be written as

$$\forall \tau \in [0, T_\mathrm{e}]: \quad X(\tau^+) \;=\; F(\tau, u(\tau), p, X(\tau), W(\tau)) \;.$$

Note that this notation is based on Assumption 5.1, which specifies our knowledge about the uncertain input w. A fairly general formulation of a robust optimal control problem for the above uncertain dynamic system can now be stated as follows:

$$\begin{array}{lrcl} \underset{u(\cdot),p,T_\mathrm{e},X(\cdot)}{\min} & M(\,p, T_\mathrm{e}, X(T_\mathrm{e})\,) & & \\ \text{s.t.} & X(\tau^+) &=& F(\tau, u(\tau), p, X(\tau), W(\tau)) \\ & X(0) &=& X_0 \\ & 0 &\geq& H(\tau, u(\tau), p, X(\tau), W(\tau)) \quad \text{for all } \tau \in [0, T_\mathrm{e}] \;. \end{array} \tag{6.1.1}$$

Here, we transfer the language from the field of nominal optimal control, which suggests to call the objective function $M : \mathbb{R}^{n_p} \times \mathbb{R}_+ \times \Pi(\mathbb{R}^{n_x}) \to \mathbb{R}$ a Mayer term, while the function $H : [0, T_\mathrm{e}] \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \Pi(\mathbb{R}^{n_x}) \times \Pi(\mathbb{R}^{n_w}) \to \mathbb{R}^{n_H}$ comprises a path constraint.

Definition 6.1: *We say that a function of the form $Z : \Pi(\mathbb{R}^{n_x}) \to \mathbb{R}$ is monotonically increasing, if for any sets $X, Y \subseteq \mathbb{R}^{n_x}$ with $X \subseteq Y$, we have that $Z(X) \leq Z(Y)$.*

In order to get familiar with the above way of formulating robust optimal control problems, we discuss possible choices for the functions M and H within the following examples. Within these examples, we also point out that most of the practically relevant choices for the functions M and H are (component-wise) monotonically increasing with respect to their set valued arguments $X(T_\mathrm{e})$ or $X(\tau)$, respectively:

Example 6.1: Let us consider the important case that we have already a nominal Mayer term of the form $m : \mathbb{R}^{n_p} \times \mathbb{R}_+ \times \mathbb{R}^{n_x} \to \mathbb{R}$. Our aim is to minimize the worst possible value for the term $m(p, T_\mathrm{e}, x(T_\mathrm{e}))$, where $x(T_\mathrm{e})$ is the state of the differential equation, which is unfortunately affected by uncertainties. In order to express this robust counterpart problem in the required form, we define the function M as

$$M(\, p, T_\mathrm{e}, X(T_\mathrm{e})\,) \;:=\; \sup_{x \in X(T_\mathrm{e})} m(\, p, T_\mathrm{e}, x\,)\;.$$

Here, we typically assume that m is continuous in x. As $X(T_\mathrm{e})$ is in many practical problems compact, we could also replace the supremum by a maximum. Note that the above robust counterpart function M is monotonically increasing in the set valued variable $X(T_\mathrm{e})$. Also note that nominal optimal control problems with Lagrange objective terms can for all theoretical purposes be reformulated into optimal control problems with Mayer terms by augmenting the differential equation with an auxiliary state. Thus, the above strategy can be applied to formulate robust counterparts for optimal control problems with given Lagrange terms, too.

Example 6.2: It is important to realize that robust optimal control problems do not have to originate from a robust counterpart formulation. For example the choice

$$M(\, p, T_\mathrm{e}, X(T_\mathrm{e})\,) \;:=\; \mathrm{diag}(X(T_\mathrm{e})) \;:=\; \sup_{x,y \in X(T_\mathrm{e})} \|x - y\|\;.$$

would lead to a minimization of the maximum distance of two points in the set $X(T_\mathrm{e})$, if $\|\cdot\| : \mathbb{R}^{n_x} \to \mathbb{R}_+$ denotes a suitable norm. Such a formulation can for example be useful, if we plan to design and optimize the robustness properties of a dynamic system directly. Another choice could be of the form

$$M(\, p, T_\mathrm{e}, X(T_\mathrm{e})\,) \;:=\; \frac{\int_{X(T_\mathrm{e})} \left\| x - \int_{X(T_\mathrm{e})} x \,\mathrm{d}x \right\|^2 \,\mathrm{d}x}{\int_{X(T_\mathrm{e})} 1 \,\mathrm{d}x}$$

planning to minimize the inertia of $X(T_\mathrm{e})$, i.e., the quadratic deviation to the center of gravity. Similarly, we can employ a function of the form

$$M(\, p, T_\mathrm{e}, X(T_\mathrm{e})\,) \;:=\; \int_{X(T_\mathrm{e})} 1 \,\mathrm{d}x$$

to formulate minimum volume problems. While Example 6.1 was based on a given Mayer term, which is assumed to have a physical interpretation within the world of nominal optimal control, the above choices have no such analogon. Rather, the three regarded choices for the function M have in common that they are always positive and that they can only be equal to zero, if the set $X(T_\mathrm{e})$ consists of one single point only, i.e., if the state at the time T_e is not affected by the uncertainties. In addition, we observe that all discussed choices for the function M are monotonically increasing with respect to the argument $X(T_\mathrm{e})$.

Example 6.3: Let us once more come back to the case that we have a given nominal Mayer term of the form $m : \mathbb{R}^{n_p} \times \mathbb{R}_+ \times \mathbb{R}^{n_x} \to \mathbb{R}$. As an alternative to robust counterpart formulations, we might be interested in an outer average formulation, which corresponds to choice

$$M(\, p, T_\mathrm{e}, X(T_\mathrm{e})\,) \;:=\; \frac{\int_{X(T_\mathrm{e})} m(p, T_\mathrm{e}, x)\;\mathrm{d}x}{\int_{X(T_\mathrm{e})} \mathrm{d}x}\;.$$

Similarly, we may regard inner average formulations of the form

$$M(\, p, T_\mathrm{e}, X(T_\mathrm{e})\,) \;:=\; m\left(p, T_\mathrm{e}, \int_{X(T_\mathrm{e})} x\,\mathrm{d}x\right)\;.$$

However, we should be aware of the fact that inner and outer average formulations are typically not monotonically increasing in $X(T_\mathrm{e})$.

Example 6.4: Note that the previous examples about the construction of meaningful objective functions M transfer similarly also for the construction of a constraint function H. For example, if we have a nominal constraint of the form

$$\forall \tau \in [0, T_\mathrm{e}]: \quad h(\,\tau,\, u(\tau),\, p,\, x(\tau),\, w(\tau)\,) \;\leq\; 0\;,$$

which should be satisfied for all realizations of the uncertainty, we can define an associated robust counterpart function H component-wise (with $\; i \in \{1, \ldots, n_H\}\;$) as

$$H_i(\tau, u(\tau), p, X(\tau), W(\tau)) \;:=\; \sup_{\substack{x \in X(\tau) \\ w \in W(\tau)}} h_i(\,\tau,\, u(\tau),\, p,\, x,\, w\,)\;.$$

In this case, the function H is component-wise monotonically increasing in the argument $X(\tau)$. The above robust counterpart construction of the path constraint function H is

the main application which we have in mind. However, in principle we may also allow any other construction which helps us to formulate "design criteria" for the uncertainty tube $X(\cdot)$. For example, we can require an upper bound on the volume or diameter of the sets $X(\tau)$ for all $\tau \in [0, T_{\mathrm{e}}]$.

Note that we are in many cases able to re-formulate our robust optimal control problem in such a way that it can be covered by formulation (6.1.1). In some other cases, we have to extend slightly this formulation, which is usually not difficult as most of the formulation strategies, which are well-known from the field of nominal optimal control, formally transfer by replacing the nominal state "x" with the set valued function "X". However, in order to discuss such techniques briefly, we collect some selected aspects in the following list without working out all details:

- **Uncertain Time-Invariant Parameters:** Formulation (6.1.1) covers the case that we have time-invariant parameters, whose actual value is unknown. This type of uncertainties should not be mixed up with the time-varying inputs w, as we do not exploit our information about the uncertainties in an efficient manner, otherwise. In order to briefly explain how to deal with this case, we start with a dynamic system of the form

  $$\forall \tau \in [0, T_{\mathrm{e}}]: \quad \dot{y}(t) \;=\; \hat{f}(\tau, u(\tau), p, y(\tau), \hat{p}, w(\tau)) \quad \text{with} \quad x(0) \;=\; x_0 \;,$$

  where the notation and assumptions are all as before, but there is an additional unknown time-invariant parameter $\hat{p} \in W_{\hat{p}} \subseteq \mathbb{R}^{n_{\hat{p}}}$. This dynamic system can be reformulated by introducing an augmented state $x : [0, T_{\mathrm{e}}] \to \mathbb{R}^{n_x}$ with dimension $n_x := n_y + n_{\hat{p}}$, which is defined as $x(t) := \left(y(t)^T , \hat{p}^T \right)^T$ and which satisfies a differential equation of the form

  $$\dot{x}(t) \;=\; \begin{pmatrix} \hat{f}(\tau, u(\tau), p, y(\tau), \hat{p}, w(\tau)) \\ 0 \end{pmatrix} \quad \text{with} \quad y(0) = \begin{pmatrix} y_0 \\ \hat{p} \end{pmatrix} .$$

  Here, we assume that the uncertain initial state $y(0)$ is known to be in a given set $Y_0 \subseteq \mathbb{R}^{n_y}$. The new uncertainty set X_0 for the initial value $x(0)$ takes the form $X_0 := Y_0 \oplus W_{\hat{p}}$. Using this re-formulation trick, formulation (6.1.1) can deal with uncertain time-varying inputs, uncertain time constant inputs, and uncertain initial values.

- **Uncertain Process Durations:** Note that robust time-optimal control problems are included in formulation (6.1.1) as the end time T_{e} can be an optimization variable as well. However, we can in principle even regard the case that the duration T_{e} of the dynamic process is unknown. In this case, the differential equation can be re-scaled with an unknown parameter, such that the previous comment applies.

- **Boundary Constraints:** The robust optimal control formulation could be extended by replacing the constraint $X(0) = X_0$ with more general constraints on the sets $X(0)$ and $X(T_{\mathrm{e}})$. Trying to transfer the typical formulation strategies for nominal optimal control problems, the first type of constraint that comes to our mind are boundary constraints, where we require the initial value constraint $X(0) = X_0$ to be jointly satisfied with an end-value constraint $X(T_{\mathrm{e}}) = X_{\mathsf{F}}$. Here, the sets $X_0, X_{\mathsf{F}} \subseteq \mathbb{R}^{n_x}$ are given. However, such a constraint is in most practical situations ambiguous, as we have usually limited degrees of freedom. In contrast, a highly relevant form of boundary constraints are the periodic boundary constraints, where we replace the initial value constraint $X(0) = X_0$, with a condition of the form $X(0) = X(T_{\mathrm{e}})$. As this case is very important, we will discuss it later in full detail within Section 6.3.

- **Generalized Lagrange Terms:** Within Example 6.1 we have already remarked that robust counterpart problems for given nominal Lagrange terms (or Bolza-objectives) can be covered by formulation (6.1.1) as Bolza-objectives can always be re-written into standard Mayer terms. However, there is a practically relevant generalization of formulation (6.1.1) possible, if we also allow generalized Langrage terms, which may explicitly depend on the uncertainty sets, and which are of the form

$$\int_0^T \mathcal{L}(\tau, u(\tau), p, T_{\mathrm{e}}, X(\tau), W(\tau))\,\mathrm{d}\tau\ . \tag{6.1.2}$$

  Here, $\mathcal{L} : \mathbb{R} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \mathbb{R}^{+} \times \Pi(\mathbb{R}^{n_x}) \times \Pi(\mathbb{R}^{n_w})$ is an appropriate scalar valued function. This type of objective terms is for example needed, if we are interested in minimizing the average volume of the sets $X(\cdot)$ on the interval $[0, T_{\mathrm{e}}]$.

Note that – together with the above remarks and minor generalizations – formulation (6.1.1) can be considered as fairly general and it covers a large class of practically relevant robust open-loop optimal control problems. However, the disadvantage of this problem formulation is that at the current status there are no efficient algorithms known which solve this class of problems in its general form, with a high numerical accuracy, and in an acceptable

run-time for moderate and large state dimensions. Here, the main challenge is that our optimization variable X is a set valued function.

Nevertheless, we shall see in the following that it is possible to develop suitable approximation strategies, which search for sub-optimal solutions to problem (6.1.1) while guaranteeing robust feasibility. For this aim, we plan to apply the construction strategies for robust positive invariant tubes, which have been discussed in the previous Section 5.2.

Assumption 6.1: *We assume that we have functions φ, Φ, q_0, and Q_0 with appropriate dimensions such that the following property is satisfied: for any given control function $u : [0, T_\mathrm{e}] \to \mathbb{R}^{n_u}$, any parameter $p \in \mathbb{R}^{n_p}$, any function $\kappa : [0, T_\mathrm{e}] \to \mathbb{R}^m_{++}$, and any vector $\kappa_0 \in \mathbb{R}^{n_0}_{++}$, which admit solutions $q : [0, T_\mathrm{e}] \to \mathbb{R}^{n_x}$ and $Q : [0, T_\mathrm{e}] \to \mathbb{S}^{n_x}_+$ of the coupled differential equation*

$$\forall \tau \in [0, T_\mathrm{e}] : \quad \left\{ \begin{array}{rclrcl} \dot{q}(\tau) & = & \varphi(\tau, u(\tau), p, q(\tau), Q(\tau), \kappa(\tau)) & q(0) & = & q_0(\kappa_0) \\ \dot{Q}(\tau) & = & \Phi(\tau, u(\tau), p, q(\tau), Q(\tau), \kappa(\tau)) & Q(0) & = & Q_0(\kappa_0) \, , \end{array} \right.$$

the set valued function $\mathbb{X}(\cdot) := \mathcal{E}(Q(\cdot), q(\cdot))$ is a robust positive invariant tube on the interval $[0, T_\mathrm{e}]$ for which the condition $X_0 \subseteq \mathbb{X}(0)$ is also satisfied.

At this point, we recall our consideration from the previous sections, where we have discussed how we can construct functions φ, Φ, q_0, and Q_0 for linear and nonlinear dynamic systems which ensure that the above assumption can be met. In order to discuss cases in which Assumption 6.1 can be used to find sub-optimal approximate solutions of the original robust optimal control problem, we regard an optimal control problem of the following form:

$$\boxed{\begin{array}{ll} \inf\limits_{\xi(\cdot), \zeta(\cdot), \pi, T_\mathrm{e}} & M(\, p, \, T_\mathrm{e}, \, \mathcal{E}(Q(T_\mathrm{e}), q(T_\mathrm{e}))\,) \\ \text{s.t.} & \left\{ \begin{array}{rcll} \dot{q}(\tau) & = & \varphi(\tau, u(\tau), p, q(\tau), Q(\tau), \kappa(\tau)) & q(0) \; = \; q_0(\kappa_0) \\ \dot{Q}(\tau) & = & \Phi(\tau, u(\tau), p, q(\tau), Q(\tau), \kappa(\tau)) & Q(0) \; = \; Q_0(\kappa_0) \, , \\ 0 & \geq & H(\tau, u(\tau), \, \mathcal{E}(Q(\tau), q(\tau)), \, W(\tau)) & \text{for all } \tau \in [0, T_\mathrm{e}] \, . \end{array} \right. \end{array}} \tag{6.1.3}$$

In this optimal control problem, we have collected the differential states in the function $\xi := (q, Q)$, the controls in the function $\zeta := (u, \kappa)$, and the parameters in the vector $\pi := (p, \kappa_0)$. Let us summarize the properties of this optimal control problem within the following Theorem.

Theorem 6.1: *Let Assumption 6.1 be satisfied. If the function H is component-wise monotonically increasing in $X(\tau)$, then every feasible input $(u(\cdot), p)$ of the auxiliary optimal control problem* (6.1.3) *corresponds to a feasible input of the original robust optimal control problem* (6.1.1)*. Moreover, if in addition the objective function M is monotonically increasing in $X(T_\mathrm{e})$, then the objective value of problem* (6.1.3) *is an upper bound on the objective value of the original problem* (6.1.1)*. In other words, any solution of problem* (6.1.3) *yields a feasible but possibly sub-optimal solution of the original robust optimal control problem.*

The main advantage of the optimal control problem (6.1.3) is that we do not need any set valued functions anymore. However, in order to reduce this problem to a standard optimal control problem, we still have to mention how the functions M and H can be evaluated in practice. The aim of the following examples is to outline how problem (6.1.3) can in most of the practically relevant situations be re-written as a standard nonlinear optimal control problem such that existing numerical algorithms and software can be applied:

Example 6.5: Let us assume that the function M is a robust counterpart objective for an existing nominal Mayer term m, as discussed within Example 6.1. If m is linear in x, i.e., if m has the form

$$m(p, T_\mathrm{e}, x) \;:=\; c(p, T_\mathrm{e})^T x + d(p, T_\mathrm{e}) \;,$$

then the evaluation of the associated robust counterpart function can explicitly be evaluated by employing the support function of an ellipsoidal set:

$$\begin{aligned} M(\,p, T_\mathrm{e}, \mathcal{E}(Q(T_\mathrm{e}), q(T_\mathrm{e}))\,) \;&=\; \sup_{x \,\in\, \mathcal{E}(Q(T_\mathrm{e}), q(T_\mathrm{e}))} m(\,p, T_\mathrm{e}, x\,) \\ &=\; \sqrt{c(p, T_\mathrm{e})^T Q(T_\mathrm{e}) c(p, T_\mathrm{e})} + c(p, T_\mathrm{e})^T q(T_\mathrm{e}) + d(p, T_\mathrm{e}) \;. \end{aligned}$$

The latter expression can simply be used as a nonlinear objective term. As most of the nonlinear optimal control solvers ask for differentiability of the objective, we have to regularize the square-root term if necessary or reformulate it as an second order cone constraint provided that our optimal control solver can deal explicitly with such problems. Similarly, we can deal with the case that m is a (not necessarily concave) quadratic form in x. In this case, we can employ the S-procedure the re-formulate the maximization problem into an equivalent minimization problem adding the required dual variables as slack-variables (parameters) to the optimal control problem. Recall that such formulation techniques have been discussed within Chapter 3 such that we do not mention all details

in this example. Finally, if m is a more general function for which an upper bound on the Hessian matrix with respect to x is available, we can still employ the Lagrangian dual relaxation techniques from Chapter 3 as long as we can accept to possibly inherit a higher level of conservatism. An analogous remark applies for the component-wise reformulation of robust counterpart constraint functions H.

Example 6.6: Let us come back to the possible choices for the Mayer term, which have been discussed within Example 6.2. For the case that we want to minimize the maximum distance of two points in the set $X(T_{\mathrm{e}})$, which is replaced in formulation (6.1.3) by an ellipsoid of the form $\mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))$, we obtain the following expression for the objective

$$M(\,p, T_{\mathrm{e}}, \mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))\,) \;=\; \mathsf{diag}(\mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))) \;:=\; 2\,\sigma_{\mathsf{max}}\,(\,Q(T_{\mathrm{e}})\,)\;.$$

Thus, this formulation requires the computation of a maximum eigenvalue to evaluate the objective. As this is a non-smooth objective, the maximum eigenvalue can alternatively be replaced by a semi-definite inequality by introducing a slack variable.

Similarly, the inertia minimization formulation leads to an term of the form

$$\begin{aligned} M(\,p, T_{\mathrm{e}}, \mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))\,) \;&:=\; \frac{\int_{\mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))} \left\| x - \int_{\mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))} x \,\mathrm{d}x \right\|^2 \,\mathrm{d}x}{\int_{\mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))} 1 \,\mathrm{d}x} \\ &=\; \frac{\mathrm{Tr}\,(\,Q(T_{\mathrm{e}})\,)}{n_x + 2}\;, \end{aligned}$$

while a minimum volume formulation leads to a term of the form

$$M(\,p, T_{\mathrm{e}}, \mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))\,) \;:=\; \int_{\mathcal{E}(Q(T_{\mathrm{e}}), q(T_{\mathrm{e}}))} 1 \,\mathrm{d}x \;=\; \frac{\pi^{\frac{n_x}{2}}\,\mathrm{Det}(\,Q(T_{\mathrm{e}})\,)}{\Gamma\left(\frac{n_x}{2}+1\right)}\;.$$

All of these objective terms can typically be treated with standard nonlinear optimal control solvers, i.e., the objective term is in all cases reduced to a standard formulation.

6.2 Interlude: Robust Optimal Control of a Tubular Reactor

In this section, we apply the robust optimal control technique from the previous section to an example which originates from the field of chemical engineering. Here, the studied setting involves a tubular chemical reactor operating under steady-state conditions. Inside

the reactor an irreversible and exothermic reaction takes place, while a surrounding jacket enables the heat removal. The reactor model adopted is based on the 1D plug flow model from [161]. The main modeling assumptions are:

1. steady-state condition,
2. no axial dispersion,
3. perfect radial mixing,
4. a constant density and heat capacity of the fluid,
5. a negligible heat resistance between the reactor and its jacket, and
6. an Arrhenius law dependence of the reaction rate on the temperature.

Using the spatial coordinate z along the reactor as the independent variable yields a highly nonlinear ODE system for $z \in [0, L]$:

$$\begin{aligned} \frac{\partial}{\partial z} x_1(z) &= \frac{\alpha}{v}(1 - x_1(z)) \exp\left(\frac{\gamma x_2(z)}{1 + x_2(z)}\right) , \\ \frac{\partial}{\partial z} x_2(z) &= \frac{\alpha\delta}{v}(1 - x_1(z)) \exp\left(\frac{\gamma x_2(z)}{1 + x_2(z)}\right) + \frac{\beta(z)}{v}\left(u(z) - x_2(z)\right) . \end{aligned}$$

Here, the states x_1, and x_2 are scaled versions of the concentration C and reactor temperature T respectively. More precisely, we have

$$C(z) := C_{\mathsf{F}}(1 - x_1(z)) \quad \text{and} \quad T(z) := T_{\mathsf{F}}(1 + x_2(z)) - 273.15^{\circ}\mathsf{C} ,$$

where C_{F} and T_{F} are the given reactant concentration and temperature of the feed stream. The parameters α, v, δ, and γ are given constants.

The input $u(z) = \frac{T_{\mathsf{J}}(z) - T_{\mathsf{F}}}{T_{\mathsf{F}}}$ contains the dimensionless version of the jacket temperature $T_{\mathsf{J}}(z)$ which can be controlled along the reactor. Here, the main difficulty is, that the heat transfer coefficient β is often hard to measure/estimate and may in particular vary along the reactor, e.g., due to local fouling at the reactor wall. In our study we assume that β is known to vary at most $\Gamma := 6\%$ around its nominal value, i.e., we use an uncertainty model of the form

$$\beta(z) := \beta_{\text{nominal}}(1 + \Gamma w(z)) ,$$

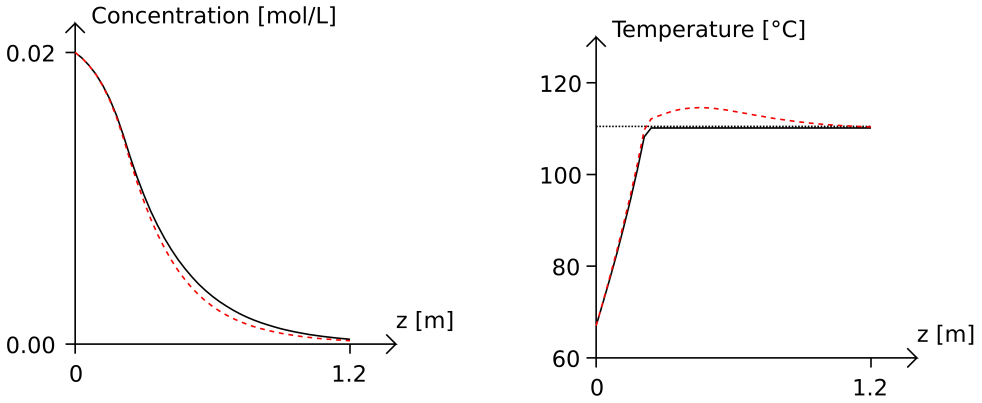


Figure 6.1: The concentration C and temperature T in dependence on the spatial coordinate $z \in [0, L]$ (with $L = 1.2\,\text{m}$) at the nominally optimal solution (solid line). The temperature constraint $T_{\text{max}} = 110°\text{C}$ is active over large parts of the reactor. The dashed line shows a simulation with 6% uncertainty in the heat transfer coefficient leading to a violation of the maximum temperature constraint.

with $w(z) \in W(z)$ denoting the scaled version of the time-varying uncertainty, i.e., we assume $W(z) := [-1, 1]$.

Maximizing the conversion in the reactor amounts to minimizing an objective of the form

$$\Phi := C_{\text{F}}(1 - x_1(L)) \,. \tag{6.2.1}$$

Now, we first minimize Φ nominally, i.e., for $\beta(z) = \beta_{\text{nominal}}$ without taking the uncertainty into account, subject to initial value conditions of the form $x(0) = 0$, a maximum temperature constraint $T(z) \leq T_{\text{max}}$, as well as upper and lower control bounds on the jacket temperature denoted as $T_{\text{J,min}} \leq T(z) \leq T_{\text{J,max}}$ which should be satisfied for all $z \in [0, L]$. All concrete numerical values for these constants are taken from [161]. Only the reactor length L and the upper reactor temperature bound T_{max} have been set to $1.2\,\text{m}$ and $110°\text{C}$, respectively. The corresponding nominally optimal result for the concentration C and temperature T are shown as the solid lines in Figure 6.1.

Note that the maximum temperature constraint is active along a large part of the reactor tube. Now, we simulate the conversion and temperature once more by applying the nominally optimal control input but choosing some disturbance w, which satisfies $w(z) \in W(z)$ for all $z \in [0, L]$. The corresponding result is also shown in Figure 6.1 in form of the dashed lines. We can clearly see an overshoot in the reactor temperature, i.e., our constraint is violated. This is due to the fact that the uncertain heat transfer coefficient

directly affects the differential equation for the reactor temperature. Moreover, higher temperatures stimulate the reaction causing additional heat to be produced. Consequently, a nominal optimization of the reactor does not lead to an acceptable solution and can even lead to hazardous situations.

In the next step, we plan to take the uncertainty into account with the aim to solve a conservative robust counterpart problem of the form (6.1.3). Here, the main difficulty is to derive the nonlinearity estimate for the functions

$$\begin{aligned} f_1(x,u,w) &:= \frac{\alpha}{v}(1-x_1)\,\exp\left(\frac{\gamma x_2}{1+x_2}\right) \\ f_2(x,u,w) &:= \frac{\alpha\delta}{v}(1-x_1)\exp\left(\frac{\gamma x_2}{1+x_2}\right) + \frac{\beta_{\text{nominal}}(1+\Gamma w)}{v}\,(\,u-x_2\,) \end{aligned}$$

The main strategy is to use the general inequality $\exp(\,y\,) \;\leq\; 1+y+\frac{y^2}{2}\exp(\,|y|\,)$ which holds globally for all $y \in \mathbb{R}^{n_x}$. Using this inequality it can be shown that we can find nonlinearity estimates Ω_{N} as follows

$$\begin{aligned} j(q,Q) &:= \frac{\gamma}{(1+q_2)(1+q_2-\sqrt{Q_{22}})} \\ r_1(q,Q) &:= j(q,Q) + \frac{\sqrt{Q_{22}}}{2}\,j(q,Q)^2\,\exp(\,\sqrt{Q_{22}}\,|j(q,Q)|\,) \\ r_2(q,Q) &:= \frac{j(q,Q)}{1+q_2} + \frac{j(q,Q)^2}{2}\,\exp(\,\sqrt{Q_{22}}\,|j(q,Q)|\,) \\ l_1(q,u,Q) &:= \frac{\alpha}{v}\exp\left(\frac{\gamma q_2}{1+q_2}\right)\left[\,r_1(q,Q)\,\sqrt{Q_{11}Q_{22}} + r_2(q,Q)\,Q_{22}\right] \\ l_2(q,u,Q) &:= \delta\, l_1(q,u,Q) + \frac{\Gamma\beta_{\text{nominal}}\sqrt{Q_{22}}}{v} \\ \Omega_{\mathsf{N}}(\,\tau,\,q,\,Q,\,u,\,\lambda\,) &:= \begin{pmatrix} \frac{l_1(q,u,Q)^2}{\lambda_1} & 0 \\ 0 & \frac{l_2(q,u,Q)^2}{\lambda_2} \end{pmatrix}. \end{aligned}$$

Note that the nonlinearity estimate depends on the matrix $Q \in \mathbb{S}^2_+$ and the linearization point (central path) $q \in \mathbb{R}^2$. Moreover, the first component f_1 of the right-hand side function does not explicitly depend on w. Consequently, the over-estimation term l_1 satisfies $l_1(y,u,Q) \;=\; \mathbf{O}(\,\|Q\|\,)$ for small Q as the derivative of f_1 with respect to the states is locally Lipschitz continuous.

Remark 6.1: *The above nonlinearity estimate in our example can be simply derived with "paper and pencil". Also from an implementation point of view it is no problem to*

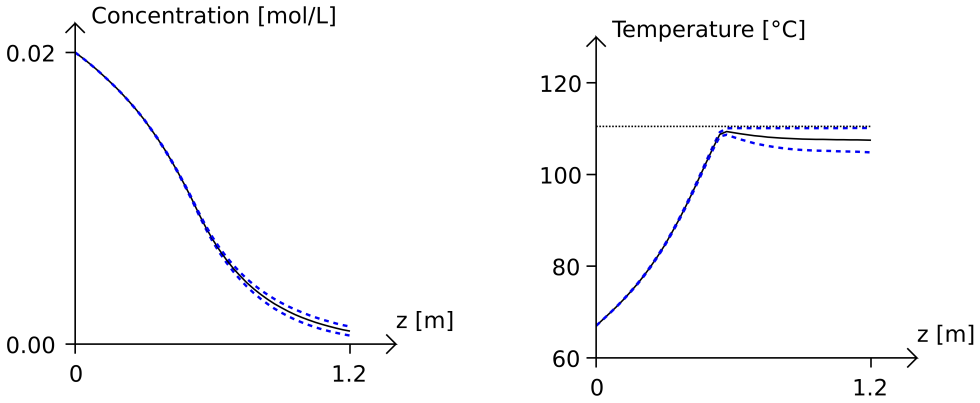


Figure 6.2: The robustly optimized concentration C and temperature T as a function of the spatial coordinate $z \in [0, L]$ (with $L = 1.2\,\mathrm{m}$) at the nominally optimal solution (solid line). The dotted lines show projections of the ellipsoid tube defining the region in which the states are guaranteed to be. Note that the maximum temperature constraint is guaranteed to be satisfied for all possible realizations of the uncertain heat transfer function assuming that the variation in β is less than 6%.

implement the function Ω_N as this requires basically to type five lines of code referring to the five equations in (6.2.2)*. However, there are many practical situations in which we end up with more lengthly right-hand side expressions. In this case, it might become inconvenient to derive nonlinearity estimates by hand. In principle, it is possible to automate the computation of nonlinearity estimates once some basic composition or chain rules are defined. Such an implementation could be analogous to existing symbolic tools like automatic differentiation or convexity detection [131, 106]. Here, even terms like exponentials, sines or cosines are not a problem on principle as we have seen in the above pendulum or chemical reactor examples. Working out this concept could be based on the ideas which have been developed in the field of interval arithmetics [27, 181]. A consequent application of these interval techniques is beyond the scope of this thesis but might be an interesting direction for future research.*

Once we have derived the above nonlinearity estimate, we can directly implement and solve the conservative robust counterpart problem of the form (6.1.3). The corresponding result is shown within Figure 6.2. As it can be seen, the robustified reactor temperature profile does not rise as sharply as before and it exhibits a back-off with respect to the upper reactor temperature bound. The ellipsoidal tube is rather narrow at the beginning, while it broadens when it is close to the upper limit. The explanation is that the uncertain

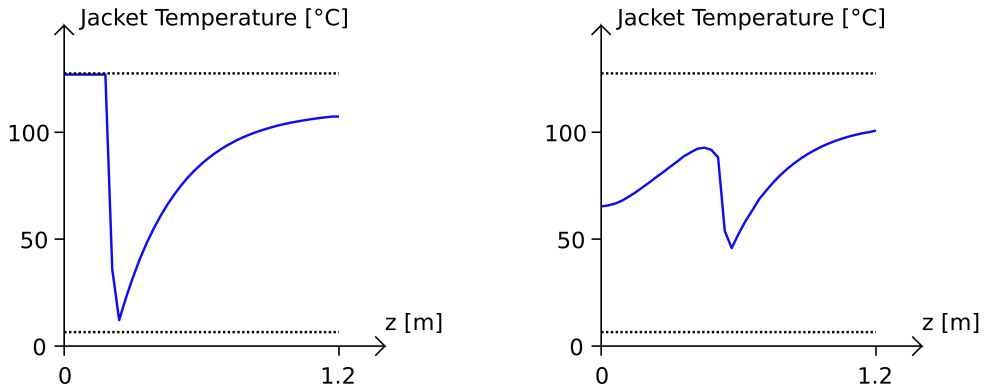


Figure 6.3: Left: The nominally optimized jacket temperature $T_J(z)$ (control input) in dependence on the spatial coordinate $z \in [0, L]$ (with $L = 1.2\,\text{m}$). The upper bound of the form $T_J(z) \leq 127°\text{C}$ is active for small z. Right: The corresponding robustly optimized jacket temperature for which the control bounds are not active.

heat transfer coefficient enters the dynamic equation for the temperature via the term $\beta(z)\,[T_J(z) - T(z)]$ where T_J is the controlled jacket temperature and T the temperature inside the reactor. Thus, if we adjust the control input $T_J(z)$ such that it coincides with the temperature $T(z)$ inside the reactor, the uncertainty cannot affect on the reaction itself, as $T_J(z) - T(z) = 0$ in this case. Thus, especially at the beginning of the tube, i.e., for small z, when there is still a high reactant concentration present, a robust optimizer chooses $T_J(z) \approx T(z)$ such that despite the large amount of reactant, the uncertainty hardly has an influence. However, as soon as $T(z)$ comes close to the upper limit, we cannot continue with this strategy as there is the danger of over-heating otherwise. The broadening ellipsoidal tube touches as expected the upper temperature limit, ensuring that the reactor temperature will not exceed this value (given the specified maximum uncertainty of 6% on the heat transfer coefficient β), while still trying to be as optimal as possible. As robustness typically induces conservatism in the optimal solution, the performance decreases also here, which is reflected by higher outlet concentrations of the reactant and, thus, lower conversions. The current loss in performance is easily explained as lower temperatures typically slow down irreversible reactions and, hence, yield lower conversions.

In the right part of Figure 6.3 the robustly optimized jacked temperature $T_J(z)$ (control input) is shown. In comparison to the nominally optimized control input (left part of Figure 6.3), we can observe a less extreme heating strategy, which is close to the reactor

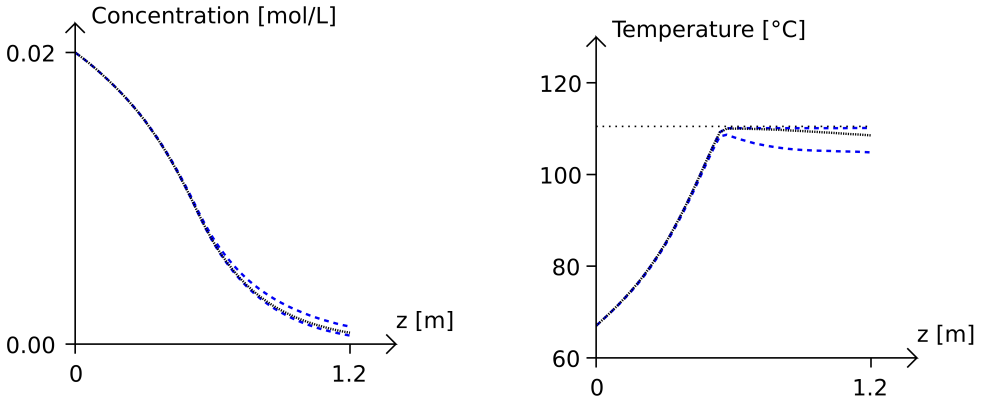


Figure 6.4: A simulation of the concentration C and temperature T (dotted lines) for 6% uncertainty in β applying the robustly optimized control input. It is guaranteed by Theorem 5.3, that the simulation result must be between the two dashed lines representing the outer ellipsoidal tube. However, the simulated concentration and temperature are in some parts of the reactor quite closed to their theoretical upper limits, i.e., the nonlinearity estimate was sufficiently accurate and did not introduce too much conservatism.

temperature as explained above. Note that in the nominally optimized case, the upper bound on the maximum jacket temperature is active while in the robustly optimized case the temperature is kept in a moderate range and is not driven to its bounds.

Finally, it remains to be discussed whether the computed robust solution is reasonable or much too conservative. Note that this is a relevant as we have used a non-trivial nonlinearity estimate, i.e., we can only qualitatively assess the level of conservatism: Figure 6.4 shows a simulation (dotted lines) of the nonlinear system applying the robustly optimized control input as well as a uncertain heat transfer coefficient whose values differ at most 6% from the nominal heat transfer coefficient β_{nominal}. As guaranteed by our theoretical result, the simulated states must be in the ellipsoidal outer tube shown as the dashed lines in Figure 6.4. However, we can also observe that the simulation drives the states quite close to their theoretical limits. The simulated temperature takes a maximum at $z^* \approx 0.6\,\text{m}$. At this point, we have $T_{\text{simulate}}(z^*) \approx 109.65°\text{C}$, which is obviously quite close to the upper limit $T_{\text{max}} = 110°\text{C}$. In this sense we may state that our nonlinearity estimate was sufficiently accurate and did not introduce an unreasonable amount of conservatism.

6.3 Robust Optimization of Periodic Systems

Let us start our analysis of periodic orbits by regarding an uncertain dynamic system of the form

$$\forall \tau \in \mathbb{R}: \quad \dot{x}(\tau) \;=\; f(\tau, x(\tau), w(\tau)) \;. \tag{6.3.1}$$

for a right-hand side function $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ which is assumed to be periodic in its first argument such that we have

$$\forall \tau \in \mathbb{R},\ \forall x \in \mathbb{R}^{n_x},\ \forall w \in \mathbb{R}^{n_w}: \quad f(\tau + T_{\mathrm{e}}, x, w) \;=\; f(\tau, x, w)$$

for some time $T_{\mathrm{e}} \in \mathbb{R}_{++}$. Here, we assume - as in the previous section - that f is uniformly Lipschitz continuous in x, while the uncertainty sets $W(\tau) \subseteq \mathbb{R}^{n_x}$ are compact and given. Moreover, we assume that we have $W(\tau + T_{\mathrm{e}}) = W(\tau)$ for all $\tau \in \mathbb{R}$, i.e., the uncertainty sets are periodic, too. Note that the corresponding set valued differential equation can be written as

$$\forall \tau \in \mathbb{R}_+: \quad X(\tau^+) \;=\; F(\tau, X(\tau), W(\tau)) \quad \text{with} \quad X(0) = X_0 \;, \tag{6.3.2}$$

where we assume that the initial uncertainty set X_0 at time 0 is given. In the following, we first review some standard definitions concerning the stability of periodic systems. However, this review must be interpreted as a preparation step for the discussion of periodic orbits of uncertain nonlinear dynamic systems which will follow in later sections. Note that we are in most of the practical applications rather interested in robustness than in stability. However, the stability of a dynamic system is in many practical situations a necessary requirement which enables us to make statements about robustness guarantees. The other way round, if we consider an unstable dynamic system there is in most of the practical situations not much hope that we can make any useful statements about robustness on an infinite time-horizon. This motivates to discuss stability aspects first.

Stability of Periodic Systems

In the following, we introduce some well-established standard definitions [36] which are needed for analyzing the stability of periodic orbits of uncertain differential equations. For this aim, we directly employ our notation of set-valued differential equations:

Definition 6.2 (Robust Stability of Periodic Orbits): *Let $x : \mathbb{R} \to \mathbb{R}^{n_x}$ be a periodic function with $x(\tau) = x(\tau + T_{\mathrm{e}})$, which satisfies the differential equation* (6.3.1) *for all functions w with $w(\tau) \in W(\tau)$ and for all $\tau \in \mathbb{R}$.*

1. *We say that the periodic orbit x is locally robustly stable, if there exists for every $\epsilon > 0$ a $\delta > 0$ such that we have for all $t \in \mathbb{R}_+$ and for all sets $X_0 \subseteq \mathbb{R}^{n_x}$ an implication of the form*

$$\sup_{\xi \in X_0} \|\xi - x(0)\| \leq \delta \quad \Longrightarrow \quad \sup_{\xi \in X(t)} \|\xi - x(t)\| \leq \epsilon \, .$$

   Here, we assume that the function X satisfies equation (6.3.2)*.*

2. *We say that the periodic orbit x is uniformly locally, asymptotically, and robustly stable, if it is locally robustly stable and if there exists an open set X_0 with $x(0) \in X_0$ such that the corresponding solution X of the propagation equation* (6.3.2) *satisfies*

$$\limsup_{t \to \infty} \sup_{\xi \in X(t)} \|\xi - x(t)\| \to 0 \, .$$

Note that there are many other stability definitions possible, as we could extend the above list with the notation of global stability, exponential stability etc., which are for example discussed by Blanchini and Miani [36], who also point out that the number of stability definitions increases with the number of possible permutations of the requirements (uniform, local v.s. global, asymptotic, robust, etc.). For the case that we have no uncertainties, i.e., $W(\tau) = \{0\}$ for all $\tau \in \mathbb{R}$, the above definition of local robust stability coincides with the notation of local stability for the nominal system and an analogous remark applies for the definition of local asymptotic stability.

One of the most famous tools to analyze stability properties of dynamic systems are Lyapunov functions, which can in our context be defined as follows:

Definition 6.3 (Lyapunov Functions): *Let x be a periodic orbit as in Definition 6.2. A locally Lipschitz continuous positive definite function $\Psi : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}_+$, which is periodic in its first argument, is said to be a (local) Lyapunov function, if there exists an open robust positive invariant tube $N : \mathbb{R} \to \Pi(\mathbb{R}^{n_x})$ with $x(t) \in N(t)$ such that the inequality*

$$\forall y \in N(t), \, \forall w \in W(t) : \quad \limsup_{h \to 0^+} \frac{\Psi(t, y + h f(t, y, w)) - \Psi(t, y)}{h} \quad \leq \quad -\alpha \left(\|y - x(t)\| \right)$$

holds for all $t \in \mathbb{R}$. Here, $\alpha : \mathbb{R} \to \mathbb{R}_+$ can be any continuous and strictly increasing function with $\alpha(0) = 0$.

It is a well-known fact [153] that a periodic orbit x is locally robustly stable inside the robust positive invariant tube N, if it admits a Lyapunov function Ψ. Note that the concept of Lyapunov functions and the concept of robust positive invariant tubes are strongly connected with each other [36]. For example, if we have a given Lyapunov function Ψ, we can construct non-trivial robust positive invariant tubes $\mathbb{X}_\epsilon : \mathbb{R} \to \Pi(\mathbb{R}^{n_x})$ by defining

$$\mathbb{X}_\epsilon(t) \;:=\; \{\, x \,|\, \Psi(t,x) \,\leq\, \epsilon \,\}$$

for all sufficiently small $\epsilon > 0$. Note that as long as the functions Ψ and W are periodic with respect to their time argument t, the functions $\mathbb{X}_\epsilon$, which are obtained by the above definition, are periodic, too. This observation implies the following statement:

Proposition 6.1: *The existence of a periodic Lyapunov function for a periodic orbit of an uncertain dynamic system implies the existence of periodic robust positive invariant tubes with non-empty interior.*

In the following, we plan to employ the result of Theorem 5.3 in order to construct conservative robust positive invariant tubes for periodic dynamic systems. Here, the aim is to derive sufficient stability conditions for dynamic systems under uncertainty. In order to make the corresponding techniques applicable for our current context of periodic systems we introduce the following Assumption:

Assumption 6.2: *Let x be a periodic orbit of the uncertain periodic differential equation* (6.3.1) *as introduced within Definition 6.2. We assume that we have an explicit nonlinearity estimate $\Omega_N : [0, T_\mathrm{e}] \times \mathbb{R}^{n_x} \times \mathbb{S}^{n_x}_+ \times \mathbb{R}^m_{++} \to \mathbb{S}^{n_x}_+$ which satisfies the requirements from Assumption 5.4, if the periodic orbit x is employed as the central path. Moreover, we assume that the following properties are satisfied:*

1. *The function Ω_N is periodic in its first argument with period time T_e, i.e we have a relation of the form $\Omega_N(t + T_\mathrm{e}, \cdot, \cdot, \cdot) \equiv \Omega_N(t, \cdot, \cdot, \cdot)$ for all $t \in \mathbb{R}$.*
2. *The function Ω_N satisfies for all $\tau \in [0, T_\mathrm{e}]$, all $Q_1, Q_2 \in \mathbb{S}^{n_x}_+$ with $Q_1 \preceq Q_2$, and all $\kappa \in \mathbb{R}^m_{++}$ a semi-definite inequality of the form*

$$\forall \alpha \in [0,1]: \quad \Omega_N(\tau, x(\tau), \alpha\, Q_1, \kappa) \;\preceq\; \alpha\, \Omega_N(\tau, x(\tau), Q_2, \kappa)\,.$$

The periodicity requirement in the above assumption is in most practical cases "automatically" satisfied, since the right-hand side function is periodic, too. Similarly, the second requirement in Assumption 6.2 does not add a major restriction, if we recall that

the function Ω_{N} is designed to over-estimate the nonlinear terms in the right-hand side equation such that Ω_{N} can typically be expected to grow at least linearly in Q. As we assume that x is a periodic orbit, we may additionally assume that we have

$$\forall \tau \in [0, T_{\mathrm{e}}]: \quad B(\tau) \;=\; \frac{\partial}{\partial w} f(\tau, x(\tau), 0) \;=\; 0$$

such that we have $\Omega_{\mathsf{total}} \;=\; \Omega_N$ and the matrix-valued right-hand side function Φ, which has originally been introduced within Definition 5.3, becomes in our case

$$\Phi(\tau, x, Q, \kappa) \quad = \quad A(\tau)Q + QA(\tau)^T + \sum_{i=1}^{m} \kappa_i Q + \Omega_{\mathrm{N}}(\tau, x, Q, \kappa)\;. \qquad (6.3.3)$$

Using this construction, we consider the following Theorem:

Theorem 6.2: *Let us assume that we have a nonlinearity estimate Ω_N for a periodic orbit x which satisfies Assumption 6.2 while the function Φ is given by equation* (6.3.3) *assuming $B(\tau) = 0$. Now, if there exists a function $\kappa : [0, T_{\mathrm{e}}] \to \mathbb{R}^m_{++}$, a scalar $\alpha \in [0, 1]$, and a symmetric positive definite function $Q : [0, T_{\mathrm{e}}] \to \mathbb{S}^{n_x}_{++}$ which satisfy the following differential equation together with the corresponding semi-definite boundary inequality*

$$\forall \tau \in [0, T_{\mathrm{e}}]: \qquad \dot{Q}(\tau) \;=\; \Phi(\tau, x(\tau), Q(\tau), \kappa(\tau)) \quad \textit{and} \quad \alpha\, Q(0) \;\succeq\; Q(T_{\mathrm{e}})\;, \qquad (6.3.4)$$

then x is a locally robustly stable periodic orbit with region of attraction $\mathcal{E}(Q(0), x(0))$. Moreover, if we can satisfy the above condition with $\alpha < 1$, then x is also asymptotically robustly stable within the mentioned region of attraction.

Proof: Let us first choose a function κ which satisfies the condition (6.3.4) on the time interval $[0, T_{\mathrm{e}}]$. This function can be continued periodically by defining $\kappa(nT_{\mathrm{e}} + t) := \kappa(t)$ for all $t \in [0, T_{\mathrm{e}}]$ and all $n \in \mathbb{N}$. Now, we regard the solution of the periodic differential equation

$$\forall \tau \in [0, \infty): \qquad \dot{Q}(\tau) \;=\; \Phi(\tau, x(\tau), Q(\tau), \kappa(\tau))\;.$$

Due to the two additional requirements on the function Ω_N, which have been introduced within Assumption 6.2, the solution Q of the above periodic differential equation must satisfy a semi-definite inequality of the form

$$\forall t \in [0, T_{\mathrm{e}}]: \quad Q(nT_{\mathrm{e}} + t) \quad \preceq \quad \alpha^n\, Q(t)\;,$$

which holds for all $n \in \mathbb{N}$. Now, we may use the result of Theorem 5.3 in order to show that we have for all $n \in \mathbb{N}$ and all $t \in [0, T_{\mathrm{e}}]$ an inclusion of the form

$$T(nT_{\mathrm{e}} + t)[\,\mathcal{E}(\,Q(0), x(0)\,)\,] \;\subseteq\; \mathcal{E}(\,\alpha^n\, Q(t), x(t)\,) \;\subseteq\; \mathcal{E}(\,\gamma\,\alpha^n\, Q(0), x(t)\,)\,.$$

Here, $\gamma < \infty$ is a uniform overshoot constant, which must exist as the function $Q(\cdot)$ is strictly positive and continuous on the compact interval $[0, T_{\mathrm{e}}]$. The statement of the Theorem is a direct consequence. □

Example 6.7: Let us regard an uncertain dynamic system of the form

$$\forall \tau \in \mathbb{R}: \quad \dot{x}(\tau) \;=\; (\,A(\tau) \,+\, C(\tau)E(\tau)D(\tau)\,)\; x(\tau)\,, \tag{6.3.5}$$

where the uncertainty $w(\tau) := \mathrm{vec}\,(\,E(\tau)\,)$ is assumed to satisfy $E(\tau)E(\tau)^T \preceq I$ while A, B, and C are periodic matrix valued functions with appropriate dimensions and period-time T_{e}. Here, the function $x(t) = 0$ is trivially a periodic orbit, but the important question is under which conditions we can guarantee that this orbit is robustly stable. Recall the nonlinearity estimate of the form

$$\Omega_{\mathsf{N}}(\,t,\, q(t),\, Q,\, \lambda\,) \;:=\; \frac{\sigma_{\mathsf{max}}\left(\,D(t)QD(t)^T\,\right)}{\lambda}\, C(t)C(t)^T\,,$$

which has been derived within Example 5.11. This nonlinearity estimate satisfies obviously the requirements from Assumption (6.2). Thus, we can apply Theorem 6.2 finding that the uncertain dynamic system (6.3.5) is (globally) robustly stable if there exists a function $\kappa : [0, T_{\mathrm{e}}] \to \mathbb{R}^{n_x}_{++}$ such that the differential equation

$$\begin{aligned} \dot{Q}(\tau) \;&=\; A(\tau)Q(\tau) + Q(\tau)A(\tau)^T + \kappa(\tau)Q(\tau) + \frac{\sigma_{\mathsf{max}}\left(\,D(\tau)Q(\tau)D(\tau)^T\,\right)}{\kappa(\tau)}\, C(\tau)C(\tau)^T \\ \alpha\, Q(0) \;&\succeq\; Q(T_{\mathrm{e}}) \end{aligned}$$

admits a positive definite solution $Q : [0, T_{\mathrm{e}}] \to \mathbb{S}^{n_x}_{++}$ for some $\alpha \in [0, 1]$. The latter sufficient condition for robust stability can be checked with standard optimal control solvers - although the formulation is in this form non-convex. Here, it should be highlighted that a condition of the above form can not so easily be obtained with the usual H_∞-framework which is well-established for linear time-invariant systems [51, 59, 82, 172, 244] and which has also been analyzed for computing the distance to instability [52, 53]. Note that even if the matrix valued functions A, C, and D are time-invariant, the above condition provides robust stability guarantees for time-varying uncertainties E.

Set Valued Periodic Systems

So far, we have concentrated on the case that there exists a single periodic orbit x of the uncertain differential equation for which we have derived sufficient stability conditions within Theorem 6.2. This result has some applications as illustrated in the example above. However, in most of the practical applications, we will typically not be able to find a single orbit x which satisfies the differential equation for all possible uncertainties. In this case, we are interested in the question whether we can find a possibly small periodic tube in which the state of the uncertain dynamic systems remains forever. In other words, we are interested in conditions which guarantee the existence of a set valued function $X : [0, T_{\mathrm{e}}] \to \Pi(\mathbb{R}^{n_x})$ which satisfies

$$\forall \tau \in [0, T_{\mathrm{e}}] : \quad X(\tau^+) \;=\; F(\tau, X(\tau), W(\tau)) \quad \text{as well as} \quad X(0) \;=\; X(T_{\mathrm{e}}) \;. \tag{6.3.6}$$

At this point, we have to rely on a quite non-trivial technical result, namely Schauder's fixed point theorem, which is well-known in the literature [45]. As this result will be the basis of the following consideration, we briefly summarize it in form of the following Lemma:

Lemma 6.1: *If we can find a convex and bounded set $Y \subseteq \mathbb{R}^{n_x}$ such that we have an inclusion of the form*

$$\boldsymbol{cl}\,(\, T(T_{\mathrm{e}}, 0)[Y]\,) \;\subseteq\; Y \;,$$

then there exists a periodic and robust positive invariant tube X which satisfies the condition (6.3.6) *as well as $X(0) \subseteq Y$.*

Proof: As the set propagation operator $T(T_{\mathrm{e}}, 0)$ is continuous on $\Pi(Y)$, the above statement is equivalent to the standard version of Schauder's fixed point theorem [45].□

The above way of guaranteeing the existence of periodic and robust positive invariant tubes is different than the strategy from Proposition 6.1, since Lemma 6.1 does not require the uncertain dynamic system to be stable in any sense. In fact, we might even argue that we are simply not interested in stability when uncertainties are present - as long as we can guarantee that the system remains within a small periodic tube. However, in many practical examples we observe that such a small periodic and robust positive invariant tube is easier to find if there exists at least a nominally stable periodic orbit. The aim of the following consideration is to develop numerical techniques which help us to find and optimize such periodic tubes.

Optimization of Periodic Robust Positive Invariant Tubes

The aim of this section is to develop a numerically tractable formulation which helps us to find and optimize periodic set valued orbits of uncertain differential equations. We still assume that the right-hand side function f is periodic with respect to its explicit time dependence, but we switch back our notation allowing that the associated set-valued propagation can be influenced by a control input u and a parameter p. The periodic robust optimal control problem of our interest takes now the following form:

$$\boxed{\begin{array}{lrcl} \min\limits_{u(\cdot),p,T_{\mathrm{e}},X(\cdot)} & \int_0^{T_{\mathrm{e}}} \mathcal{L}(\tau,u(\tau),T_{\mathrm{e}},X(\tau),W(\tau))\,\mathrm{d}\tau \;+\; M(\,p,T_{\mathrm{e}},X(T_{\mathrm{e}})\,) & & \\ \text{s.t.} & X(\tau^+) &=& F(\tau,u(\tau),p,X(\tau),W(\tau)) \\ & X(0) &=& X(T_{\mathrm{e}}) \\ & 0 &\geq& H(\tau,u(\tau),p,X(\tau),W(\tau)) \quad \text{for all } \tau\in[0,T_{\mathrm{e}}]\,. \end{array}} \tag{6.3.7}$$

Here, the constraint function H, and the Lagrange term $\mathcal{L}$ are assumed to be periodic (with period time T_{e}) with respect to their explicit time dependence. The rest of the notation is as in the previous sections.

Assumption 6.3: *We assume that we have functions φ, Φ with appropriate dimensions such that the following property is satisfied: for any given function $u:[0,T_{\mathrm{e}}]\to\mathbb{R}^{n_u}$, any vector $p\in\mathbb{R}^{n_p}$, and any function $\kappa:[0,T_{\mathrm{e}}]\to\mathbb{R}^{m}_{++}$, which admit solutions $q:[0,T_{\mathrm{e}}]\to\mathbb{R}^{n_x}$ and $Q:[0,T_{\mathrm{e}}]\to\mathbb{S}^{n_x}_{+}$ of the coupled differential equation*

$$\forall\tau\in[0,T_{\mathrm{e}}]:\quad \left\{\begin{array}{rcl} \dot{q}(\tau) &=& \varphi(\tau,u(\tau),p,q(\tau),Q(\tau),\kappa(\tau)) \\ \dot{Q}(\tau) &=& \Phi(\tau,u(\tau),p,q(\tau),Q(\tau),\kappa(\tau))\,, \end{array}\right.$$

the set valued function $\mathbb{X}(\cdot) := \mathcal{E}(Q(\cdot),q(\cdot))$ is a robust positive invariant tube on the interval $[0,T_{\mathrm{e}}]$. Here, φ and Φ are assumed to be time-periodic with respect to their explicit time-dependence.

The strategy is very similar to the previous section, i.e., we assume that Assumption 6.3 is satisfied and consider the auxiliary periodic optimal control problem, which is associated

with the above original problem formulation (6.3.7):

$$\begin{array}{l} \inf\limits_{\xi(\cdot),\zeta(\cdot),\pi,T_\mathrm{e}} \int_0^{T_\mathrm{e}} \mathcal{L}(\tau,u(\tau),T_\mathrm{e},\mathcal{E}(Q(\tau),q(\tau)),W(\tau))\,\mathrm{d}\tau + M(p,T_\mathrm{e},\mathcal{E}(Q(T_\mathrm{e}),q(T_\mathrm{e}))) \\ \text{s.t.} \quad \left\{ \begin{array}{rcll} \dot{q}(\tau) & = & \varphi(\tau,u(\tau),p,q(\tau),Q(\tau),\kappa(\tau)) & q(0) \;=\; q(T_\mathrm{e}) \\ \dot{Q}(\tau) & = & \Phi(\tau,u(\tau),p,q(\tau),Q(\tau),\kappa(\tau)) & Q(0) \;=\; Q(T_\mathrm{e})\;, \\ 0 & \geq & H(\tau,u(\tau),\,\mathcal{E}(Q(\tau),q(\tau)),\,W(\tau)) & \text{for all } \tau\in[0,T_\mathrm{e}]\;. \end{array} \right. \end{array} \tag{6.3.8}$$

For this auxiliary problem we can prove the following result:

Theorem 6.3: *Provided that Assumption 6.3 is satisfied, the following statements hold:*

1. *If the function H is component-wise monotonically increasing in $X(\tau)$, then every feasible input $(u(\cdot),p)$ of the auxiliary optimal control problem* (6.3.8) *corresponds to a feasible input of the original robust optimal control problem* (6.3.7)*.*

2. *If the function H is component-wise monotonically increasing in $X(\tau)$, while the objective functions $\mathcal{L}$ and M are monotonically increasing in $X(\tau)$ and $X(T_\mathrm{e})$, respectively, then the objective value of problem* (6.3.8) *is an upper bound on the objective value of the original problem* (6.3.7)*.*

3. *If the function Φ is constructed from a nonlinearity estimate Ω_N, which satisfies the requirements from Assumption 6.2, such that*

$$\Phi(\tau,x,Q,\kappa) \;=\; A(\tau)Q + QA(\tau)^T + \sum_{i=1}^{m}\kappa_i Q + \Omega_{\mathrm{total}}(\tau,x,Q,\kappa)\;,$$

and if the auxiliary problem 6.3.8 has a positive definite solution Q, then the central path q is a nominally stable periodic orbit.

Proof: The first two statements of the proof are in principle analogous to the considerations in the previous sections, in the sense that we can use the relation

$$T(t,0)[\mathcal{E}(Q(0),q(0))] \;\subseteq\; \mathcal{E}(Q(t),q(t))\;,$$

which holds for all $t\in\mathbb{R}_+$. However, a non-trivial part of the statement is that the above inclusion implies already the existence of periodic tubes which satisfy the feasibility condition

$$\forall\tau\in[0,T_\mathrm{e}]: \quad X(\tau^+) \;=\; F(\tau,u(\tau),p,X(\tau),W(\tau)) \quad \text{as well as} \quad X(0) \;=\; X(T_\mathrm{e})\;.$$

In order to show this, we have to use Lemma 6.1, which can be applied, since we have

$$\mathbf{cl}\,(\,T(T_{\mathrm{e}},0)[\,\mathcal{E}(Q(0),q(0))\,]\,)\;\subseteq\;\mathcal{E}(Q(0),q(0))\;.$$

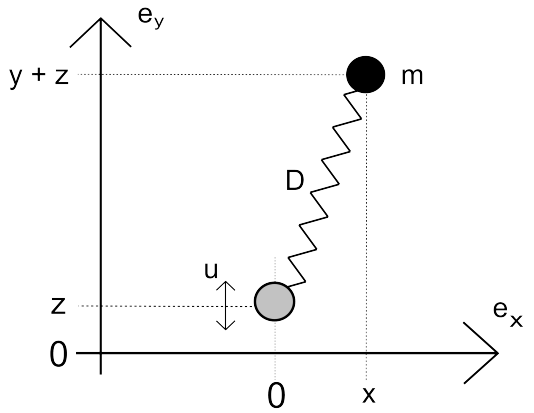
Thus, we may conclude the first two statements of the theorem. Finally, we remark that the third statement is a direct consequence of Theorem 6.2. □

6.4 Open-Loop Stable Orbits of an Inverted Spring Pendulum

In this section, we illustrate how the techniques from the previous section can be used to find open loop stable orbits for periodic systems in practice. In order to derive a simple but nonlinear model for an inverted spring pendulum in the 2-dimensional Euclidean space $\mathbb{R}^2$, we first introduce the mass m, which is attached at one end of a spring with given relaxed length l and spring constant D. The other end of the spring is mounted at a point, which can move along the vertical axis. We assume that this mounting point has at time t the coordinate $(0,z(t))^T$, while we can control the associated acceleration $u(t) := \ddot{z}(t)$. The velocity of the mounting point will be denoted by $v_z(t) := \dot{z}(t)$. Moreover, we assume that the position of the mass point is given by $(x(t),z(t)+y(t))^T$, i.e., (x,y) is the relative position coordinate of the mass with respect to the moving oscillatory base. Note that Figure 6.5 shows a sketch of this construction as well as the numerical values for the given physical constants.

The associated equations of motion of the form $\dot{\xi}(t) = f(\xi(t),u(t),w(t))$ can easily be derived with Newton's law by employing standard assumptions on the friction-, spring-, and gravitational forces, which act at the mass point. Here, we summarize the states of our dynamic system within one vector $\xi := (x,y,v_x,v_y,z,v_z)$ such that the right-hand side function f becomes

$$f(\xi,u,w) \;=\; \begin{pmatrix} v_x \\ v_y \\ -\frac{Dx}{m}\left(1-\frac{l}{\sqrt{x^2+y^2}}\right) - bv_x + w \\ -g + u - \frac{Dy}{m}\left(1-\frac{l}{\sqrt{x^2+y^2}}\right) - bv_y \\ v_z \\ u \end{pmatrix}. \tag{6.4.1}$$



Parameter	Value
L	$1\,\mathrm{m}$
m	$0.1\,\mathrm{kg}$
D	$700\,\frac{\mathrm{N}}{\mathrm{m}}$
g	$9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}$
b	$5\,\frac{1}{\mathrm{s}}$
$\overline{w}$	$0.03\,\frac{\mathrm{N}}{\mathrm{kg}}$
$\overline{u}$	$200\,\frac{\mathrm{N}}{\mathrm{kg}}$
$\overline{v}_z$	$3.2\,\frac{\mathrm{m}}{\mathrm{s}}$

Figure 6.5: A sketch of the inverted spring pendulum showing the choice of coordinates as well as all given numerical values for the parameters which are needed within the problem formulation.

In this context, the function w is assumed to be a model uncertainty, which could e.g. be due to an uncertain and time-varying force, which acts at the mass point in a horizontal direction. The associated uncertainty set is in our example assumed to be a simple interval of the form $W(\tau) := [-\overline{w}, \overline{w}]$.

Our aim is to operate the spring pendulum in an open-loop stable periodic orbit with period time $T_\mathrm{e} \in \mathbb{R}_{++}$ at its "inverted" position. For this aim, we suggest to minimize the time-average over the maximum displacement of the mass point in x-direction, i.e., we introduce a generalized Lagrange term of the form

$$\mathcal{L}(\tau, u(\tau), T_\mathrm{e}, X(\tau), W(\tau)) \;:=\; \max_{\xi \in X(\tau)} \frac{\left(e_x^T \xi\right)^2}{T_\mathrm{e}} \quad \text{with} \quad e_x^T \;:=\; (1, 0, \ldots, 0)^T \in \mathbb{R}^6 \,.$$

Finally, the constraint function H is in our example used to formulate simple bounds on the control input u as well as on the velocity v_z of the mounting point:

$$H(\tau, u(\tau), X(\tau)) \;:=\; \begin{pmatrix} u(\tau) - \overline{u} \\ -u(\tau) + \overline{u} \\ \max\limits_{\xi \in X(\tau)} e_{v_z}^T \xi - \overline{v}_z \\ \min\limits_{\xi \in X(\tau)} -e_{v_z}^T \xi + \overline{v}_z \end{pmatrix} \quad \text{with} \quad e_{v_z}^T \;:=\; (0, \ldots, 0, 1)^T \in \mathbb{R}^6 \,.$$

Here, the values for these bounds are all given in Figure 6.5 such that we have all ingredients which are needed within the problem formulation (6.3.7) remarking that we do not have a Mayer term in this example while the period time $T_\mathrm{e} > 0$ is a free optimization variable.

In order to construct a conservative but tractable formulation for this optimal control problem, we need to find a suitable nonlinearity estimate. For this aim, we first observe that only the third and fourth component of the right-hand side function (6.4.1) include nonlinear terms. In order to over-estimate the influence of these terms, we first define the terms

$$l_3(q,Q) \;:=\; \frac{Dl}{m}\frac{\sqrt{Q_{33}Q_{44}}}{q_4(q_4-\sqrt{Q_{44}})} + \frac{1}{2}\frac{Dl}{m}\frac{(Q_{33})^{\frac{3}{2}}}{q_4(q_4-\sqrt{Q_{44}})^2} \tag{6.4.2}$$

$$l_4(q,Q) \;:=\; \frac{Dl}{m}\frac{Q_{33}}{(q_4-\sqrt{Q_{44}})^2}\,, \tag{6.4.3}$$

which are designed to overestimate the nonlinear terms in third an fourth component of the right-hand side function f such that the nonlinearity estimate becomes:

$$\Omega_{\mathsf{N}}(\,\tau,\,q,\,Q,\,u,\,\lambda\,) \;:=\; \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{l_3(q,u,Q)^2}{\lambda_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l_4(q,u,Q)^2}{\lambda_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Now, we have all ingredients which are needed to setup the robust periodic optimal control problem of the form (6.3.7) and to approximately solve it based on a formulation of the form (6.3.8). Here, we mention that the Lagrange term can be evaluated as

$$\mathcal{L}(\tau,u(\tau),T_{\mathrm{e}},\mathcal{E}(Q(\tau),q(\tau)),W(\tau)) \;:=\; \max_{\xi\in\mathcal{E}(Q(\tau),q(\tau))} \frac{\left(e_x^T\xi\right)^2}{T_{\mathrm{e}}} \;=\; \frac{Q_{3,3}(\tau)}{T_{\mathrm{e}}}\,.$$

Note that the problem of the form (6.3.8) requires in this example 6 differential states to implement the dynamics of the central path $q:[0,T_{\mathrm{e}}]\to\mathbb{R}^6$ as well as 36 differential states for the associated nonlinear differential equation for $Q:[0,T_{\mathrm{e}}]\to\mathbb{R}^{6\times 6}$. However, we can still reduce the number of states by using that the matrix valued function is symmetric and that the states z and v_z are not affected by the uncertainties such that in total only a differential equation with $6+\frac{5\cdot 4}{2} \;=\; 16$ states has to be implemented. Collecting the control inputs, we need one primal control input u, which denotes the acceleration of the oscillatory base, and 3 dual control inputs $\kappa\in\mathbb{R}^3$ to optimize the estimate of the influence the uncertainty itself and the nonlinear terms respectively. Thus, we need 4 control inputs in total.

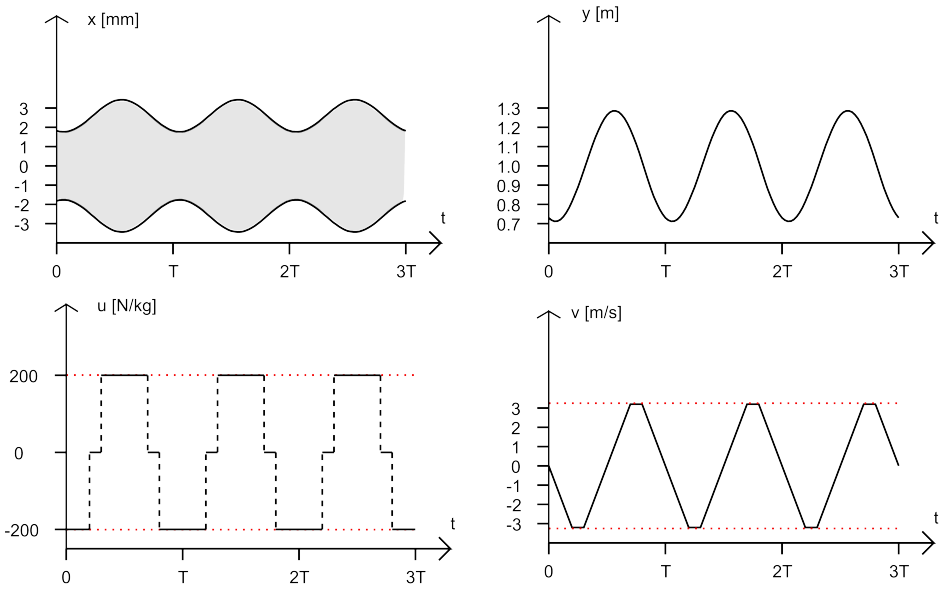


Figure 6.6: The upper left part of the figure shows a projection of the optimized periodic robust positive invariant tube $\mathcal{E}(Q(\cdot))$ onto the $t-x$-plane. Here, the grey shaded area represents the region in which the horizontal displacement $x(t)$ of the mass point against the vertical axis can be guaranteed to be. The upper right part of the figures shows the optimal central path of the y-coordinate of the mass point. Finally, in the lower left part of the figure, we can find the optimal control input over three periods while the associated vertical velocity profile of the mounting point is shown in the lower right part.

Remark 6.2: *Note that the existence of open-loop stable periodic orbits of the inverted spring pendulum is well-known in the literature. As early as in 1908 Stephenson has predicted this phenomenon [224]. For a more recent article we refer to the work of Arinstein and Gitterman [10], where the open-loop stable orbits of an inverted spring pendulum are theoretically analyzed with an approximation technique using Mathieu's differential equation [238]. In this thesis, we have used this existing approximate analysis to find a good initial guess for the optimal control algorithm. In addition, we refer to the work of Kabamba, Meerkov, and Poh [136] on stability and robustness in vibrational control, where similar periodically operated dynamic systems are discussed from a control perspective.*

A locally optimal and robustly open-loop stable periodic orbit is visualized in Figure 6.6. This orbit has been found by solving the above problem formulation numerically using

ACADO Toolkit (cf. Chapter 7). The optimal value for the cycle duration is in this example $T_{\mathrm{e}} \approx 79\,\mathrm{ms}$. Note that for a sinusoidal driving force at the oscillatory base the resonance frequency of the spring would be

$$\omega_{\mathsf{r}} \;=\; \omega_0 \sqrt{1 - \frac{1}{2}\left(\frac{b}{\omega_0}\right)^2} \quad \text{with} \quad \omega_0 \;:=\; \sqrt{\frac{D}{m}}\,.$$

If we use the parameters from Figure 6.5, we find a corresponding resonance cycle duration of

$$T_{\mathsf{r}} \;=\; \frac{2\pi}{\omega_{\mathsf{r}}} \;\approx\; 75\,\mathrm{ms}\,,$$

at which a maximum nominal oscillation amplitude of the spring in y-direction can be expected. Note that the value for T_{r} is close to our optimal result for T_{e}. The interpretation of this effect is that we need a significant amplitude of the spring oscillation in order to obtain stability – at least, if there would be no oscillation of the spring, the pendulum cannot possibly be stable at its inverted position. Thus, from a physical point of view, it is clear that we have to choose a driving frequency which is close to resonance. On the other hand, if the system is exactly at resonance, it might be more sensitive with respect to disturbances. In this sense the numerical result for the time T_{e} is in sound with our physical expectation. Also note that the optimized control input has a bang-bang structure, which is in our example affected by the bounds on the velocity of the oscillatory base, too.

Finally, we highlight once more that the considered robust optimal control formulation yields a guarantee for the region in which the horizontal position x of the nonlinear pendulum will remain. The corresponding projection of the computed robust positive invariant tube onto the (t, x)-plain is shown as the grey shadowed region in the upper right part of Figure 6.6. This statement holds for all disturbance inputs which satisfy $w(\tau) \in [-\overline{w}, \overline{w}]$ for all times $\tau \in \mathbb{R}$. Moreover, if there are no uncertainties, i.e., for the case $w = 0$, we can make the following statement: whenever we start the dynamic system inside the computed robust positive invariant tube, it will be attracted by the periodic orbit until it swings in its the nominal orbit at the inverted position. In other words, we have a guarantee on the region of attraction, which has been proven in Theorem 6.3.

Part III

Software & Applications

Chapter 7

ACADO Toolkit – Automatic Control and Dynamic Optimization

In this chapter, the software ACADO Toolkit is presented [245], which is based on a joint development effort together with my colleague Hans Joachim Ferreau and our supervisor Prof Moritz Diehl. ACADO Toolkit is an optimal control software, which has been developed for solving general nonlinear optimal control problems. It provides also tailored algorithms for special classes of optimal control problems such as parameter estimation problems [38, 207], model predictive control problems [4, 132, 197], multi-objective optimal control problems [157, 168], as well as robust optimization problems for dynamic systems, which are the focus of this thesis. Note that this software has been the basis for all the numerical results which are presented in this thesis. Especially the numerical results for the robust optimization of the tubular reactor from Section 6.4 as well as the stability optimization for the inverted spring pendulum from Section 6.2 would not have been possible without the algorithms which are implemented in ACADO. Note that the following overview sections about the software ACADO have been published in [131].

7.1 Introduction

The last decades have seen a rapidly increasing number of applications where control techniques based on dynamic optimization lead to improved performance. These techniques use a mathematical model in form of differential equations of the process to be controlled

to predict its future behavior and calculate optimized control actions. This optimization can be performed once, offline, before the runtime of the process resulting in optimized open-loop controller. Alternatively, the optimization can be performed, online, during the runtime of the process in order to obtain a feedback controller. In both cases, the numerical solution of optimal control problems is the main algorithmic step. Thus, efficient and reliable optimization algorithms for performing this step are of great interest.

Review of Existing Optimal Control Software

Searching the literature, we can can find a number of optimization algorithms which have been implemented for solving optimal control problems. We only discuss some of the most common packages: Let us start the list with the open-source package IPOPT [232, 233], originally developed by Andreas Wächter and Larry Biegler, which implements an interior point algorithm for the optimization of large scale differential algebraic systems. It can be combined with collocation methods for the discretization of the continuous dynamic system while a filter strategy is implemented as a globalization technique. IPOPT is written in C/C++ and Fortran, but uses modeling languages like AMPL or MATLAB in order to provide a user interface and to allow automatic differentiation.

Furthermore, a MATLAB package named PROPT [248] receives more and more attention. PROPT is a commercial tool, developed by the Tomlab Optimization Inc.. PROPT solves optimal control problems based on collocation techniques, while using existing NLP solvers such as KNITRO, CONOPT, SNOPT or CPLEX. Due to the MATLAB syntax, the package PROPT appears user-friendly – at the price that it is not open-source.

Recently, an open-source optimal control code has been published by Brian C. Fabien [87] under the name dsoa. This package is written in C/C++ and discretizes differential algebraic systems based on implicit Runge-Kutta methods. Unfortunately, the package does only implement single-shooting methods, which is often not advisable for nonlinear optimal control problems. On the optimization level sequential quadratic programming techniques are employed.

Similar to dsoa, the proprietary package MUSCOD-II, originally developed by Daniel Leineweber [151], is suitable for solving optimal control problems. MUSCOD-II discretizes the differential algebraic systems based on BDF or Runge Kutta integration methods and uses Bock's direct multiple shooting technique [43]. Sequential quadratic programming is used for solving the resulting NLPs. The algorithms implemented in MUSCOD-II are

written in C/C++ and Fortran and based on advanced algorithmic strategies – in particular because multiple shooting is used instead of single shooting. MUSCOD-II implements highly efficient code and its algorithmic concepts [43, 40, 150, 151] were an important source of inspiration for the development of ACADO Toolkit .

Finally, software packages dedicated to nonlinear model predictive control in the process industry exist, like OptCon [216], or NEWCON [201], which are both based on multiple shooting.

The Concept of the ACADO Toolkit

The ACADO Toolkit has been designed to be a freely available open-source optimal control package [245]. It is distributed under the GNU Lesser General Public License (LGPL), which allows the user to link the package against proprietary software. At the current status there are direct optimal control methods implemented, which are mainly based on single- and multiple shooting. For this aim, ACADO provides tailored Runge Kutta as well as BDF integrators [15] in order to discretize dynamic systems. These integrators can also compute first and second order sensitivities of the state trajectory with respect to external control inputs or parameters based on internal numeric or internal automatic differentiation (IND/IAD) [39]. On the optimization level, specialized sequential quadratic and sequential convex programming methods (SQP/SCP) are implemented which exploit the particular structures arising in the context of multiple shooting [150]. These optimization algorithm are also tailored for parametric optimization problems which arise for example in the context of model predictive control, moving horizon estimation, and multi-objective optimization as elaborated within Section 7.2, where we briefly describe the classes of optimization problems to which ACADO Toolkit can be applied. Depending on the objective, several SQP implementations are available which can for example be based on exact Hessians, tailored Block-BFGS updates, or Gauss-Newton Hessian approximations.

Besides the efficient implementation of optimal control algorithms, ACADO Toolkit has a special emphasis on user-friendliness. The aim is to provide a syntax which allows the user to state optimal control problems in a way that is very close to the usual mathematical syntax. For experienced users this might only be a question of convenience. However, given the fact that dynamic optimization is more and more widely used in many different engineering applications, also non-experts should be able to formulate their control problems within a reasonable period of time. The ACADO Toolkit makes intensive use of the object-oriented capabilities of C++ in order to come up with powerful symbolic

tools which do not only allow the convenient setup of optimal control problems from the user perspective, but which also allow automatic differentiation, symbolic manipulations, optimized C-code export, as well as auto-detection routines, which recognize the structure of the problem formulation which is then exploited within the algorithms. The symbolic tools can be seen as the basis of ACADO which make the tool unique in comparison to existing software packages, as explained within Section 7.3, where also the algorithmic features as well as the main underlying software modules of ACADO Toolkit are motivated and described in more detail. Tutorial examples illustrating the use of ACADO Toolkit's syntax are given in Section 7.4.

7.2 Problem Classes Constituting the Scope of the Software

ACADO Toolkit highlights three important problem classes. The first problem class are offline dynamic optimization problems, where the aim is to find an open-loop control which minimizes a given objective functional. The second class are parameter and state estimation problems, where parameters or unknown control inputs should be identified by measuring an output of a given nonlinear dynamic system. The third class are combined online estimation and model predictive control problems, where parameterized dynamic optimization problems have to be solved repeatedly to obtain a dynamic feedback control law.

Optimal Control Problems

One of the basic problem classes which can be solved with ACADO Toolkit are standard optimal control problems. These problems typically consist of a dynamic system with differential states $x : \mathbb{R} \to \mathbb{R}^{n_x}$, an optional time varying control input $u : \mathbb{R} \to \mathbb{R}^{n_u}$, and time constant parameters $p \in \mathbb{R}^{n_p}$. In some cases the formulation of the dynamic system requires also algebraic states, which we denote by $z : \mathbb{R} \to \mathbb{R}^{n_z}$. The standard formulation of an optimal control problem is shown in Figure 7.1.

For standard optimal control problems the objective functional Φ is typically a Bolza functional of the form

$$\Phi[x(\cdot), z(\cdot), u(\cdot), p, T] = \int_{t_0}^{T} L(\tau, x(\tau), z(\tau), u(\tau), p, T)\,\mathrm{d}\tau \;+\; M(x(T), p, T)\;. \quad (7.2.1)$$

A general optimal control problem formulation (OCP):

$$
\begin{array}{lll}
\underset{x(\cdot),z(\cdot),u(\cdot),p,T}{\text{minimize}} & \Phi[x(\cdot), z(\cdot), u(\cdot), p, T] & \\
\text{subject to:} & & \\
\forall t \in [t_0, T]: \quad 0 = f(t, \dot{x}(t), x(t), z(t), u(t), p, T) & & \text{(OCP)} \\
0 = r(x(0), z(0), x(T), z(T), p, T) & & \\
\forall t \in [t_0, T]: \quad 0 \geq s(t, x(t), z(t), u(t), p, T) & &
\end{array}
$$

Example for an optimal control problem formulation implemented with ACADO:

```
#include <acado_toolkit.hpp>

int main( ){

    DifferentialState       x;              // a differential state
    AlgebraicState          z;              // an algebraic state
    Control                 u;              // a control
    Parameter               p;              // a parameter
    DifferentialEquation    f;              // a differential equation

    f << dot(x) == -0.5*x-z+u*u;            // example for a differential-
    f <<      0 ==  z+exp(z)+x-1.0+u;       // algebraic equation.

    OCP ocp( 0.0, 4.0 );                    // OCP with t_0 = 0.0 and T = 4.0
    ocp.minimizeMayerTerm( x*x + p*p );     // a Mayer term to be minimized

    ocp.subjectTo( f );                     // OCP should regard the DAE
    ocp.subjectTo( AT_START, x     == 1.0 ); // an initial value constraint
    ocp.subjectTo( AT_END  , x + p == 1.0 ); // an end (or terminal) constraint

    ocp.subjectTo( -1.0 <= x*u <= 1.0 );    // a path constraint

    OptimizationAlgorithm algorithm(ocp);   // define an algorithm
    algorithm.solve();                      // to solve the OCP.

    return 0;
}
```

Figure 7.1: A general mathematical formulation of an optimal control problem and a tutorial code example for an implementation with ACADO.

The algorithms, which are currently implemented in `ACADO Toolkit` assume that the right-hand side function f is smooth or at least sufficiently often differentiable depending on which specific discretization method is used. Moreover, we assume that the function $\frac{\partial f}{\partial(\dot{x},z)}$ is always regular, i.e., the index of the DAE should be one. The remaining functions, namely the Lagrange term L, the Mayer term M, the boundary constraint function r, as well the path constraint function s are assumed to be at least twice continuously differentiable in all their arguments.

Note that Figure 7.1 shows next to the general mathematical formulation also an ACADO implementation example. This example demonstrates how the natural syntax of the toolkit can be used to implement and solve standard optimal control problems.

Note that some parts of the above formulation are from a mathematical point of view redundant: For example a Mayer term can always be formulated as a Lagrange term and vice versa by introducing slack variables. Also the time horizon T and the constant parameter p could be omitted in the formulation above as they can always be eliminated by introducing auxiliary differential states. However, from a numerical point of view it makes sense to use as much structure as possible, such that the above formulation seems natural. Finally, we mention that optimal control problems contain standard nonlinear programs (NLPs) by leaving away the constraint "`ocp.subjectTo(f)`".

Parameter and State Estimation

An important class of optimal control problems, which requires special attention, are state and parameter estimation problems. This subclass of optimal control problems has also the form (OCP). However, as it will be explained in Section 7.3, parameter estimation problems with least-square objective terms can be treated with a specialized algorithm known under the name generalized Gauss-Newton method. Thus, in the case of a general parameter estimation problem the objective functional Φ takes the form:

$$\Phi[x(\cdot), z(\cdot), u(\cdot), p, T] \quad = \quad \sum_{i=0}^{N} \| \, h_i(t_i, x(t_i), z(t_i), u(t_i), p) - \eta_i \|^2_{S_i} \, .$$

Here, h is called a measurement function while $\eta_1, \ldots, \eta_N$ are the measurements taken at the time points $t_1, \ldots, t_N \in [0, T]$. Note that the least-squares term is in this formulation weighted with positive semi-definite weighting matrices $S_1, \ldots, S_N$, which are typically the inverses of the variance covariance matrices associated with the measurement errors. In ACADO the syntax

```
ocp.minimizeLSQ( S, h, eta );
```

can be used to define least-square objectives.

Model Based Feedback Control

Model based feedback control constitutes the third main problem class that can be tackled with ACADO Toolkit. It comprises two kinds of online dynamic optimization problems: the Model Predictive Control (MPC) problem of finding optimal control actions to be fed back to the controlled process, and the Moving Horizon Estimation (MHE) problem of estimating the current process states using measurements of its outputs. The MPC problem is a special case of an (OCP) for which the objective takes typically the form:

$$\Phi[x(\cdot), z(\cdot), u(\cdot), p, T] \;=\; \int_{t_0}^{T} \|y(t, x(t), z(t), u(t), p) - y_{\mathsf{r}}\|_S^2 + \|y^{\mathsf{e}}(x(T), p) - y_{\mathsf{r}}^{\mathsf{e}}\|_R^2 \, .$$

Therein, y_{r} is a tracking reference for the output function y and $y_{\mathsf{r}}^{\mathsf{e}}$ a reference for a terminal-weight. The matrices S and R are symmetric and positive semi-definite weighting matrices with appropriate dimensions. In contrast to OCPs, MPC problems are assumed to be formulated on a fixed horizon T and employing the above tracking objective function.

In case not all differential states of the process can be measured directly, an estimate has to be obtained using an online state estimator. This is usually done by one of the many Kalman filter variants or by solving an MHE problem. The MHE problem has basically the same form as a parameter estimation problem. Both, the MPC and the MHE problem are solved repeatedly, during the runtime of the process, to yield a model and optimization based feedback controller.

Note that throughout this chapter, the terms MPC and MHE are used for both the linear-quadratic as well as for the general nonlinear case.

7.3 Software Modules and Algorithmic Features

We now discuss details of the software design of ACADO Toolkit and describe its main software modules. Along with that we highlight a couple of algorithmic features, in particular the functionality to handle symbolic expressions, that give rise to ACADO Toolkit's unique capabilities.

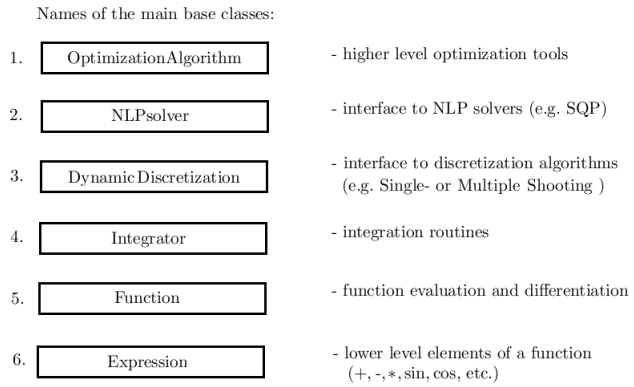


Figure 7.2: The main algorithmic base classes of ACADO Toolkit

The Basic Structure of the Toolkit

The basic structure of ACADO is outlined in Figure 7.2. In this figure, the six most important base classes are shown. Starting at the bottom of the figure, a lower level interface for elementary operations is provided. Classes like "Addition", "Multiplication", etc. inherit from their base class with the name "Expression. This class structure is used to build up function evaluation trees. Note that the functionality of this low-level part of ACADO will be explained below.

On the next main level, the base class "Function" is introduced. Functions can for example consist of symbolic expression trees, or linked C-code, or user-written model specifications, etc. However, the main concept of this base class is that higher level algorithms - e.g. integration routines - do not need to know what happens inside, i.e., they can evaluate or differentiate a function independent of whether a symbolic expression tree or a C-function is evaluated in the background. The functions in ACADO automatically tell the higher level algorithms whether they provide automatic differentiation.

While the base class "Integrator" is an interface for any kind of integration routines the class "DynamicDiscretization" organizes the discretization techniques in the context of optimal control techniques. Note that the class "Integrator" can also be used as a base to interface external integration routines. Again, the class "DynamicDiscretization" hides the specific mode of discretization in a generic way, i.e., for example an SQP algorithm does not need to know, whether a differential equation is discretized by collocation or by a shooting method. Moreover, the class "DynamicDiscretization" could also be used to

interface a PDE discretization tool. Note that this form of modularity is organized in such a way that the efficiency is not affected, i.e., an optimization method can always ask the discretization modules for the details, if additional information needs to be passed via suitable data-structures.

On the higher level NLP solvers can be interfaced via the base class "NLPsolver". NLP solvers are used in the "OptimizationAlgorithm" which auto-selects and initializes the algorithmic sub-modules. Specific implementations of the "OptimizationAlgorithm" inherit from this base class providing tailored drivers for the selected algorithms. For example the class "RealTimeAlgorithm" inherits from "OptimizationAlgorithm" and implements drivers for e.g. running an SQP method with real-time iterations. Finally, we mention that all the above classes can also be used stand-alone as demonstrated and explained within the tutorial codes that come with ACADO. In this sense, users and developers can choose at which part and also at which level of abstraction they want to use or extend the ACADO toolkit.

Symbolic Expressions

One of the fundamental requirements for an optimal control package is that functions such as objectives, right-hand sides of differential equations, constraint function etc. can be provided by the user in a convenient manner. One way to achieve this is that the user links, for example, a simple `C` function. However, `ACADO Toolkit` implements more powerful features: The idea is to use symbolic expressions as a base class to build up complex model equations by making extensive use of the `C++` class concept as well as operator overloading. The benefit of this way of implementing functions is that e.g. automatic detection of dependencies and dimensions, automatic as well as symbolic differentiation, convexity detection etc. are available.

In order to explain this concept, we consider the `ACADO` tutorial code Listing 7.1. Compiling and running this simple piece of code with a standard `C++` compiler linking `ACADO` shows – as expected – that the dimension of the defined function `f` is two and that it depends on one differential state. Moreover, the convexity of the components of `f` is recognized. Note that these auto-detection routines are typically needed by developers. In most situations, the only remaining work for a user is to define his/her function, while the dimension, structure etc. can be detected by the algorithms we want to use. Due to operator overloading the syntax can be used as if we would write standard `C/C++` code.

Listing 7.1: Dimension and convexity detection for symbolic functions

```
int main( ){

    DifferentialState x;
    IntermediateState z;
    TIME              t;
    Function          f;

    z = 0.5*x + 1.0     ;

    f << exp(x) + t     ;
    f << exp(z+exp(z)) ;

    printf("the dimension of f is %d   \n",  f.getDim() );
    printf("f depends on  %d  states   \n",  f.getNX () );
    printf("f depends on  %d  controls \n",  f.getNU () );

    if( f.isConvex() == BT_TRUE )
        printf("all components of function f are convex. \n");

    return 0;
}
```

For example, the intermediate variable `z` in Listing 7.1 would only be evaluated once if we evaluate `f` at a given point, i.e., the symbolic expressions behave as expected.

For a complete overview of the features that are implemented, we refer to the manual [130], where also a lot of commented tutorial codes can be found. Here, we only briefly outline some of the features:

- **Automatic differentiation:** The symbolic notation of functions enables us to provide not only numeric- but also automatic- and symbolic differentiation. The automatic differentiation [33, 109, 110] is implemented in its forward as well as in the adjoint mode for first and (mixed) second order derivatives. Moreover, all expressions can symbolically be differentiated returning again an expression, like AD with source code transformation. This functionality can be used recursively leading to arbitrary orders of symbolic differentiation. Note that the class "Function" in Figure 7.2 does not necessarily need to evaluate symbolic expression trees, it is also possible to link a C-function as well as evaluation routines for the corresponding directional derivatives. Thus, it is also possible to link existing AD packages such as ADOL-C [110]. However, using the built-in ACADO routines avoids unnecessary overhead.

Listing 7.2: The definition of a linear function with ACADO

```
Matrix                  A(3,3);
Vector                  b(3);
DifferentialStateVector x(3);
Function                f;

A.setZero() ;
A(0,0) = 1.0;  A(1,1) = 2.0;  A(2,2) = 3.0;
b(0)   = 1.0;  b(1)   = 1.0;  b(2)   = 1.0;

f << A*x + b;
```

- **Convexity detection:** As we have already illustrated in the example code, functions can be tested for convexity/concavity. The corresponding algorithmic routines are based on disciplined convex programming [106]. Note that the syntax for the routines is (almost) the same as in the MATLAB package CVX [107]. As the ACADO code is C++ based, the convexity detection is in general faster than MATLAB. However, this is minor advantage in the sense that convexity detection is typically only used as a pre-processing tool.

- **Code optimization:** In the context of optimal control algorithms, right-hand side functions are typically evaluated many times. Thus, it is efficient to pre-optimize functions internally during the initialization phase. In the ACADO Toolkit this pre-optimization is automatically done. For example if a linear function f is defined by the code piece in Listing 7.2, we would expect that a single evaluation of the function f at a given vector x$\in \mathbb{R}^3$ would involve 18 flops: 9 multiplications and 9 additions, as the matrix-vector product A*x with the matrix A$\in \mathbb{R}^{3\times 3}$ together with the addition of the vector b$\in \mathbb{R}^3$ requires this complexity. However, ACADO Toolkit auto-detects the zero entries in the matrix A, which is in this example diagonal, such that the evaluation of f costs only 6 flops – 3 multiplications and 3 additions. The price that we have to pay for this internal code optimization is that the "loading" of the functions takes longer, which is however usually a worthwhile investment of computation time, if f is evaluated very often. Finally, it remains to be mentioned that ACADO Toolkit would in this case also detect that f is linear.

- **C Code Generation:** Writing a model function within the ACADO notation does not mean to go into a one way street. A symbolic function can later also be exported in form of (optimized) standard C code. This is especially interesting for model predictive controllers where the time for one function evaluation can be crucial.

In [132] the idea of automatically generated C-codes is employed to export highly efficient real-time Gauss-Newton methods, for which the presented symbolic tools in ACADO are used.

Integration Algorithms

For the optimization of dynamic systems based on single or multiple shooting methods [43] it is necessary to simulate differential or differential-algebraic equations. In addition, sensitivities of the state trajectory with respect to initial values, control inputs, etc. must be provided. For this aim, ACADO Toolkit comes along with state of the art integration routines such as several step-size controlled Runge-Kutta methods as well as a BDF (backward differentiation formula) method which is used for stiff differential or differential algebraic equations. Note that the ACADO BDF integrator, which is based on the algorithmic ideas in [11, 15, 186], can also deal with fully implicit differential algebraic equations of index 1, which have the form

$$\forall t \in [0,T]: \quad F(\dot{y}(t), y(t), u(t), p, T) = 0\ . \tag{7.3.1}$$

Here, differential and algebraic states are merged into one state vector y, while u, p, and T are defined as in Section 7.2. However, note that in most optimal control problems arising in practice the right-hand side F is linear in $\dot{y}$. Moreover, the ACADO BDF integrator uses a diagonal implicit Runge-Kutta starter in order to avoid too small steps taken by the multi-step method at the beginning of each multiple shooting interval.

All integrators provide first and second order differentiation techniques in order to compute sensitivities of the state trajectory with respect to initial values and control/parameter inputs. Here, the differentiation can either be based on internal numerical differentiation [39, 15] or on (internal) automatic differentiation. However, for automatic differentiation, the right hand side functions must be provided in the ACADO syntax, i.e., in form of the class Function. For the case that plain C++ or MATLAB functions are linked the expression for the Jacobian should be provided, too. Otherwise, numeric differentiation will be used.

Finally, it should be mentioned that the integration routines that are currently implemented within the ACADO Toolkit are very similar to existing integrator packages like Sundials [247] or DAESOL [15] with respect to both the algorithmic strategies as well as the performance. In order to provide consistent and self-contained C++ code, the integration routines have been implemented in cooperation with the class Function,

which detects for example the sparsity patterns of the right-hand side functions. In the current release, ACADO Toolkit provides the possibility to use sparse linear algebra solver (e.g. the solver C-Sparse [64]) within the BDF integrator.

Note that also a stand-alone sub-package ACADO Integrators is available [245] that also provides an elaborate MATLAB interface.

Discretization of Dynamic Systems

Once an integrator for dynamic systems is avaliable, the original continuous optimal control problem (OCP) can be discretized. Here, several strategies can be applied. The most simple strategy is to regard the simulation of the system as a function evaluation depending on the initial values, parameters, control inputs, etc.. The corresponding discretization method is known under the name single shooting. In ACADO Toolkit not only single shooting but also multiple shooting methods are implemented, which have turned out to out-perform single shooting methods in many cases [43, 150]. In multiple shooting methods, the whole time interval is divided into several multiple shooting intervals on each of which the dynamic system is discretized using an integrator.

As an alternative to multiple shooting, collocation methods have attracted a lot of attention during the last decades [31, 233]. Here, the dynamic system is discretized at the level of the NLP leading to quite large and sparse nonlinear programs. In ACADO Toolkit, collocation methods are actually under development and will be released in the near future.

Nonlinear Optimization Algorithms

Once a dynamic system can be discretized, the optimal control problems that have been introduced in Section 7.2 can be transformed into nonlinear programs (NLPs). The mathematical standard form of such NLPs is

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \Phi(x) \\ \text{subject to} & G(x) = 0 \\ & H(x) \leq 0 \end{array} \tag{7.3.2}$$

Note that in the optimal control context, the discretized NLP has a certain structure. For the case that multiple shooting is used for the discretization, ACADO Toolkit exploits the structure via condensing techniques that are based on the ideas in [43, 150]. In order

to solve the usually nonlinear NLPs, state of the art optimization algorithms are needed. Currently, ACADO Toolkit provides several SQP-type methods that can e.g. be based on BFGS Hessian approximations, as described in [190], or on Gauss-Newton methods [38]. In addition, line search globalization routines [190, 115] as well as auto-initialization techniques are implemented to make the optimization routines as reliable as possible. In case an underlying quadratic program (QP) becomes infeasible during the SQP iterations, all QP constraints are automatically relaxed using slack variables that are ℓ_1-penalized in the objective function. A tutorial code explaining how these optimization tools can be used will be discussed in Section 7.4. Note that collocation methods combined with interior point techniques, as e.g. described in [31], are not yet supported in the current release of ACADO Toolkit.

However, although the ACADO Toolkit comes along with its own optimization routines, it is designed to be extended with existing implementations of optimization algorithms. The software design, which makes use of well-established C++ interface concepts such as abstract base classes and inheritance, allows to use ACADO Toolkit as a test and implementation platform for new developments. For example, in the current implementation the plain C++ code qpOASES [246] is linked as default QP solver. Thus, ACADO Toolkit is not only designed as a high-end tool for solving optimal dynamic optimization and control problems but also as a framework that can be filled and extended in many ways.

Real-Time Iterations

As mentioned in Section 7.2, MPC problems can from a pure algorithmic optimization perspective be interpreted as a special kind of optimal control problems. In particular, they depend parametrically on the current initial value x_0 of the process. This special property is exploited within the ACADO Toolkit by applying the real-time iteration scheme presented in [69, 73]. It builds on a direct multiple shooting discretization and only performs one SQP-type iteration using a Gauss-Newton Hessian approximation per feedback loop.

The computations in each iteration are divided into a long "preparation phase", in which the system linearization, possible elimination of algebraic variables and condensing of the linearized subproblem are performed, and a much shorter "feedback phase" that just solves one condensed quadratic program. This feedback phase can be orders of magnitude shorter than the preparation phase. In the case of a linear process model, the real-time iteration scheme gives the same feedback as a linear MPC controller. Error bounds and

closed-loop stability of the scheme have been established for nonlinear MPC with shifted and non-shifted initializations in [76] and [75].

7.4 Tutorial Examples and Numerical Tests

In this section we discuss two code examples: the first one implements a simple time optimal control problem while the second one explains how to set up a simple state and parameter estimation problem with `ACADO`. Tutorials for closed-loop simulations using real-time iterations can be found on the `ACADO Toolkit` web-site [245]. Moreover, the efficiency of the ACADO implementation of real-time iterations is demonstrated in [91], where a online simulation of a kite system is discussed. Note that ACADO is currently designed for systems with 10 to 100 states [91, 157]. For large scale systems new algorithmic features need to be added or external optimization packages can be linked.

An Introductory Optimal Control Problem

In this section it is explained how to setup a simple optimal control problem using the `ACADO Toolkit`. The aim of this tutorial is to solve an example problem of the form:

$$\begin{array}{ll} \underset{s(\cdot),v(\cdot),m(\cdot),u(\cdot),T}{\text{minimize}} & T \\ \text{subject to:} & \\ \forall t \in [0,T]: & \dot{s}(t) = v(t) \\ & \dot{v}(t) = \frac{u(t) - 0.2\,v(t)^2}{m(t)} \\ & \dot{m}(t) = -0.01\,u(t)^2 \\ & s(0) = 0\ ,\ v(0) = 0\ ,\ m(0) = 1 \\ & s(T) = 10\ ,\ v(T) = 0 \\ & -0.1 \leq v(t) \leq 1.7 \\ & -1.1 \leq u(t) \leq 1.1 \\ & 5 \leq T \leq 15 \end{array} \quad . \tag{7.4.1}$$

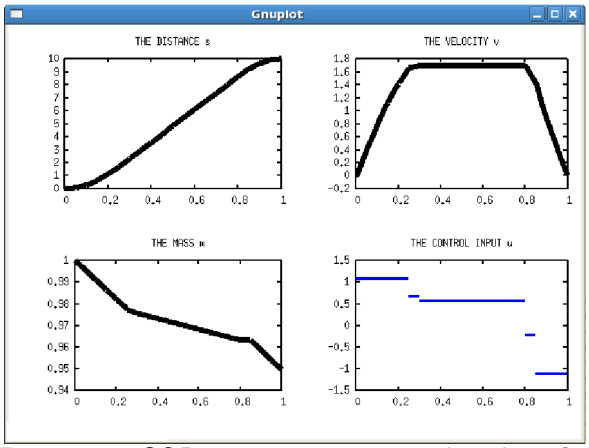
This problem is based on a simple free-space rocket model with 3 states: the distance s, the velocity v, and the mass m of the rocket. The aim is to fly in minimum time T from

Listing 7.3: An implementation of the optimal control problem (7.4.1)

```
int main( ){

    DifferentialState       s,v,m     ;   // the differential states
    Control                 u         ;   // the control input u
    Parameter               T         ;   // the time horizon T
    DifferentialEquation    f( 0.0, T );   // the differential equation

//  ------------------------------------
    OCP ocp( 0.0, T );                     // time horizon of the OCP: [0,T]
    ocp.minimizeMayerTerm( T );            // the time T should be optimized

    f << dot(s) == v;                      // an implementation
    f << dot(v) == (u-0.2*v*v)/m;          // of the model equations
    f << dot(m) == -0.01*u*u;              // for the rocket.

    ocp.subjectTo( f                    );  // minimize T s.t. the model,
    ocp.subjectTo( AT_START, s ==  0.0 );   // the initial values for s,
    ocp.subjectTo( AT_START, v ==  0.0 );   // v,
    ocp.subjectTo( AT_START, m ==  1.0 );   // and m,

    ocp.subjectTo( AT_END  , s == 10.0 );   // the terminal constraints for s
    ocp.subjectTo( AT_END  , v ==  0.0 );   // and v,

    ocp.subjectTo( -0.1 <= v <=  1.7   );   // as well as the bounds on v
    ocp.subjectTo( -1.1 <= u <=  1.1   );   // the control input u,
    ocp.subjectTo(  5.0 <= T <= 15.0   );   // and the time horizon T.
//  ------------------------------------

    OptimizationAlgorithm algorithm(ocp);   // the optimization algorithm
    algorithm.solve();                      // solves the problem.

    return 0;
}
```

$s(0) = 0$ to $s(T) = 10$, while constraints on the velocity v and the control input u should be satisfied. The rocket starts with velocity $v(0) = 0$ and should stop at the end time T, which can be formulated in form of the constraint $v(T) = 0$.

The corresponding ACADO code, which solves the above optimal control problem numerically, can be found in Listing 7.3. In this example, we do not specify the NLP solver explicitly, but the class `OptimizationAlgorithm` chooses by default a multiple shooting discretization method with 20 multiple shooting and control intervals in combination with an SQP algorithm. For the integration, a Runge Kutta solver with order 4 and error control order 5 is chosen. Please note that in this code example no initialization is specified. Here, an



#:	KKT tol.	Obj. Value
1:	1.001e+03	1.000e+01
2:	5.766e+00	9.950e+00
3:	2.946e−02	9.932e+00
4:	7.481e−02	9.906e+00
.	.	.
.	.	.
.	.	.
12:	8.740e−04	7.442e+00
13:	3.308e−07	7.442e+00

convergence achieved.

Figure 7.3: SQP iteration output and a plot of the optimal results for problem (7.4.1).

auto-initialization routine has been implemented, which works well for optimal control problems that are either not too nonlinear or convex. Otherwise an initialization can for example be provided in form a simple txt-file or as a matrix containing an initial guess for the optimal solution.

In order to visualize the results, a user-friendly Gnuplot interface is available, whose use is outlined in the numerous tutorial examples coming along with the `ACADO Toolkit`. A Gnuplot screenshot together with the output of the iterations taken by the SQP method is shown in Figure 7.3.

Comparing the implementation in Listing 7.3 with the corresponding mathematical problem (7.4.1) the syntax can quite intuitively be understood. Note that the dimensions of the problem as well as the dependencies have been auto detected. In addition, the structure of the problem is exploited by the numerical algorithm: for example the control constraints of the form

$$-1.1 \;\leq\; u(t) \;\leq\; 1.1$$

are internally detected as bounds. In contrast, e.g. a general constraint of the form

$$u(t)^2 + u(t) \;\leq\; 1.1$$

would have been more expensive as the derivative of the constraint function needs to be evaluated during the SQP iterations. Moreover, the bounds are efficiently used within the QP solver, which is needed during the SQP iterations. Note that all these types of auto-detection routines are a major advance in comparison to most other existing optimal control packages in terms of user-friendliness and automatic generation of efficient code.

A Tutorial Parameter Estimation Problem

Similar to the standard optimal control problem case from the last section, we discuss in this section a tutorial which explains how parameter and state estimation problems can be formulated and solved within `ACADO Toolkit`. For this aim, we consider the problem

$$\boxed{\begin{array}{ll} \underset{\phi(\cdot),\alpha,l}{\text{minimize}} & \sum_{i=1}^{10} \left(\phi(t_i) - \eta_i\right)^2 \\ \text{subject to:} & \\ \forall t \in [0,T]: & \ddot{\phi}(t) = -\frac{g}{l}\phi(t) - \alpha\dot{\phi}(t) \\ & 0 \leq \alpha \leq 4 \\ & 0 \leq l \leq 2 \end{array}} \quad . \tag{7.4.2}$$

Here, a simple pendulum model is regarded, which consists of the state ϕ representing the excitation angle; variable $\dot{\phi}$ denotes the angular velocity. The constant $g = 9.81$ is the gravitational constant while the friction coefficient α and the length l of the cable are only known to lie between certain bounds. We assume that the state ϕ has been measured at several times.

In Listing 7.4 a tutorial code is shown which solves problem (7.4.2) numerically. Note that the data file, which is read by the routine, is shown in the left part of Figure 7.4. Here, 2 of the 10 measurements were not successful leading to "`nan`" entries in the data file. Moreover, the measurements have not been taken on a equidistant time grid. Nevertheless, the `ACADO` code which solves the above parameter estimation problem is easily set up and deals automatically with the non-equidistant measurements and with the failures in the measurement data.

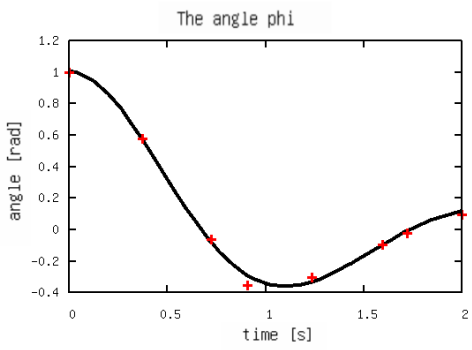
Note that the parameter estimation algorithm chooses by default a Gauss-Newton SQP method using the structure of the least-squares objective. In the output of the method the result for the parameter estimation is displayed in the form which is shown in the lower right part of Figure 7.4. Note that the computation of the standard deviations of the parameter estimates is based on a linear approximation in the optimal solution as proposed in [38].

Listing 7.4: An implementation of the parameter estimation problem (7.4.2)

```
int main( ){

    DifferentialState           phi, dphi;    // the states of the pendulum
    Parameter                   l, alpha ;    // its length and the friction
    const double                g = 9.81 ;    // the gravitational constant
    DifferentialEquation        f        ;    // the model equations
    Function                    h        ;    // the measurement function

//  ---------------------------------------
    OCP ocp( 0.0, 2.0 )                  ;    // construct an OCP
    h << phi                             ;    // the state phi is measured
    ocp.minimizeLSQ( h, "data.txt" )     ;    // fit h to the data

    f << dot(phi ) == dphi               ;    // a symbolic implementation
    f << dot(dphi) == -(g/l) * sin( phi )    // of the model
                      -alpha * dphi      ;    // equations

    ocp.subjectTo( f                    );    // solve OCP s.t. the model,
    ocp.subjectTo( 0.0 <= alpha <= 4.0  );    // the bounds on alpha
    ocp.subjectTo( 0.0 <=   l   <= 2.0  );    // and the bounds on l.
//  ---------------------------------------

    ParameterEstimationAlgorithm algorithm(ocp); // the parameter estimation
    algorithm.solve();                           // solves the problem.

    return 0;
}
```

ACSII file "data.txt" containing the measurements:

TIME POINTS	MEASUREMENTS
0.00000e+00	1.00000e+00
2.72321e−01	nan
3.72821e−01	5.75146e−01
7.25752e−01	−5.91794e−02
9.06107e−01	−3.54347e−01
1.23651e+00	−3.03056e−01
1.42619e+00	nan
1.59469e+00	−9.64208e−02
1.72029e+00	−1.97671e−02
2.00000e+00	9.35138e−02



The fitting results:

```
I     = 1.001e+00 +/- 1.734e-01
alpha = 1.847e+00 +/- 4.059e-01
```

Figure 7.4: Data file containing the measurements as well as the fitting results obtained by the Gauss-Newton method applied to problem (7.4.2).

Chapter 8

An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC

In this chapter we present an automatic C-code generation strategy for real-time nonlinear model predictive control (NMPC), which is designed for applications with kilohertz sample rates. The corresponding code export module has been implemented within the software package `ACADO Toolkit` (cf. Chapter 7). It is capable of exporting fixed step-size integrators together with their sensitivities as well as a real-time Gauss-Newton method. Here, we employ the symbolic representation of optimal control problems in `ACADO` in order to auto-generate plain C-code which is optimized for final production. The exported code has been tested for model predictive control scenarios comprising constrained nonlinear dynamic systems with four states and a control horizon of ten samples. The numerical simulations show a promising performance of the exported code being able to provide feedback in much less than a millisecond. Note that the following sections are based on a journal article [132], which has been accepted for publication in Automatica.

8.1 Introduction

A recent trend in the field of convex optimization goes into the direction of automatic code export leading to automatically generated and customized interior point solvers.

These optimized solvers have proven to be real-time feasible for online optimization with a sampling time in the microsecond range [166]. The advantages of code generation are at first place the efficiency of the exported plain C-code. In addition, auto-generated code can increase the reliability as all memory can be made static and conditional jumps can be mostly avoided. In addition, plain and self-contained C-code can easily be compiled for PC-like embedded hardware and possibly also for field-programmable gate arrays (FPGAs).

If a process model is derived from first-principle physical laws, we often end up with a nonlinear dynamic system, for which convex optimization techniques can typically not be applied. Although convex optimization covers a wide range of applications [21, 46], it can usually not directly deal with nonlinear dynamics which often arise if the process model is derived from first-principle physical laws. In order to be able to deal with non-linear dynamics in the control context, nonlinear model predictive control (NMPC) algorithms are a well-known tool [5, 32, 79]. The idea to use code generation for NMPC has been introduced by Ohtsuka in form of the tool AutoGenU [183]. It exports C-code, and uses a continuation Newton method for the optimality system. At each sampling instant one linear system has to be solved with a GMRES algorithm. Computation times of 1.5 ms per iteration have been reported for an experimental hovercraft setup [210]. Another approach is the advanced step NMPC controller [242] which, however, solves a full nonlinear program at each sampling instant. Yet a different approach is the nonlinear real-time iteration (RTI) scheme [68, 73]. Like the previous approaches, it uses a similar continuation Newton-type framework for which nominal stability has been shown [75] but solves one QP at each iteration. This allows for multiple active set changes and thus ensures that the nonlinear MPC algorithm cannot perform worse than a linear MPC controller. An overview of existing algorithms for fast nonlinear MPC can be found in [74].

The RTI scheme has originally been developed for large scale chemical engineering applications. The aim of this chapter is to demonstrate that NMPC algorithms based on the RTI scheme can be optimized and auto-generated efficiently aiming at sampling times in the milli- and micro-second range. In order to allow for these ultra-fast execution times, we reduce the algorithmic components of the nonlinear real-time iteration scheme [73] to the absolute minimum. This allows us to auto-generate optimized C-code that is suitable for ultra-fast computation and export onto embedded hardware. Here, a symbolic representation of optimal control problems is indispensable as this allows for efficient dependency and sparsity detection as well as automatic differentiation and code export. Consequently, the open-source software `ACADO Toolkit` [131] – which is based on symbolic optimal control problem formulations – is a natural framework for the implementation of C-code export tools for nonlinear model predictive control.

In Section 8.2 we introduce the algorithmic core of the real-time iteration scheme, while Section 8.3 explains the newly developed `ACADO Code Generation` tool. In Section 8.4 we demonstrate the performance of the implemented tools.

8.2 The Real-Time Iteration Algorithm for Nonlinear Optimal Control

Throughout this chapter we are interested in nonlinear optimal control problems of the form

$$\begin{array}{ll} \underset{\xi(\cdot),\zeta(\cdot)}{\min} & \int_0^T \left(\|\xi(\tau)\|_2^2 + \|\zeta(\tau)\|_2^2 \right) \mathrm{d}\tau \\ \text{s.t.} & \dot{\xi}(t) \;=\; f(\,\xi(t),\,\zeta(t)\,) \\ & \xi(0) \;=\; \xi_0 \\ & \underline{z} \;\leq\; \zeta(t) \;\leq\; \overline{z} \quad \text{for all } \; t \in [0,T]\;. \end{array} \tag{8.2.1}$$

Here, $\xi : \mathbb{R} \to \mathbb{R}^n$ denotes the state, $\zeta : \mathbb{R} \to \mathbb{R}^m$ the control input, and $\underline{z}, \overline{z} \in \mathbb{R}^m$ the control bounds. The right-hand side function f can be non-linear in both states and controls, while the objective is a least-squares tracking term with $\|\cdot\|_2$ denoting the Euclidean norm. In the context of nonlinear MPC, $\xi_0 \in \mathbb{R}^n$ is the current state measurement.

A more general optimal control problem formulation would also include non-autonomous dynamics, time-varying tracking references, a weighting in the objective, a quadratic Mayer term penalizing $\xi(T)$, non-linear state and control constraints, zero terminal constraints, etc. The proposed algorithms and software implementations can deal with all these issues as illustrated within the examples in Section 8.4. However, we prefer to keep our presentation simple and work for the moment with the formulation (8.2.1) which can later be generalized.

Recall that direct methods for optimal control [32, 68] proceed in two steps: first, we discretize the problem. And second, we solve a finite dimensional nonlinear program.

Discretization of the Optimal Control Problem

For the discretization of the nonlinear dynamics several options exist. Collocation methods [32] directly represent the states as polynomials with a finite number of coefficients.

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Figure 8.1: A suitable Butcher tableau for a Runge-Kutta integrator with order 4 for fixed, pre-optimized step-sizes.

Alternatively, single- or multiple shooting discretization methods [38, 152, 68] can be employed where an integrator is used in order to simulate the dynamic system. In this chapter, we concentrate on single- and multiple shooting techniques which use in the simplest case a piecewise constant control discretization

$$\zeta(t) \approx \sum_{i=1}^{N} z_i I_{[t_i,t_{i+1})}(t) \ ,$$

where $I_{[a,b)}(t)$ is equal to 1 if $t \in [a,b)$ and equal to 0 otherwise. The time sequence $0 = t_1 < t_2 < \ldots < t_{N+1} = T$ can e.g. be equidistant. We define

$$z := \left(z_1^T, \ldots, z_N^T\right)^T \in \mathbb{R}^{n_z}$$

with $n_z := Nm$ to achieve a convenient notation.

Let us regard the solution $\xi(t_{i+1})$ (with $i \in \{1, \ldots, N\}$) of the differential equation

$$\forall \tau \in [t_i, t_{i+1}] : \ \dot{\xi}(\tau) = f(\xi(\tau), z_i) \ \text{ and } \ \xi(t_i) = x_i$$

as a function $\Xi_i(x_i, z_i) = \xi(t_{i+1})$ depending on the discrete control input z_i and on the multiple shooting node x_i which is the initial value for the i-th control interval. Here, the operator Ξ_i is the solution operator of the differential equation, which can numerically be evaluated by using an integrator. As a reasonable step-size choice can often be pre-optimized before run-time, we suggest to employ a standard Runge-Kutta method using a constant step-size once the integrator is running in online mode. Note that a Runge-Kutta integrator whose Butcher tableau contains many zero entries can be beneficial. For example, exploiting the four zero-entries of the Butcher tableau of order 4 shown in Figure 8.1 reduces the computational load of each integration step by about one third.

We discretize the continuous least-squares objective as

$$\int_0^T \Big(\|\xi(\tau)\|_2^2 + \|\zeta(\tau)\|_2^2 \Big) \, \mathrm{d}\tau \;\approx\; \|F(x,y,z)\|_2^2$$

where $F(x,y,z) := \Big(y^T,\, x^T,\, z^T \Big)^T$. Here, the vector $x := \left(x_1^T, \ldots, x_N^T\right)$ summarizes the multiple shooting nodes but the initial value $y := x_0$. Moreover, we denote the multiple-shooting residual as

$$G(x,y,z) \;\; := \;\; \begin{pmatrix} x_1 - \Xi(y, z_1) \\ x_2 - \Xi(x_1, z_2) \\ \vdots \\ x_N - \Xi(x_{N-1}, z_N) \end{pmatrix} . \tag{8.2.2}$$

Now, we can summarize the result of the multiple-shooting discretization as

$$\begin{array}{ll} \underset{x,y,z}{\min} & \|F(x,y,z)\|_2^2 \\ \text{s.t.} & y \;=\; \xi_0 \\ & 0 \;=\; G(x,y,z) \\ & \underline{z} \;\leq\; z \;\leq\; \overline{z}\,. \end{array} \tag{8.2.3}$$

This large but sparse nonlinear program must be solved in real-time and for changing measurement inputs ξ_0. The next section explains this in more detail.

Real-Time Iteration Algorithm

In order to solve least-squares NLPs of the form (8.2.3), generalized Gauss-Newton methods, as originally proposed in [38], have turned out to perform very well in practice. An offline full-step version of this method starts from an initial guess (x^0, y^0, z^0) and generates iterates of the form $x^+ = x + \Delta x$, $y^+ = y + \Delta y$ and $z^+ = z + \Delta z$ where $(\Delta x, \Delta y, \Delta z)$ solves the convex QP

$$\begin{array}{ll} \underset{\Delta x, \Delta y, \Delta z}{\min} & \| \, F + F_x \Delta x + F_y \Delta y + F_z \Delta z \, \|_2^2 \\ \text{s.t.} & y + \Delta y \;=\; \xi_0 \\ & G + G_x \Delta x + G_y \Delta y + G_z \Delta z \;=\; 0 \\ & \underline{z} \leq\; z + \Delta z \;\leq\; \overline{z}\,. \end{array} \tag{8.2.4}$$

Real Time Iterations for Nonlinear MPC:

Initialization: Choose initial values for (x, y, z).

Repeat Online:

1) Evaluate F, G and $F_{x,y,z}, G_{x,y,z}$ at (x, y, z).
2) Perform the condensing, i.e., compute R_y, R_z, R.
3) Wait for the measurement ξ_0.
4) Compute Q, i.e., perform the initial value embedding step (8.2.7).
5) Solve the condensed QP (8.2.8).
6) Send the control input z_1^+ immediately to the process.
7) Update $(x, y, z) \leftarrow (x^+, y^+, z^+)$ and shift the time.

Figure 8.2: An illustration of the real-time iteration scheme.

Here, we have introduced the following short hands:

$$F := F(x, y, z)\,, \quad F_x := \partial_x F(x, y, z) \quad \text{etc.}$$

Although the Gauss-Newton method converges in general only linearly[1] to a local minimizer (x^*, y^*, z^*) of the problem (8.2.3), it can perform very well in practice if either the least-squares residual is small or if the function G is only mildly non-linear [38].

In the context of model predictive control, the above method is separated into a preparation and a feedback step [73], as the current measurement ξ_0 might not yet be available when we start solving the problem (8.2.3). Following the real-time iteration idea, as originally proposed in [73], we separate the algorithmic strategy into a preparation and a feedback step. In the preparation step, we evaluate F and G, compute the associated sensitivities F_x, F_y, F_z and G_x,G_y,G_z and perform a condensing step without knowing ξ_0 yet. Here, condensing refers to the computation of

$$\begin{aligned} R_y := F_y - F_x G_x^{-1} G_y\,, \quad R_z := F_z - F_x G_x^{-1} G_z\,, \\ \text{and} \qquad\qquad R := F - F_x G_x^{-1} G\,. \end{aligned} \tag{8.2.5}$$

[1] We assume that F and G are differentiable with Lipschitz continuous Jacobians, while the reduced Jacobian R_z from equation (8.2.5) is assumed to have always full column-rank.

This means that the sparse quadratic problem (8.2.4) is reduced to a smaller QP of the form

$$\begin{array}{ll} \min\limits_{\Delta y, \Delta z} & \| R_y \Delta y + R_z \Delta z + R \|_2^2 \\ \text{s.t.} & y + \Delta y \;=\; \xi_0 \\ & \underline{z} \leq \; z_i + \Delta z_i \;\leq\; \overline{z} \, . \end{array} \tag{8.2.6}$$

Once the measurement ξ_0 is available, we perform an initial value embedding step, i.e., we compute the matrix vector product

$$Q := R_y \, (\xi_0 - y) + R \tag{8.2.7}$$

constructing a dense and convex QP of the form

$$\min_{\Delta z} \quad \| R_z \Delta z + Q \|_2^2 \quad \text{s.t.} \quad \underline{z} \leq \; z_i + \Delta z_i \;\leq\; \overline{z} \tag{8.2.8}$$

which has only the input sequence Δz as a remaining degree of freedom. Solving this small and dense QP with a suitable QP solver completes the feedback step. Once its solution z^+ is available, we apply the first control z_1^+ to the process, shift the time horizon of the MPC, and perform the next preparation step. Figure 8.2 illustrates this real-time iteration idea in form of a pseudo code. For a mathematical foundation of this method including stability theorems, we refer to [4, 68, 75, 242]. Please note that the above algorithm can also be transferred to the single shooting discretization if we use $x_{i+1} := \Xi(x_i, z_i)$ for all $i \in \{0, \ldots, N\}$. In this case we have always $G(x, y, z) = 0$ during the iteration, which implies $R = F$.

Limitations of the Real-Time Iteration Scheme

When using the above nonlinear MPC algorithm in real-world applications, the following issues may lead to a failure of the algorithm:

Infeasibility: If formulation (8.2.1) additionally comprises state constraints or non-convex control constraints, two types of infeasibility can occur: first, the nonlinear online optimization problem itself can be infeasible. And second, the underlying quadratic programming problem within the SQP-type algorithm can become infeasible.

The first type of infeasibility is a general problem of NMPC algorithms. A possible remedy is the use of zero-terminal constraints and to solve the online optimization problem in every

step exactly. Assuming that the first optimization problem including the zero-terminal constraint is feasible, that no uncertainties occur, and that exact state measurements are available, it can be shown that all online optimization problems remain feasible [197].

For handling the second type of infeasibility suitable strategies exist [60]. Note that QP infeasibility cannot occur if only control bounds $\underline{z} \leq z \leq \overline{z}$ are present.

Instability: Even if we assume that the online optimization problems all remain feasible, there are two reasons why the closed-loop system can become unstable. First, the closed-loop system can be unstable, as we have a finite horizon only. This problem can be addressed by using suitable end weights or zero terminal constraints [197]. Second, we might leave the region of contraction of the Gauss-Newton method, e.g. due to a large disturbance. This must be avoided employing suitable globalization strategies if necessary. Note that for the above real-time iteration scheme, local asymptotic closed-loop stability can be guaranteed assuming suitable end weights in combination with other regularity conditions [68, 72, 75].

8.3 The ACADO Code Generation Tool

The `ACADO Toolkit` is an open-source software tool for automatic control and dynamic optimization [131]. The aim of this section is to explain the newly developed `ACADO Code Generation` tool which makes use of the symbolic features of `ACADO` to export optimized C-code. Here, we follow an idea from [166], where automatic generation of C-code for convex optimization was suggested, and extend it to nonlinear dynamic systems.

Symbolic Representations of MPC Formulations

Let us consider a simple example for a nonlinear function defined as

$$f(\phi, \omega) \; := \; -g \sin(\phi) - a \cos(\phi) - b\omega \, . \tag{8.3.1}$$

Once we implement this function in plain C, we can evaluate it for a given input. However, for example the fact that f is affine in ω can not explicitly be detected if f is given in form of a standard C-function. In order to overcome this limitation, `ACADO Toolkit` represents functions in a symbolic form as illustrated in Figure 8.3.

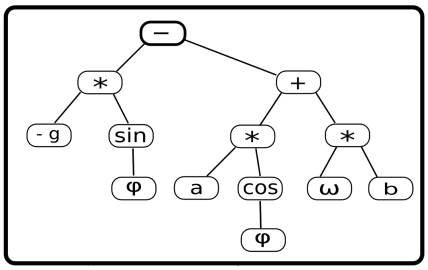


Figure 8.3: The operator based tree-representation of the example function (8.3.1) as used in the software `ACADO Toolkit`.

This enables us for example to compute the derivative of f with machine precision using automatic differentiation or to detect the zero-entry in the Jacobian of f with the aim to generate highly efficient C-code.

In `ACADO` this concept of symbolic representation is employed to define the whole MPC optimization problem (cf. Figure 8.4). In this example, we define a least squares objective of the form

$$\int_0^T \left[\xi(\tau)^T \mathcal{Q} \xi(\tau) + \zeta(\tau)^T \mathcal{R} \zeta(\tau)\right] \, \mathrm{d}\tau \ ,$$

where $\mathcal{Q}$ and $\mathcal{R}$ are not necessarily unit matrices. The differential equation `f` in the tutorial code would in a mathematical notation be given by

$$\begin{aligned} \dot{p}(t) &= v(t) \\ \dot{v}(t) &= a(t) \\ \dot{\phi}(t) &= \omega(t) \\ \dot{\omega}(t) &= -g \sin(\phi(t)) - a(t) \cos(\phi(t)) - b\omega(t) \ , \end{aligned} \tag{8.3.2}$$

where $\xi = (p, v, \phi, \omega)^T$ is the state and $\zeta = a$ the control. The control bounds have the form $-1 \leq a(t) \leq 1$.

For a more detailed documentation and for further tutorials on how to specify more general MPC formulations in `ACADO` we refer to [245, 91].

```
#include <acado_toolkit.hpp>

int main( ){

// INTRODUCE THE VARIABLES:
// -----------------------
    DifferentialState     p;  // setup four
    DifferentialState     v;  // differential states
    DifferentialState   phi;
    DifferentialState omega;
    Control               a;  // setup control input

    DifferentialEquation  f;  // setup an ODE
    double T = 3.0;           // length of time horizon

    Matrix Q = eye(4);        // weighting matrix Q
    Matrix R = eye(1);        // weighting matrix R

// SETUP THE MPC FORMULATION:
// -------------------------
    OCP ocp( 0, T );          // construct an optimal
    ocp.minimizeLSQ( Q, R );  // control problem (OCP)
                              // with tracking objective

    f << dot(p)     == v;     // define four
    f << dot(v)     == a;     // ODE equations
    f << dot(phi)   == omega;
    f << dot(omega) == -g*sin(phi)-a*cos(phi)-b*omega;

    ocp.subjectTo( f );       // set model equations
    ocp.subjectTo( -1 <= a <= 1 );  // define bounds on
                                    // control input

// EXPORT TAILORED C-CODE:
// -----------------------
    MPCexport mpc(ocp);       // construct module for
    mpc.exportCode();         // auto-generating code

    return 0;
}
```

Figure 8.4: A tutorial C++ code for a MPC problem using ACADO.

Automatic Code Generation

Once a specific model predictive control problem has been set up with ACADO, we can export the code via the MPCexport class. This module will generate optimized C-code which is based on hard-coded dimensions and which uses static memory only. There are four major optimized C-functions generated:

- First, the possibly non-linear right-hand side as well as its derivatives with respect to the states and controls are exported as C-code. Here, the derivatives are symbolically simplified employing automatic differentiation tools and using zero-entries in the Jacobian.
- Second, a tailored Runge-Kutta method for the model equations is generated. This Runge-Kutta routine also integrates the associated variational differential equations which are needed to compute the derivatives of the function G. For non-adaptive step-sizes this is equivalent to automatic differentiation in forward mode.
- Third, a discretization algorithm is exported which organizes the single- or multiple-shooting evaluation together with the required linear algebra routines for condensing.
- Fourth, the real-time iteration Gauss-Newton method is auto-generated. At this point, the ACADO Code Generation tool employs a tailored algorithm for solving dense QPs of the form (8.2.8): either the code generation tool CVXGEN [166] to export a tailored C-code or an adapted variant of the online QP solver qpOASES [246] using fixed dimensions and static memory.

In order to illustrate how the exported code looks like, Figure 8.5 shows a snap-shot of an automatically generated initial value embedding step in plain C. We might still be able to guess that this piece of code implements a hard coded matrix-vector multiplication for a system with four states. However, auto-generated code is not designed to be easily readable but to be efficient and reliable.

Note that the ACADO Code Generation tool generates self-contained code, i.e., no additional libraries need to be linked. It does not contain any if or switch statements, i.e., we can completely exclude that the program runs into a part of code which we have accidentally never tested. Moreover, there are no malloc/free or new/delete statements in the auto-generated code. All the memory is static and global while the dimensions are hard-coded, i.e., no segmentation faults can occur. Moreover, we avoid

```
[...]
void initialValueEmbedding( ){
params.g[0] = acadoWorkspace.g[4] +
acadoWorkspace.H[4]*acadoWorkspace.deltaY[0] +
acadoWorkspace.H[18]*acadoWorkspace.deltaY[1] +
acadoWorkspace.H[32]*acadoWorkspace.deltaY[2] +
acadoWorkspace.H[46]*acadoWorkspace.deltaY[3];
params.g[1] = acadoWorkspace.g[5] +
acadoWorkspace.H[5]*acadoWorkspace.deltaY[0] +
[...]
```

Figure 8.5: A snap-shot of automatically generated code: the produced C-code is hard to read but efficient and reliable.

`for`-loops whenever reasonable in order to ensure maximum efficiency, though this might also be done by the compiler.

Finally, the `ACADO Code Generation` tool offers an option to export code using single precision arithmetic. This is advantageous for certain hardware platforms but limits the applicability of the exported code to more well-conditioned problem formulations.

Remarks on Embedded QP Solvers

The `ACADO Code Generation` tool interfaces two QP solvers based on different algorithmic strategies:

The first one is a primal-dual interior-point solver which is auto-generated by the package `CVXGEN` [166]. The exported algorithm is implemented in highly efficient plain C code that only makes use of static memory. A major advantage of interior-point algorithms is their relatively constant calculation times for each occurring QP [46].

Active-set algorithms form a second class of suitable QP solvers, thus also the open-source package `qpOASES` [246] – which implements an online active set strategy [89] – is interfaced. For the `ACADO Code Generation` tool, a modification using hard-coded dimensions and static memory is employed. Calculation times of active-set solvers strongly depend on the number of required active set changes, which is hard to predict. On the other hand each active set iteration is much faster than an interior-point iteration. In addition, the availability of dedicated hot-starting procedures are an advantage of active set methods.

8.4 The Performance of the Auto-Generated NMPC Algorithm

In order to demonstrate the performance of auto generated NMPC algorithms we apply the ACADO auto-generation tools to two benchmark problems arising in mechatronics and chemical engineering. Both examples are tested using online optimization problem formulations with and without state constraints.

NMPC for a crane model

Let us consider a crane with mass m, line length L, excitation angle ϕ, and horizontal trolley position p. Here, our control input is the acceleration a of the trolley. With v being the trolley velocity and ω being the angular velocity of the mass point, the system can be described by a simple but non-linear differential equation system of the form (8.3.2), where b is a positive damping constant. We use the parameters $m = 1$, $L = 1$, $b = 0.2$ as well as $g = 9.81$.

Now, we export a nonlinear MPC code using the ACADO Code Generation tool. Our MPC formulation coincides with the problem (8.2.1), where $\xi := (p, v, \phi, \omega)^T$ is the state and $\zeta := a$ the control while the corresponding right-side function f is given above. The control bounds are $\underline{z} = -1$ and $\overline{z} = 1$. Additionally, zero terminal constraints are imposed at the end of the horizon.

Running the real-time loop leads to a computation time of about $95\,\mu$s per real-time iteration (or about $86\,\mu$s without zero-terminal constraints). This result has been obtained

Table 8.1: Run-time performance of the auto-generated NMPC algorithm applied to the crane model.

	CPU time	%
Integration & sensitivities	$53\ \mu$s	$56\,\%$
Condensing	$24\ \mu$s	$25\,\%$
QP solution (with qpOASES)	$13\ \mu$s	$13\,\%$
Remaining operations	$< 5\ \mu$s	$< 6\,\%$
One complete real-time iteration	$95\ \mu$s	$100\,\%$

on a 2.8 GHz processor with 4 GB RAM by running the real-time iteration loop 10^4 times and taking the average. Note that the compiled code for the whole controller has a size of 160 kB (under Linux) and requires 14 kB for storing problem data and intermediate results. Table 8.1 shows a more detailed list with the computation times. Here, we have employed a Runge-Kutta integrator of order 4 using 20 integrator steps which yields an sufficient integrator accuracy of $\approx 10^{-3}$. The time horizon of lenth $T = 3$ was divided into 10 control intervals, which determines the dimension of the dense QP.

Note that the time for the feedback step is mainly determined by the time which is needed to solve the dense QP (8.2.8). Due to the efficiency of the QP solution, the time between availability of the measurement and the application of the new control is only $13\,\mu$s. The code generated by `ACADO` allows us to perform the preparation step within the remaining $82\,\mu$s.

NMPC for a Continuous Stirred Tank Reactor

As a second example, we consider a benchmark problem of a continuous stirred tank reactor (CSTR) with four states and two controls. The corresponding model has been proposed in [68, 141]. Here, the first two states, c_A and c_B, are the concentrations of cyclopentadiene (substance A) and cyclopentenol (substance B), respectively, while the other two states, ϑ and ϑ_K, denote the temperature in the reactor and temperature in the cooling jacket of the tank reactor. The state vector is $\xi = (c_A, c_B, \vartheta, \vartheta_K)^T$. Our first input is the feed inflow which is controlled via its scaled rate $\zeta_1 = \frac{\dot{V}}{V_R}$, while the temperature ϑ_K is held down by an external heat exchanger whose heat removal rate $\zeta_2 = \dot{Q}_K$ can be controlled as well.

The following nonlinear model can be found in [68, 141]:

$$\begin{aligned}
\dot{c}_A(t) &= u_1(c_{A0} - c_A(t)) - k_1(\vartheta(t))c_A(t) - k_3(\vartheta(t))(c_A(t))^2 \\
\dot{c}_B(t) &= -u_1 c_B(t) + k_1(\vartheta(t))c_A(t) - k_2(\vartheta(t))c_B(t) \\
\dot{\vartheta}(t) &= u_1(\vartheta_0 - \vartheta(t)) + \frac{k_w A_R}{\rho C_p V_R}(\vartheta_K(t) - \vartheta(t)) \\
&\quad -\frac{1}{\rho C_p}\left[k_1(\vartheta(t))c_A(t)H_1 + k_2(\vartheta(t))c_B(t)H_2 + k_3(\vartheta(t))(c_A(t))^2 H_3 \right] \\
\dot{\vartheta}_K(t) &= \frac{1}{m_K C_{PK}}\left(u_2 + k_w A_R(\vartheta(t) - \vartheta_K(t))\right) .
\end{aligned}$$

Therein, the reaction rate functions k_i are given by

$$k_i(\vartheta(t)) = k_{i0} \cdot \exp\left(\frac{E_i}{\vartheta(t) + 273.15\,^\circ\mathrm{C}}\right) \text{ with } i \in \{1,2,3\} .$$

We set up a closed-loop scenario using the above model, where the parameters, control bounds, and end weights are taken from [68]. For illustration, we chose three different set points, one of which is constructed such that it can not be tracked exactly due to over-restrictive state constraints:

$$\vartheta(t) \geq 98\,^\circ C\,,\ \vartheta_K(t) \geq 92\,^\circ C\,.$$

The first set point, simulated for $t \in [0, 3000\mathrm{s}]$ corresponds to the choice in [68]. Moreover, the closed-loop setup is similar to the one reported in [41, 57] where the main difference is that we divide the prediction horizon of $T = 1500\,\mathrm{s}$ into 10 instead of 22 control intervals of equal length (which was found to hardly affect the control performance).

Figure 8.6 shows the result of the closed-loop simulation using auto-generated code (binary size of $220\,\mathrm{kB}$ and $25\,\mathrm{kB}$ for the data). We use a sampling time of $5\,\mathrm{s}$ and employ a Runge-Kutta integrator of order 4 using 20 integrator steps. Running it on the same hardware as used for the crane example, we obtain the run-times listed in Table 8.2. The worst measured total run-time for one real-time iteration is about $400\,\mu\mathrm{s}$, which is several orders of magnitude faster than run-times reported earlier. For example, computation times in the order of minutes for a very similar setup have been reported fifteen years ago in [57], while [41, 68] report computation times of about one second (considering the

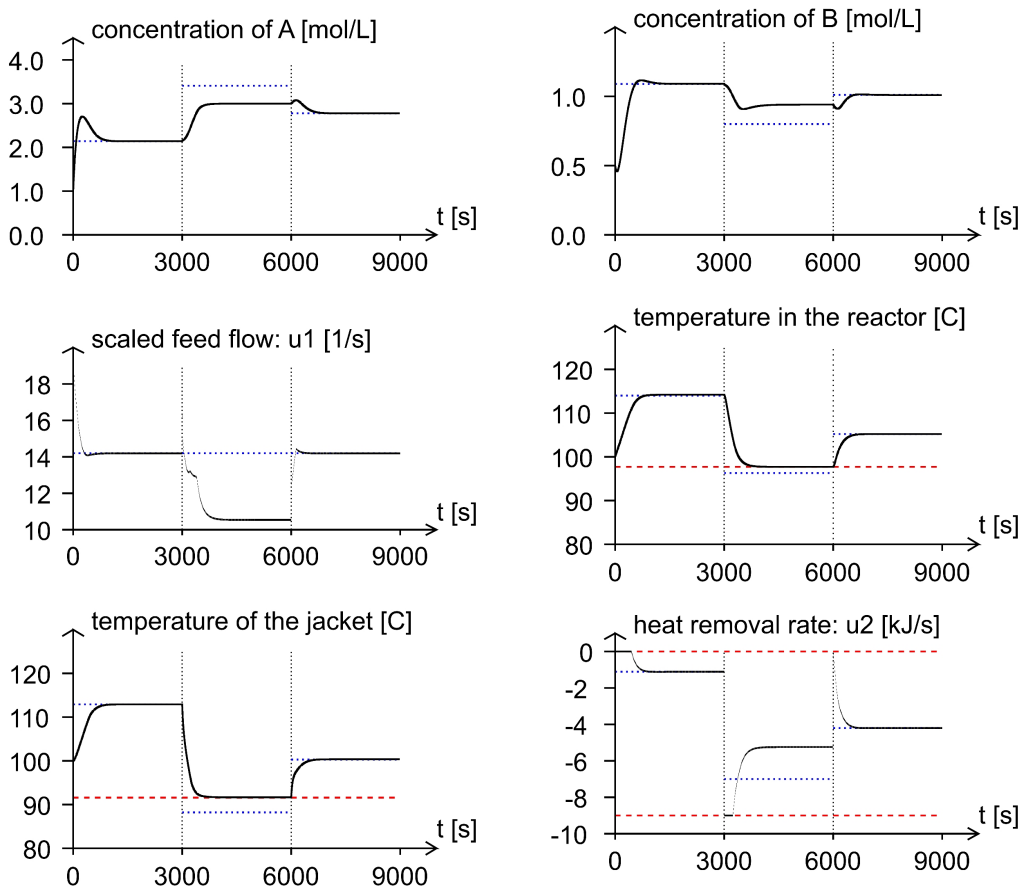


Figure 8.6: Simulation of a closed-loop scenario for the continuous stirred tank reactor showing all four states and the two control inputs. The second set-point (dotted and blue) cannot be reached due to the over-restrictive constraints (dashed and red).

Table 8.2: Worst-case run-time performance of the auto-generated NMPC algorithm applied to the CSTR model using 10 control intervals with state constraints.

	CPU time	%
Integration & sensitivities	$121\ \mu$s	$30\,\%$
Condensing	$98\ \mu$s	$24\,\%$
QP solution (with qpOASES)[3]	$180\ \mu$s	$44\,\%$
Remaining operations	$< 5\ \mu$s	$< 2\,\%$
A complete real-time iteration	$404\ \mu$s	$100\,\%$

computer hardware used that time, not more than a factor of 10 in the run-time difference can be explained by the speed-up of PCs). If state constraints are left away, the QP solution time for our scenario reduces to less than $30\,\mu$s (and condensing becomes slightly faster), thus an overall real-time iteration would take not more than $240\,\mu$s in that case[2].

Note that up to 68% (or 45% if no state constraints are present) of the computation time is spent in the QP solver and the condensing routine, whose computational load grows cubically with the number of control intervals. Thus, when longer control horizons are necessary, a tailored sparse QP solver should be used instead.

[2] Auto-generated code using 22 control intervals and no state constraints would still be very fast taking less than 1.5 ms per real-time iteration.

[3] This worst-case runtime corresponds to 21 active set changes required once during transition to the infeasible set point. Each additional change would require $6.5\ \mu$s extra.

Chapter 9

A Quadratically Convergent Inexact SQP Method for DAE Systems

This chapter is about an inexact SQP method, which is tailored for large scale optimal control problems which include differential algebraic equations (DAE). Note that the corresponding algorithm is not exclusively for robust optimization problems, but rather a general algorithm for nonlinear optimal control problems which include differential algebraic equations. This algorithm has been implemented within `ACADO Toolkit` (cf. Chapter 7) and tested by optimizing a distillation column with 82 differential and 122 algebraic states. The algorithm itself together with results of this optimization study are presented here. Note that this chapter is based on a publication [128] which is currently under review.

9.1 Introduction

For the numerical solution of nonlinear optimal control problems with an underlying differential algebraic equation (DAE) typically nonlinear programming (NLP) methods - also called direct approaches - are applied. Here, the DAE is discretized first and the optimal control problem is transformed into a finite dimensional NLP. Many researchers have developed collocation methods to discretize the dynamic model [30, 208]. This allows to discretize the DAE exclusively at the level of the NLP, but leads to extremely large and sparse optimization problems, while the collocation scheme must be adapted to control the discretization error. Alternatively, a single shooting approach, as introduced

in [143, 204], or a multiple shooting approach, as introduced in [43, 187], can be applied to discretize the problem on a moderate number of coarse intervals making use of an integration routine which adaptively discretizes the DAE in an accurate way within these coarse intervals.

In this chapter, we concentrate on a particular aspect of direct multiple shooting methods for DAE: as the differential algebraic equation needs to be simulated subsequently during the iterations of the NLP algorithm, it is not efficient to compute consistent initializations in every step of the optimization algorithm. In the simplest form of multiple shooting methods for DAE [121, 122] this aspect has simply been neglected, leading to more expensive DAE integration phases, where in every step of the optimization algorithm and in every multiple shooting node the set of nonlinear consistency conditions must be solved by Newton's method. However, in [42] and later in [150] this problem has been overcome by the introduction of a DAE relaxation allowing to satisfy the consistency conditions only in the optimal solution. This relaxation function is weakening the algebraic consistency conditions by introducing slack parameters in such a way that the DAE is always consistent. Actually, it turns out that the sensitivities with respect to these slack parameters are only needed in certain directions for the case that partially reduced sequential quadratic programming (PRSQP) methods are used on the top-level of the optimization as introduced by Leineweber [150]. However, these methods suffer from the fact that the implementation of the partially reduced SQP strategy and the sensitivity generation of the DAE are deeply intertwined, which makes the implementation complicated.

The main contribution of this chapter is divided into two parts: First, a special parameterized relaxation function for the DAE is suggested. This special relaxation function is chosen in such a way that the sensitivity directions of the state trajectory with respect to the relaxation parameters vanish in the optimal solution. And second, an inexact SQP method is proposed, which uses this property of the new relaxation function in a systematic way. This inexact SQP method has the interesting property that the approximations of the Hessian and Jacobian matrices become exact within the optimal solution. As discussed in [237, 80], general purpose inexact SQP methods can achieve locally q-superlinear convergence for the case that a suitable update method for the matrix approximations is applied. However, although the specialized SQP method proposed in this chapter is also inexact, i.e., the Hessian and the constraint Jacobian in the QPs are only approximated, the q-quadratic convergence properties can be recovered.

In Section 9.2 we review the direct multiple shooting discretization approach for DAE optimal control problems and introduce the basic notation as well as the concept of

relaxation functions. In the next step, we concentrate on a special class of relaxation functions for DAE systems, which are analyzed in Section 9.3 and for which we can show desirable properties. These properties are used in Section 9.4, where the inexact SQP method is constructed. Within this section, we discuss the q-quadratic convergence properties of the method. In Section 9.5, the new approach is applied and successfully tested with both: a small-scale toy problem and with a large-scale real-world DAE optimization problem arising in the context of optimal control of continuous distillation processes. The latter model, which was validated at a real-world distillation column [68], includes 122 implicit algebraic as well as 82 differential states.

9.2 Discretization of DAE Optimization Problems

In this section we introduce the following standard formulation of equality constrained DAE optimization problems:

$$\boxed{\begin{array}{lrcl} \underset{x(\cdot),z(\cdot),u(\cdot),p}{\text{minimize}} & & & J[x(\cdot),z(\cdot),u(\cdot),p] \\ \text{subject to:} & & & \\ \forall t \in [0,m]: & \dot{x}(t) & = & f(t,x(t),z(t),u(t),p) \\ \forall t \in [0,m]: & 0 & = & g(t,x(t),z(t),u(t),p) \\ & 0 & = & r(x(0),x(T),p) \end{array}} \tag{9.2.1}$$

Here, $x : [0,m] \to \mathbb{R}^{n_x}$ and $z : [0,m] \to \mathbb{R}^{n_z}$ denote differential and algebraic states respectively while $u : [0,m] \to \mathbb{R}^{n_u}$ is a time dependent control and $p \in \mathbb{R}^{n_p}$ a time constant parameter. For ease of notation, we assume that m is a given integer. Note that this can always be achieved by rescaling the time horizon if necessary. The right-hand side functions f, g and r are assumed to be twice continuously differentiable in all arguments and the function g should additionally satisfy that $\frac{\partial g}{\partial z}$ is regular, i.e., the DAE is assumed to have the index 1. Moreover, J denotes an objective functional. Note that for notational simplicity the above DAE optimal control problem (9.2.1) does not take any inequality constraints into account. However, the following considerations can easily be generalized for the case that we have additional inequalities using SQP methods [54, 44].

We are interested in the case that the above DAE optimization problem is discretized by Bock's direct multiple shooting approach [43]. This means that we discretize our control input u on the finite mesh $0 = t_0 < t_1 < \ldots < t_m = m$ with $t_i = i$ for all

$i \in \{0, ..., m\}$ writing the piecewise constant control as $\forall t \in [i, i+1]: \; u(t) := u_i \in \mathbb{R}^{n_u}$ with $i \in \{0, \ldots, m-1\}$. The main idea of multiple shooting is to take not only the discretized control input but also the initial values $s_0 := x(0), \ldots, s_m := x(m)$ at the nodes into the formulation of the discrete NLP - in contrast to single shooting, where only s_0 is regarded as a free variable. More precisely, we define the functions $\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_{m-1} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ to be the solutions $\hat{X}_i(s_i, u_i, p) := \hat{x}_i(i+1)$ of the given DAE on the multiple shooting intervals ($i \in \{0, \ldots, m-1\}$):

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_i(t) &= f(t, \hat{x}_i(t), \hat{z}_i(t), u_i, p) \\ 0 &= g(t, \hat{x}_i(t), \hat{z}_i(t), u_i, p) \\ \text{with} \quad \hat{x}_i(0) &= s_i \; , \end{aligned} \tag{9.2.2}$$

for all $t \in [i, i+1]$. Note that the functions $\hat{X}_i$ are well defined on their domains, in the sense that the solution of the DAE (9.2.2) uniquely exists as we assume that f and g are twice continuously differentiable and $\frac{\partial g}{\partial z}$ regular.[1]

As the optimal solution for the state x should be continuous, we have to require matching conditions to be satisfied at the multiple shooting nodes. The initial value s_{i+1} associated with the i-th interval should in the optimal solution be equal to the end value $\hat{X}_{i-1}(s_{i-1}, u_{i-1}, p)$ of the previous interval. Thus, the matching conditions can be summarized in the form

$$\hat{H}(\alpha) \; := \; \begin{pmatrix} s_1 - \hat{X}_0(s_0, u_0, p) \\ s_2 - \hat{X}_1(s_1, u_1, p) \\ \vdots \\ s_m - \hat{X}_{m-1}(s_{m-1}, u_{m-1}, p) \end{pmatrix} \; = \; 0 \; . \tag{9.2.3}$$

Here, we collect the $(m+1)$ initial values at the nodes, the m control input pieces as well as the free parameters in the variable

$$\alpha := \left(s_0^T, s_1^T, \ldots, s_m^T, u_0^T, u_1^T, \ldots u_{m-1}^T, p^T\right)^T \; \in \mathbb{R}^{n_\alpha} \; ,$$

where the dimension n_α is given by $n_\alpha := (m+1)n_x + mn_u + n_p$.

[1] For a proof of this uniqueness and existence statement we refer to [198].

In the next step, the objective functional and the boundary constraints must be discretized. For this aim, we introduce the algebraic node values

$$\beta \;:=\; \left(\beta_0^T, \beta_1^T, \ldots, \beta_m^T\right)^T \;:=\; \left(z(0)^T, z(1)^T, \ldots, z(m)^T\right)^T \;.$$

These algebraic node values allow us to finally write the discretized version of the optimal control problem (9.2.1) in the form

$$\boxed{\begin{array}{lll} \underset{\alpha,\beta}{\text{minimize}} & F(\alpha,\beta) & \\ \text{subject to:} & G(\alpha,\beta) & = 0 \\ & \hat{H}(\alpha) & = 0 \end{array}}\;. \tag{9.2.4}$$

Here, the algebraic consistency conditions and the discretized boundary constraint have been summarized in the function $G : \mathbb{R}^{n_\alpha} \times \mathbb{R}^{n_\beta} \to \mathbb{R}^{(m+1)n_z+n_r}$ which is defined as

$$G(\alpha,\beta) \;:=\; \begin{pmatrix} g(0, s_0, \beta_0, u_0, p) \\ g(1, s_1, \beta_1, u_1, p) \\ \vdots \\ g(m-1, s_{m-1}, \beta_{m-1}, u_{m-1}, p) \\ g(m, s_m, \beta_m, u_{m-1}, p) \\ r(s_0, s_m, p) \end{pmatrix}. \tag{9.2.5}$$

Finally, the function $F : \mathbb{R}^{n_\alpha} \times \mathbb{R}^{n_\beta} \to \mathbb{R}$ represents the discrete version of the objective functional J evaluated at the multiple shooting points. In the following we assume that F is twice continuously differentiable in all its arguments.

Once the discrete optimization problem (9.2.4) is derived, we can of course use a standard NLP solver - e.g. an SQP method - to solve this structured nonlinear program. As it was mentioned in the introduction, this strategy has been applied in some approaches of multiple-shooting for DAE [121, 122]. However, note that the evaluation of the function $\hat{H}$ - or, more precisely, the evaluation of the functions $\hat{X}_0, \ldots, \hat{X}_{m-1}$ - is typically the most expensive part of the algorithm, as it requires to solve m DAEs of the form (9.2.2) in each step of the SQP algorithm. Numerical integration routines which are able to numerically solve these DAE systems (9.2.2), usually proceed in two phases: in the first phase, a consistent algebraic initialization point $\beta_i^* \in \mathbb{R}^{n_z}$ is generated which satisfies

$\|g(i, s_i, \beta_i^*, u_i, p)\| \leq \epsilon$, where $\epsilon \geq 0$ is a small constant depending on how accurate we want to solve the DAE. This first phase can for example be performed by a Newton method, as the Jacobian $\frac{\partial g}{\partial z}$ is assumed to be regular. In the second phase, the integration algorithm is started, which can for example be based on an implicit Runge-Kutta method [114] or on backward differentiation formula (BDF) methods [15, 61].

However, if we regard optimization methods for DAE, it is usually not efficient to compute a consistent initialization in every step of the optimization algorithm. The key idea to avoid this first phase is to first modify the DAE such that it is by definition consistent and then to simulate the corresponding relaxed differential algebraic equation during the optimization. This strategy has originally been developed in [42] and was refined in [150]. Here, relaxation means that the original algebraic condition in (9.2.2) is (for all $i \in \{1, \ldots, m-1\}$) replaced by a modified equation of the form

$$0 \quad = \quad g(t, x_i(t), z_i(t), u_i, p) - \vartheta\left(\gamma_i, t - i\right) \tag{9.2.6}$$

for all $t \in [i, i+1]$, where $\vartheta : \mathbb{R}^{n_z} \times [0,1] \to \mathbb{R}^{n_z}$ is a relaxation function and

$$\gamma_i := g(i, s_i, \beta_i, u_i, p) \ .$$

This relaxation function is required to satisfy the conditions

$$\forall \gamma \in \mathbb{R}^{n_z} : \qquad \vartheta(\gamma, 0) = \gamma \tag{9.2.7}$$

$$\forall \tau \in [0,1] : \qquad \vartheta(0, \tau) = 0 \tag{9.2.8}$$

such that equation (9.2.6) is by construction satisfied at $t = i$. In addition, if the algebraic consistency condition $\gamma_i = g(i, s_i, \beta_i, u_i, p) = 0$ holds at the i-th shooting node, the condition (9.2.8) guarantees that the function ϑ vanishes on the corresponding shooting interval such that the relaxed condition (9.2.6) coincides with the original algebraic condition.

In [42] the function

$$\vartheta_1(\gamma, \tau) := \gamma \quad \text{for all } (\gamma, \tau) \in \mathbb{R}^{n_z} \times [0,1] \tag{9.2.9}$$

was chosen as a relaxation function, while in [15] and in [150] it is reported that the function

$$\vartheta_2(\gamma, \tau) := \gamma \, \exp\left(-\delta\tau\right) \quad \text{for all } (\gamma, \tau) \in \mathbb{R}^{n_z} \times [0,1] \tag{9.2.10}$$

with the empirical value $\delta = 5$ works better in practice. One of the two main contributions of this chapter is to suggest a third choice for the function ϑ, which will be defined in equation (9.3.1). In Sections 9.3 and 9.4 we will discuss the advantages and desirable properties of this new relaxation function in comparison to the existing choices (9.2.9) and (9.2.10). However, let us first address the question what the introduction of a relaxation function changes with regard to the discretization of the continuous optimal control problem: for this aim, we consider the relaxed differential algebraic equations on the multiple shooting intervals ($i \in \{0, \ldots, m-1\}$):

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} x_i(t) &= f(t, x_i(t), z_i(t), u_i, p) \\ 0 &= g(t, x_i(t), z_i(t), u_i, p) - \vartheta\left(\gamma_i, t - i\right) \\ \text{with} \quad x_i(0) &= s_i \ , \end{aligned} \tag{9.2.11}$$

for all $t \in [i, i+1]$. Analogous to the functions $\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_{m-1}$ we define the functions $X_0, X_1, \ldots, X_{m-1} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ to be the solutions

$$X_i(s_i, \gamma_i, u_i, p) := x_i(i+1)$$

of the relaxed DAE system (9.2.11). Moreover, we define a function H by

$$H\left(\alpha, \Gamma\right) := \begin{pmatrix} s_1 - X_0(s_0, \gamma_0, u_0, p) \\ s_2 - X_1(s_1, \gamma_1, u_1, p) \\ \vdots \\ s_m - X_{m-1}(s_{m-1}, \gamma_{m-1}, u_{m-1}, p) \end{pmatrix} . \tag{9.2.12}$$

Here, we us the definition

$$\Gamma \ := \ \left(\gamma_0^T, \gamma_1^T, \ldots, \gamma_{m-1}^T, \gamma_m^T, \gamma_r^T\right)^T \ := \ G(\alpha, \beta) \tag{9.2.13}$$

as a notation for the components of the function G. Now, the matching conditions for the relaxed DAE can be summarized as

$$H(\alpha, G(\alpha, \beta)) \ = \ 0 \ . \tag{9.2.14}$$

The associated discretized optimization problem takes the form

$$\boxed{\begin{aligned} &\underset{\alpha,\beta}{\text{minimize}} && F(\alpha, \beta) \\ &\text{subject to:} && 0 \ = \ G(\alpha, \beta) \\ & && 0 \ = \ H(\alpha, G(\alpha, \beta)) \end{aligned}} \ . \tag{9.2.15}$$

Note that the optimization problems (9.2.4) and (9.2.15) are equivalent by construction:

Lemma 9.1: *If the relaxation function ϑ satisfies $\vartheta(0,\tau) = 0$ for all $\tau \in [0,1]$, the optimization problems* (9.2.4) *and* (9.2.15) *are equivalent. I.e., a point (α^*, β^*) is an optimal point of problem* (9.2.4) *if and only if it is an optimal point of problem* (9.2.15).

Proof: If (α^*, β^*) is a feasible point of one of the optimization problems, it must satisfy $G(\alpha^*, \beta^*) = 0$, i.e., the consistency condition $\gamma_i^* = g(t_i, s_i^*, \beta_i^*, u_i^*, p^*) = 0$ holds for all $i \in \{1, \ldots, m-1\}$. Thus, the relaxation function ϑ satisfies $\vartheta(\gamma_i^*, \tau) = \vartheta(0, \tau) = 0$ for all $\tau \in [0,1]$, i.e., the relaxation vanishes at an optimal point such that the relaxed and the original DAE exactly coincide. Consequently, we have $\hat{H}(\cdot) \equiv H(\cdot, G(\alpha^*, \beta^*)) \equiv H(\cdot, 0)$ and the optimization problems (9.2.4) and (9.2.15) must be equivalent. □

The advantage of the function H in comparison to the function $\hat{H}$ is that a single evaluation is cheaper, as the simulation of the relaxed DAE (9.2.11) does not require a first phase, in which a consistent initial value is computed. However, the function H does not only depend on α but implicitly also on the variable β which enters via the variable $\Gamma = G(\alpha, \beta)$. Thus, assuming that H is differentiable, the computation of the Jacobian of H with respect to Γ can be expensive - especially, if we have many algebraic states. In order to overcome this problem, it has in [150] been proposed to use partially reduced sequential quadratic programming (PRSQP) methods to solve the relaxed problem (9.2.15). These methods have the advantage that only $n_\alpha + 1$ directional derivatives of H per SQP step are needed - in comparison to $n_\alpha + n_\beta$ directional derivatives of H, which would be needed for a computation of the whole Jacobian needed within the full-space SQP method. However, PRSQP methods suffer from the fact that the simulation and derivative generation of the DAE system and the optimization routine itself are deeply intertwined [150], which makes the implementation complex and less attractive from a programming point of view.

The first contribution of this chapter is the introduction of a novel relaxation function ϑ defined by equation (9.3.1). The desirable properties (cf. Section 9.3) of this special relaxation function will allow us to avoid the computation of derivatives of H with respect to β, i.e., we need only n_α derivatives of H per SQP step, while the optimization and DAE simulation together with the derivative generation can be implemented in a modular way. In the following sections we will explain this idea step by step.

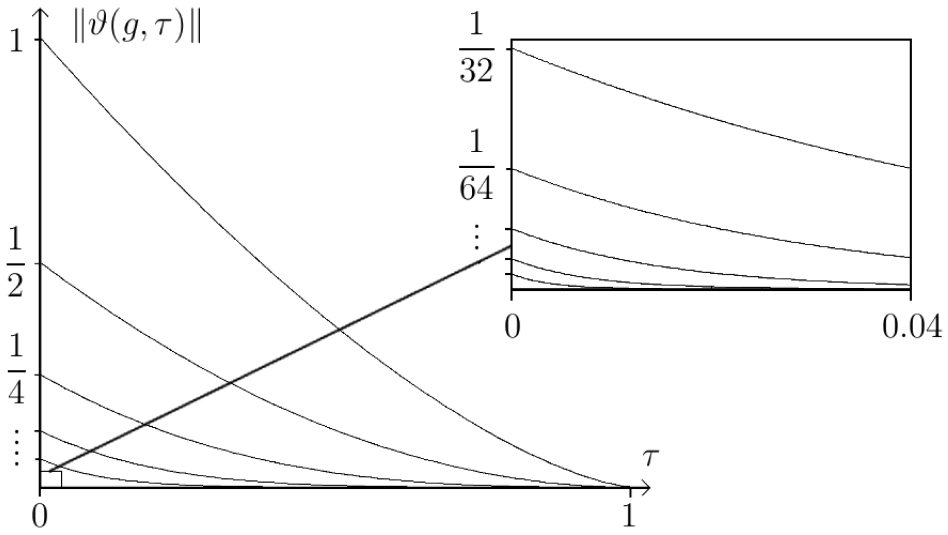


Figure 9.1: The function $\|\vartheta(\gamma,\tau)\|$ in dependence on τ for several values of $\|\gamma\|$ (with the values $\|\gamma\| \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{256}\}$) for the case $a = \frac{1}{2}$ and $b = 1$.

9.3 Properties of the New Relaxation Function

In this section we are interested in the analysis of the relaxed DAE (9.2.11) under the assumption that the relaxation function $\vartheta : \mathbb{R}^{n_z} \times [0,1] \to \mathbb{R}^{n_z}$ has the form

$$\vartheta(\gamma,\tau) := \begin{cases} \gamma\,(1-\tau)^{\frac{a+b\|\gamma\|}{\|\gamma\|}} & \text{if } \|\gamma\| \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{9.3.1}$$

for all $(\gamma,\tau) \in \mathbb{R}^{n_z} \times [0,1]$. Here, $a > 0$ and $b \geq 1$ are positive design parameters while $\|\cdot\|$ denotes the Euclidean norm.[2] In Figure 9.1 the norm $\|\vartheta(\gamma,\tau)\|$ is plotted as a function of the variable $\tau \in [0,1]$ for several values of $\|\gamma\|$ using the choice $a = \frac{1}{2}$ and $b = 1$.

Lemma 9.2: *The relaxation function ϑ, which is defined by equation* (9.3.1)*, has the following properties:*

(*i*) *The function ϑ is continuous on its domain $\mathbb{R}^{n_z} \times [0,1]$ and satisfies the fundamental relations* (9.2.7) *and* (9.2.8)*.*

[2] In the following, we will use the notation $\|\cdot\|$ not only for the Euclidean norm on $\mathbb{R}^{n_z}$, but also for induced matrix or tensor norms.

(ii) *The function ϑ is differentiable with respect to τ and the associated derivative $\dot{\vartheta} := \frac{\mathrm{d}\vartheta}{\mathrm{d}\tau}$ satisfies*

$$\forall \tau \in [0,1]: \quad \left\| \dot{\vartheta}(\gamma,\tau) \right\| \;\leq\; a + \mathbf{O}\left(\|\gamma\|\right) \;. \tag{9.3.2}$$

(iii) *On the domain $(\mathbb{R}^{n_z} \setminus \{0\}) \times [0,1]$ the function ϑ is twice differentiable with respect to γ and the associated derivatives $\vartheta' := \frac{\mathrm{d}\vartheta}{\mathrm{d}\gamma}$ and $\vartheta'' := \frac{\mathrm{d}^2\vartheta}{\mathrm{d}\gamma^2}$ satisfy*

$$\begin{aligned} \int_0^1 \left\| \vartheta'(\gamma,\tau) \right\| \,\mathrm{d}\tau \;&\leq\; \mathbf{O}\left(\|\gamma\|\right) \;, \\ \int_0^1 \left\| \vartheta'(\gamma,\tau) \right\|^2 \,\mathrm{d}\tau \;&\leq\; \mathbf{O}\left(\|\gamma\|\right) \;, \\ \textit{and} \quad \int_0^1 \left\| \vartheta''(\gamma,\tau) \right\| \,\mathrm{d}\tau \;&\leq\; \mathbf{O}\left(1\right) \end{aligned} \tag{9.3.3}$$

for all $\gamma \in \mathbb{R}^{n_z} \setminus \{0\}$.

Proof: The statement (i) follows immediately from the definition of ϑ. Let us define the exponent κ by

$$\kappa := \frac{a + b\|\gamma\|}{\|\gamma\|} \;\geq\; b \geq 1 \;.$$

We can compute the slope of the function ϑ by direct computation:

$$\left\| \dot{\vartheta}(\gamma,\tau) \right\| \;=\; \left\{ \begin{array}{ll} \left(a + b\|\gamma\|\right)\left(1-\tau\right)^{\kappa-1} & \text{if } \|\gamma\| \neq 0 \\ 0 & \text{otherwise} \end{array} \right\} \;. \tag{9.3.4}$$

Obviously, we have

$$\left\| \dot{\vartheta}(\gamma,\tau) \right\| \;\leq\; a + b\|\gamma\| \;=\; a + \mathbf{O}\left(\|\gamma\|\right) \tag{9.3.5}$$

for all $\tau \in [0,1]$, which shows the statement (ii). Moreover, the differentiability of ϑ on the domain $(\mathbb{R}^{n_z} \setminus \{0\}) \times [0,1]$ is a rather trivial consequence of its definition (9.3.1). Explicitly, ϑ' can be written as

$$\vartheta'(\gamma,\tau) \;=\; \left[1 - \frac{a\gamma\gamma^T \log\left(1-\tau\right)}{\|\gamma\|^3} \right] \left(1-\tau\right)^{\kappa} \;, \tag{9.3.6}$$

where we understand the above right-hand side expression at the point $\tau = 1$ in the limit sense, which is justified since the inequality $\kappa \geq 1$ guarantees that the limit

$$\lim_{t \to 1} \log(1-\tau)(1-\tau)^{\kappa} = 0 , \tag{9.3.7}$$

exists. It is a simple exercise to show that for all integers $n \in \{0, 1, 2\}$ the following relation holds (for all $\kappa \geq 1$):

$$\int_0^1 \log(1-\tau)^n (1-\tau)^{\kappa} \, \mathrm{d}\tau = (-1)^n \frac{1}{(1+\kappa)^{n+1}} . \tag{9.3.8}$$

Now, we use formula (9.3.8) once with $n = 0$ and once with $n = 1$ to find

$$\begin{aligned} \int_0^1 \|\vartheta'(\gamma,\tau)\| \, \mathrm{d}\tau &= \int_0^1 \left[1 - \frac{a}{\|\gamma\|} \log(1-\tau)\right] (1-\tau)^{\kappa} \, \mathrm{d}\tau \\ &= \frac{1}{\kappa+1} + \frac{a}{\|\gamma\|} \frac{1}{(\kappa+1)^2} \leq \frac{2}{a} \|\gamma\| . \end{aligned} \tag{9.3.9}$$

Similarly, we compute the integral

$$\begin{aligned} \int_0^1 \|\vartheta'(\gamma,\tau)\|^2 \, \mathrm{d}\tau &= \int_0^1 \left[1 - \frac{a}{\|\gamma\|} \log(1-\tau)\right]^2 (1-\tau)^{2\kappa} \, \mathrm{d}\tau \\ &= \frac{1}{2\kappa+1} + 2\frac{a}{\|\gamma\|} \frac{1}{(2\kappa+1)^2} + \frac{a^2}{\|\gamma\|^2} \frac{1}{(2\kappa+1)^3} \leq \frac{9}{8a} \|\gamma\| , \end{aligned}$$

where we have used formula (9.3.8) once for $n = 0$, once for $n = 1$, and once for $n = 2$ as well as $\kappa \geq \frac{a}{\|\gamma\|}$. Finally, we compute

$$\begin{aligned} \int_0^1 \|\vartheta''(\gamma,\tau)\| \, \mathrm{d}\tau &\leq \int_0^1 \frac{a^2}{\|\gamma\|^3} \log(1-\tau)^2 (1-\tau)^{\kappa} \, \mathrm{d}\tau \\ &\quad - \int_0^1 \frac{3a}{\|\gamma\|^2} \log(1-\tau)(1-\tau)^{\kappa} \, \mathrm{d}\tau \\ &= \frac{a^2}{\|\gamma\|^3} \frac{1}{(\kappa+1)^3} + \frac{3a}{\|\gamma\|^2} \frac{1}{(\kappa+1)^2} \leq \frac{4}{a} , \end{aligned}$$

which leads to the statement (iii). □

Remark 9.1: *The statement* (ii) *of the above Lemma is important, as the slope of* ϑ *will influence the performance of the numerical integration algorithm applied to the associated relaxed DAE. Whenever the residual* $\|\gamma\|$ *is sufficiently small, which means in our context that the algebraic consistency condition is approximately satisfied, estimate* (9.3.2) *guarantees that this slope can directly be influenced by the design parameter* a*.*

Recall that the solutions of the relaxed differential algebraic equations (9.2.11) are denoted by $X_i(s_i, \gamma_i, u_i, p)$. In the following, we analyze these functions X_i in more detail under the assumption that the relaxation function ϑ satisfies the properties (i) and (iii) of the above Lemma.

Theorem 9.1: *Let* $i \in \{0, ..., m-1\}$*,* s_i*,* u_i*, and* p *be given, the functions* f *and* g *twice continuously differentiable, and* $\frac{\partial g}{\partial z}$ *regular. If the function* ϑ *satisfies the properties* (i) *and* (iii) *of Lemma 9.2, then the function* X_i*, which is defined to be the solution of the DAE system* (9.2.11)*, is differentiable with respect to* γ_i *and its Jacobian* $X_i' := \frac{\partial X_i}{\partial \gamma_i}$ *is locally Lipschitz continuous. Moreover, we have* $X_i'(s_i, 0, u_i, p) = 0$*.*

Proof: Let us pick a $\overline{\gamma} \in \mathbb{R}^{n_z}$. For theoretical purposes, we eliminate the algebraic states of the system (9.2.11) in a neighborhood of $\overline{\gamma}$. More precisely, the implicit function theorem guarantees the existence of a bounded open neighborhood $D \subset \mathbb{R}^{n_z}$ with $\overline{\gamma} \in D$ and a twice continuously differentiable function $h : [i, i+1] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ such that the differential state x_i satisfies the ODE

$$\forall t \in [i, i+1]: \quad \dot{x}_i(t) \;=\; h(t, x_i(t), \vartheta(\gamma_i, t-i), u_i(t), p) \;\text{ with }\; x_i(i) = s_i \quad (9.3.10)$$

for all $\gamma_i \in D$. Here we have used the assumption that f and g are twice continuously differentiable in all arguments, while $\frac{\partial g}{\partial z} \neq 0$ is regular. Note that the differential equation (9.3.10) can also be written in its integral form

$$x_i(t) \;=\; s_i + \int_i^{i+1} h\left(\tau, x_i(\tau), \vartheta(\gamma_i, \tau - i), u_i(\tau), p\right) \mathrm{d}\tau \qquad (9.3.11)$$

for all $\gamma_i \in D$. Let us define the set $D^* := \{\gamma \in D \mid \gamma \neq 0\}$. The property (iii) of Lemma 9.2 guarantees that the function ϑ is at least on the domain D^* differentiable with respect to γ_i. Consequently, the derivative function $x_i' := \frac{\partial x_i}{\partial \gamma_i}$ exists on D^* and can

also be written in its integral form

$$\begin{aligned} x_i'(t) &= \int_i^t \frac{\partial h}{\partial x}\,(\tau, x_i(\tau), \vartheta(\gamma_i, \tau - i), u_i(\tau), p)\; x_i'(\tau)\,\mathrm{d}\tau \\ &\quad + \int_i^t \frac{\partial h}{\partial \vartheta}\,(\tau, x(\tau), \vartheta(\gamma_i, \tau - i), u_i(\tau), p)\; \vartheta'(\gamma_i, \tau)\,\mathrm{d}\tau \end{aligned}$$

for all $\gamma_i \in D^*$ and all $t \in [i, i+1]$. As the function h is twice continuously differentiable and as we have assumed that D is bounded, we can always find local Lipschitz constants $H_x, H_\vartheta > 0$ such that

$$\|x_i'(t)\| \;\leq\; \int_i^t H_x \|x_i'(\tau)\|\,\mathrm{d}\tau + H_\vartheta \int_i^t \|\vartheta'(\gamma_i, \tau)\|\,\mathrm{d}\tau\;. \tag{9.3.12}$$

for all $\gamma_i \in D^*$. Now, an application of the integral form of Gronwall's Lemma yields

$$\|x_i'(i+1)\| \;\leq\; H_\vartheta\, e^{H_x} \int_0^1 \|\vartheta'(\gamma_i, \tau)\|\,\mathrm{d}\tau\;, \tag{9.3.13}$$

i.e., using the property (iii) of Lemma 9.2 leads to the estimate

$$\|X_i'(s_i, \gamma_i, u_i, p)\| \;=\; \|x_i'(i+1)\| \;\leq\; \mathbf{O}\left(\|\gamma_i\|\right) \tag{9.3.14}$$

for all $\gamma_i \in D^*$.

For the case that $0 \notin D$ we have $D^* = D$, i.e., we know already that X is differentiable on the set D. Otherwise, i.e., for the case $0 \in D$, we use equation (9.3.11) in combination with Gronwall's Lemma to conclude that there exists a constant $C < \infty$ with

$$\begin{aligned} \|X_i(s_i, \gamma, u_i, p) - X_i(s_i, 0, u_i, p)\| &\leq C \max_{\tau \in [0,1]} \|\vartheta(\gamma, \tau) - \vartheta(0, \tau)\| \\ &= C \max_{\tau \in [0,1]} \|\vartheta(\gamma, \tau)\| \leq C\|\gamma\| \end{aligned} \tag{9.3.15}$$

for all $\gamma \in D$. Here, we have used the property (i) of Lemma 9.2, which guarantees that we have $\vartheta(0, \tau) = 0$ for all $\tau \in [0, 1]$.

Let $(\gamma^n) \in D^*$ be a sequence with

$$\lim_{n \to \infty} \gamma^n \;=\; 0\;.$$

Now, it is guaranteed by the estimates (9.3.14) and (9.3.15) that the function value sequences $X_i(s_i, \gamma^n, u_i, p)$ and $X_i'(s_i, \gamma^n, u_i, p)$ converge both uniformly to

$X_i(s_i, 0, u_i, p) \in \mathbb{R}^{n_x}$ and $0 \in \mathbb{R}^{n_x \times n_z}$ respectively, if n tends to infinity. Thus, we can conclude that the function X_i' exists on the whole set D and that we have $X_i'(s_i, 0, u_i, p) = 0$. Moreover, as the point $\overline{\gamma}$ around which the open set D has been constructed was arbitrary, we can transfer our statement to the whole domain $\mathbb{R}^{n_z}$, i.e., X_i' exists everywhere and we have $X_i'(s_i, 0, u_i, p) = 0$.

It remains to be shown that the function X_i' is locally Lipschitz continuous. For this aim, we start again with the integral representation (9.3.11) for $\gamma \in D^*$. Differentiating twice with respect to γ, taking the norm, and applying Gronwall's Lemma shows that there exist constants $C_0, C_1, C_2, C_3 < \infty$ with

$$\begin{aligned} \|X_i''(s_i, \gamma_i, u_i, p)\| \quad \leq \quad & C_0 + C_1 \int_0^1 \|\vartheta'(\gamma_i, \tau)\| \, \mathrm{d}\tau + C_2 \int_0^1 \|\vartheta'(\gamma_i, \tau)\|^2 \, \mathrm{d}\tau \\ & + C_3 \int_0^1 \|\vartheta''(\gamma_i, \tau)\| \, \mathrm{d}\tau \end{aligned} \tag{9.3.16}$$

for all $\gamma \in D^*$. Here, $X'' := \frac{\partial^2 X}{\partial y_0^2}$ denotes the second derivative of X with respect to y_0. This second derivative exists on D^* as the second derivative $\vartheta'' := \frac{\partial^2 \vartheta}{\partial y_0^2}$ does exist on this domain. Using the property (iii) from Lemma 9.2, we conclude that there must be a constant $M < \infty$ such that

$$\forall \gamma_i \in D^* : \quad \|X_i''(s_i, \gamma_i, u_i, p)\| \quad \leq \quad M \, . \tag{9.3.17}$$

In other words, the derivative X_i'' exists on D^* and is uniformly bounded on this set. This means that X_i' is a continuous function on D, whose derivative exists almost everywhere and is uniformly bounded. Hence, X_i' is locally Lipschitz continuous on D. With the same argumentation as above, we can continue this statement to the whole $\mathbb{R}^{n_z}$, i.e., X_i' exists everywhere and is locally Lipschitz continuous. □

Remark 9.2: *The properties (i) and (iii) from Lemma 9.2 do not uniquely characterize the relaxation function ϑ. The above theorem holds for all relaxation functions ϑ which satisfy these two properties. However, in this chapter we concentrate on the choice defined in equation* (9.3.1) *which turned out to work well in practice as it will also later be discussed in Section 9.5.*

Remark 9.3: *For the case that the relaxation function ϑ is defined by equation* (9.3.1)*, it is worthwhile to discuss that the estimate* (9.3.14) *has the explicit form*

$$\|X_i'(s_i, \gamma_i, u_i, p)\| \;\leq\; \frac{2H_\vartheta\, e^{H_x}}{a}\, \|\gamma_i\| \;. \tag{9.3.18}$$

This can be seen by using the estimate (9.3.9)*. The estimate* (9.3.18) *gives us an idea on how the function X_i' behaves with respect to the design parameter a. Indeed, choosing a very large a will lead to a small norm of X_i'. However, we should also recall that the slope of the function ϑ at the point $t = 0$ has in Proposition 9.2 been estimated by*

$$\left\|\dot{\vartheta}(\gamma, \tau)\right\| \;\leq\; a + \mathbf{O}\left(\|\gamma\|\right) \;. \tag{9.3.19}$$

Thus, if we choose a very large a the slope of ϑ can not be guaranteed to be small anymore, which might lead to small steps taken by a numerical integration routine that is used to numerically solve the relaxed DAE system (9.2.11) *based on an adaptive step size control.*

In the next step, we discuss a generalization of Theorem 9.1 to derivatives of the function X_i with respect to the variables $\alpha_i := \left(s_i^T\,,\, u_i^T, p^T\right)^T$, which enter the differential algebraic equation as the initial value, the control input, and the parameter, respectively.

Corollary 9.1: *Requiring the same assumptions as in Theorem 9.1 the function $X_{i,\alpha_i} := \frac{\partial X_i}{\partial \alpha_i}$ is differentiable with respect to γ_i and its Jacobian $X'_{i,\alpha_i} := \frac{\partial^2 X_i}{\partial \alpha_i \partial \gamma_i}$ is a locally Lipschitz continuous function. Moreover, we have $X'_{i,\alpha_i}(s_i, 0, u_i, p) = 0$.*

Proof: The proof of this corollary is almost analogous to the proof of Theorem 9.1. We start again with the integral form (9.3.11) for $\gamma_i \in D^*$ and differentiate with respect to γ_i and α_i:

$$\begin{aligned}\frac{\partial}{\partial \alpha_i} x_i'(t) \;&=\; \int_i^t \frac{\partial h}{\partial x}\, \frac{\partial}{\partial \alpha_i} x_i'(\tau)\, \mathrm{d}\tau \;+\; \int_i^t \left[\frac{\partial^2 h}{\partial x \partial \alpha_i} + \frac{\partial^2 h}{\partial x^2}\frac{\partial x_i}{\partial \alpha_i}\right] x_i'(\tau)\, \mathrm{d}\tau \\ &\quad + \int_i^t \left[\frac{\partial^2 h}{\partial \vartheta \partial \alpha_i} + \frac{\partial^2 h}{\partial \vartheta \partial x}\frac{\partial x_i}{\partial \alpha_i}\right] \vartheta'(\gamma_i, \tau)\, \mathrm{d}\tau \end{aligned} \tag{9.3.20}$$

for all $\gamma_i \in D^*$ and all $t \in [i, i+1]$. Thus, as we have $\|x_i'(\tau)\| \leq H_\vartheta e^{H_x} \int_i^\tau \|\vartheta'(\gamma_i, \tau')\|\, \mathrm{d}\tau'$ for all $\tau \in [i, i+1]$, we can find local Lipschitz constants H_x and $H_{\vartheta,\alpha}$ such that

$$\left\|\frac{\partial}{\partial \alpha_i} x_i'(t)\right\| \;\leq\; \int_i^t H_x \left\|\frac{\partial}{\partial \alpha_i} x_i'(t)\right\| \mathrm{d}\tau + \; H_{\vartheta,\alpha} \int_i^t \|\vartheta'(\gamma_i, \tau)\|\, \mathrm{d}\tau \;.$$

Now the integral form of Gronwall's Lemma yields

$$\|X'_{i,\alpha_i}(s_i,\gamma_i,u_i,p)\| \;=\; \left\|\frac{\partial}{\partial\alpha_i}x'_i(i+1)\right\| \;\leq\; H_{\vartheta,\alpha}e^{H_x}\int_0^1 \|\vartheta'(\gamma_i,\tau)\|\,\mathrm{d}\tau \;\leq\; \mathbf{O}\left(\|\gamma_i\|\right),$$

for $\gamma_i \in D^*$. For the case $0 \in D$ we use the Lipschitz relation

$$\|X_{i,\alpha_i}(s_i,\gamma_i,u_i,p) - X_{i,\alpha_i}(s_i,0,u_i,p)\| \;\leq\; \mathbf{O}\left(\|\gamma_i\|\right),$$

such that we can apply the same argumentation as in Theorem 9.1: for every sequence $(\gamma^n) \in D^*$ with $\lim\limits_{n\to\infty} \gamma^n = 0$ the function value sequences $X_{i,\alpha_i}(s_i,\gamma^n,u_i,p)$ and $X'_{i,\alpha_i}(s_i,\gamma^n,u_i,p)$ converge for $n\to\infty$ uniformly to $X_{i,\alpha_i}(s_i,0,u_i,p)$ and 0 respectively. Thus, the function X_{i,α_i} is differentiable on D and we have $X'_{i,\alpha_i}(s_i,0,u_i,p) = 0$. This statement can be continued to the whole domain.

It remains to show the local Lipschitz continuity of the function X'_{i,α_i}. Again, the argumentation is analogous to Theorem 9.1: first, we differentiate the integral form (9.3.20) once more with respect to γ_i, take the norm on both sides, apply the integral form of Gronwall's Lemma and find the estimate

$$\begin{aligned}\|X''_{i,\alpha_i}(s_i,\gamma_i,u_i,p)\| \;\leq\;& C_{\alpha,0} + C_{\alpha,1}\int_0^1 \|\vartheta'(\gamma_i,\tau)\|\,\mathrm{d}\tau + C_{\alpha,2}\int_0^1 \|\vartheta'(\gamma_i,\tau)\|^2\,\mathrm{d}\tau \\ &+ C_{\alpha,3}\int_0^1 \|\vartheta''(\gamma_i,\tau)\|\,\mathrm{d}\tau < \infty \qquad (9.3.21)\end{aligned}$$

for some constants $C_{\alpha,0}, C_{\alpha,1}, C_{\alpha,2}, C_{\alpha,3} < \infty$ and for all $\gamma_i \in D^*$. Obviously, we can now apply a completely analogous argumentation as in Theorem 9.1 to show that the function X'_{i,α_i} is locally Lipschitz continuous. □

9.4 Inexact SQP Methods for DAE Systems

In this section we come back to the question how to numerically solve the discretized optimization problem (9.2.15) of the form

$$\boxed{\begin{array}{ll}\underset{\alpha,\beta}{\text{minimize}} & F(\alpha,\beta) \\ \text{subject to:} & 0 \;=\; G(\alpha,\beta) \\ & 0 \;=\; H(\alpha,G(\alpha,\beta))\end{array}}\;. \qquad (9.4.1)$$

Recall that F and G are twice continuously differentiable functions, where G is defined by equation (9.2.5) while F is the discretized objective. The function H has in Section 9.2 been defined within equation (9.2.12).

Proposition 9.1: *Let the relaxation function ϑ satisfy the assumptions of Theorem 9.1. Then the following statements hold:*

(i) *The function H is totally differentiable with respect to the arguments (α, β) and the associated total derivative functions*

$$\frac{\mathrm{d}H}{\mathrm{d}\alpha} := \frac{\partial H}{\partial \alpha} + \frac{\partial H}{\partial G}\frac{\partial G}{\partial \alpha} \quad \textit{and} \quad \frac{\mathrm{d}H}{\mathrm{d}\beta} := \frac{\partial H}{\partial G}\frac{\partial G}{\partial \beta}$$

are locally Lipschitz continuous. In addition, we have $\frac{\mathrm{d}H}{\mathrm{d}\alpha}(\alpha, 0) = \frac{\partial H}{\partial \alpha}(\alpha, 0)$ as well as $\frac{\mathrm{d}H}{\mathrm{d}\beta}(\alpha, 0) = 0$.

(ii) *The derivative of $H(\cdot, G(\cdot, \cdot))$ with respect to its first argument, denoted by $\frac{\partial H}{\partial \alpha}$, is totally differentiable in (α, β) and the total derivative function $\frac{\mathrm{d}}{\mathrm{d}(\alpha,\beta)}\frac{\partial H}{\partial \alpha}$ is locally Lipschitz continuous. In addition, we have $\frac{\mathrm{d}}{\mathrm{d}\alpha}\frac{\partial H}{\partial \alpha}(\alpha, 0) = \frac{\partial^2 H}{\partial \alpha^2}(\alpha, 0)$ as well as $\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial H}{\partial \alpha}(\alpha, 0) = 0$.*

Proof: We use the notation $X := \left(X_0^T, \ldots, X_{m-1}^T\right)^T$. The derivative of H can now be reduced to the derivatives of X as we have

$$\frac{\mathrm{d}H}{\mathrm{d}\alpha} = \frac{\partial H}{\partial \alpha} + \sum_{i=0}^{m-1} \frac{\partial H}{\partial \gamma_i}\frac{\mathrm{d}\gamma_i}{\mathrm{d}\alpha} = \frac{\partial H}{\partial \alpha} - \sum_{i=0}^{m-1} \frac{\partial X}{\partial \gamma_i}\frac{\mathrm{d}\gamma_i}{\mathrm{d}\alpha} \tag{9.4.2}$$

as well as

$$\frac{\mathrm{d}H}{\mathrm{d}\beta} = \sum_{i=0}^{m-1} \frac{\partial H}{\partial \gamma_i}\frac{\mathrm{d}\gamma_i}{\mathrm{d}\beta} = -\sum_{i=0}^{m-1} \frac{\partial X}{\partial \gamma_i}\frac{\mathrm{d}\gamma_i}{\mathrm{d}\beta}, \tag{9.4.3}$$

where the existence and local Lipschitz continuity of the derivatives $\frac{\partial X}{\partial \gamma_i}$ is for all indices $i \in \{0, \ldots, m-1\}$ guaranteed by Theorem 9.1. In particular, Theorem 9.1 guarantees that the derivatives of the form $\frac{\partial X}{\partial \gamma_i}$ in equations (9.4.2) and (9.4.3) vanish at every consistent initialization point, i.e., we have $\frac{\mathrm{d}H}{\mathrm{d}\alpha}(\alpha, 0) = \frac{\partial H}{\partial \alpha}(\alpha, 0)$ as well as $\frac{\mathrm{d}H}{\mathrm{d}\beta}(\alpha, 0) = 0$ and the statement (i) is proven.

In the next step we use Corollary 9.1 to show that also the Lipschitz continuous derivatives

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\frac{\partial H}{\partial \alpha} = \frac{\partial^2 H}{\partial \alpha^2} - \sum_{i=0}^{m-1} \frac{\partial^2 X}{\partial \alpha \partial \gamma_i}\frac{\mathrm{d}\gamma_i}{\mathrm{d}\alpha} \tag{9.4.4}$$

$$\text{and} \quad \frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial H}{\partial \alpha} = -\sum_{i=0}^{m-1} \frac{\partial^2 X}{\partial \alpha \partial \gamma_i}\frac{\mathrm{d}\gamma_i}{\mathrm{d}\beta} \tag{9.4.5}$$

exist. In particular, it follows from Corollary 9.1 that we have $\frac{\mathrm{d}}{\mathrm{d}\alpha}\frac{\partial H}{\partial \alpha}(\alpha, 0) = \frac{\partial^2 H}{\partial \alpha^2}(\alpha, 0)$ as well as $\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial H}{\partial \alpha}(\alpha, 0) = 0$, which leads to the statement (ii). □

Note that an important consequence of the above proposition is that all functions in the optimization problem (9.4.1) are continuously differentiable. i.e., we can formulate first order KKT conditions.

In the following, we regard the optimization problem (9.4.1) independent of our DAE context. Although we might of course still have relaxed DAEs in mind, we will from now on only require that the functions F and G are twice continuously differentiable, while the function H can be any function which satisfies the properties (i) and (ii) of Proposition 9.1.

Lemma 9.3: *Let F and G be continuously differentiable while H satisfies the properties (i) and (ii) of Proposition 9.1. Let (α, β) be a minimizer of problem* (9.4.1) *at which the matrix*

$$\begin{pmatrix} \frac{\mathrm{d}G}{\mathrm{d}\alpha}(\alpha,\beta) & \frac{\mathrm{d}G}{\mathrm{d}\beta}(\alpha,\beta) \\ \frac{\partial H}{\partial \alpha}(\alpha, 0) & 0 \end{pmatrix} \tag{9.4.6}$$

has full rank. Then (α, β) is a KKT point and there exist multipliers $\lambda \in \mathbb{R}^{n_G}$ and $\mu \in \mathbb{R}^{n_H}$ such that the following conditions are satisfied:

$$0 = \frac{\mathrm{d}F}{\mathrm{d}\alpha}(\alpha,\beta) + \lambda^T \frac{\mathrm{d}G}{\mathrm{d}\alpha}(\alpha,\beta) + \mu^T \frac{\partial H}{\partial \alpha}(\alpha, G(\alpha,\beta)) \tag{9.4.7}$$

$$0 = \frac{\mathrm{d}F}{\mathrm{d}\beta}(\alpha,\beta) + \lambda^T \frac{\mathrm{d}G}{\mathrm{d}\beta}(\alpha,\beta) \tag{9.4.8}$$

$$0 = G(\alpha,\beta) \tag{9.4.9}$$

$$0 = H(\alpha, G(\alpha,\beta))\,. \tag{9.4.10}$$

Proof: If the point (α, β) is a minimizer of problem (9.4.1) it must satisfy the feasibility condition $G(\alpha, \beta) = 0$. Consequently, we can use the properties (i) and (ii) of Proposition 9.1 to conclude that the matrix

$$\begin{pmatrix} \frac{\mathrm{d}G}{\mathrm{d}\alpha}(\alpha,\beta) & \frac{\mathrm{d}G}{\mathrm{d}\beta}(\alpha,\beta) \\ \frac{\mathrm{d}H}{\mathrm{d}\alpha}(\alpha, G(\alpha,\beta)) & \frac{\mathrm{d}H}{\mathrm{d}\beta}(\alpha, G(\alpha,\beta)) \end{pmatrix} = \begin{pmatrix} \frac{\mathrm{d}G}{\mathrm{d}\alpha}(\alpha,\beta) & \frac{\mathrm{d}G}{\mathrm{d}\beta}(\alpha,\beta) \\ \frac{\partial H}{\partial \alpha}(\alpha, 0) & 0 \end{pmatrix}$$

has full rank, i.e., the linear independence constraint qualification (LICQ) is satisfied at the minimizer (α, β). Thus, (α, β) is a KKT point. The corresponding first order necessary conditions are equivalent to the conditions (9.4.7)-(9.4.10). This can be verified by using the properties (i) and (ii) of Proposition 9.1 again. □

Remark 9.4: *An important observation in the above Lemma is that an evaluation of the KKT conditions* (9.4.7)-(9.4.10) *does not require the computation of the derivative of H with respect to β. The algorithm which is proposed in the following will make use of this observation.*

In order to numerically determine a KKT point $w := (\alpha, \beta, \lambda, \mu)$ satisfying the conditions (9.4.7)-(9.4.10) we plan to apply a Newton type method. For this aim, we start at an initial point w^0 generating iterates of the form $w^{k+1} := w^k + A(w^k)^{-1}\Phi(w^k)$, where the function $\Phi : \mathbb{R}^{n_w} \to \mathbb{R}^{n_w}$ (with $n_w := n_\alpha + n_\beta + n_G + n_H$) is defined as

$$\forall w \in \mathbb{R}^{n_w} : \quad \Phi(w) := \begin{pmatrix} \frac{\mathrm{d}F}{\mathrm{d}\alpha}(\alpha,\beta)^T + \frac{\mathrm{d}G}{\mathrm{d}\alpha}(\alpha,\beta)^T\lambda + \frac{\partial H}{\partial \alpha}(\alpha, G(\alpha,\beta))^T\mu \\ \frac{\mathrm{d}F}{\mathrm{d}\beta}(\alpha,\beta)^T + \frac{\mathrm{d}G}{\mathrm{d}\beta}(\alpha,\beta)^T\lambda \\ G(\alpha,\beta) \\ H(\alpha, G(\alpha,\beta)) \end{pmatrix}, \quad (9.4.11)$$

while the matrix valued function $A : \mathbb{R}^{n_w} \to \mathbb{R}^{n_w \times n_w}$ is defined as

$$A(w) := \begin{pmatrix} \frac{\mathrm{d}^2 L(\alpha,\beta,\lambda)}{\mathrm{d}\alpha^2} + \frac{\partial^2 \mu^T H}{\partial \alpha^2}(\alpha, G(\alpha,\beta)) & \frac{\mathrm{d}^2 L(\alpha,\beta,\lambda)}{\mathrm{d}\alpha \mathrm{d}\beta} & \frac{\mathrm{d}G(\alpha,\beta)}{\mathrm{d}\alpha}^T & \frac{\partial H}{\partial \alpha}(\alpha, G(\alpha,\beta))^T \\ \frac{\mathrm{d}^2 L(\alpha,\beta,\lambda)}{\mathrm{d}\beta \mathrm{d}\alpha} & \frac{\mathrm{d}^2 L(\alpha,\beta,\lambda)}{\mathrm{d}\beta^2} & \frac{\mathrm{d}G(\alpha,\beta)}{\mathrm{d}\beta}^T & 0 \\ \frac{\mathrm{d}G(\alpha,\beta)}{\mathrm{d}\alpha} & \frac{\mathrm{d}G(\alpha,\beta)}{\mathrm{d}\beta} & 0 & 0 \\ \frac{\partial H}{\partial \alpha}(\alpha, G(\alpha,\beta)) & 0 & 0 & 0 \end{pmatrix} \quad (9.4.12)$$

for all $w \in \mathbb{R}^{nW}$, where we use the notation

$$L(\alpha, \beta, \lambda) := F(\alpha, \beta) + \lambda^T G(\alpha, \beta) \ .$$

For the moment, we assume here that the matrix $A(w)$ is always invertible such that the iterations are well-defined, but we will discuss later in more detail, under which conditions this can be guaranteed.

Lemma 9.4: *Let F and G be twice continuously differentiable while H satisfies the properties (i) and (ii) of Proposition 9.1. The function Φ is a differentiable function and its Jacobian $\Phi' := \frac{\partial \Phi}{\partial w}$ is locally Lipschitz continuous on $\mathbb{R}^{n_w}$. Moreover, for every point $w^* \in \mathbb{R}^{n_w}$, which satisfies $\Phi(w^*) = 0$ we have $\Phi'(w^*) = A(w^*)$.*

Proof: The question whether the function Φ is differentiable reduces obviously to the question whether the derivatives

$$\frac{\mathrm{d}H}{\mathrm{d}\beta} \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial H}{\partial \alpha} \tag{9.4.13}$$

exist. As the existence and local Lipschitz continuity of both terms is guaranteed by the properties (i) and (ii) of Proposition 9.1, Φ is differentiable and its Jacobian is locally Lipschitz continuous. Now, we have

$$\Phi'(w^*) - A(w^*) = \begin{pmatrix} 0 & \mu^T \frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial H}{\partial \alpha}(\alpha^*, G(\alpha^*, \beta^*)) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{\mathrm{d}H}{\mathrm{d}\beta}(\alpha^*, G(\alpha^*, \beta^*)) & 0 & 0 \end{pmatrix} = 0 \ , \tag{9.4.14}$$

as we have $G(\alpha^*, \beta^*) = 0$ at every point w^* which satisfies $\Phi(w^*) = 0$. □

Remark 9.5: *The above Newton type method can of course also be interpreted as an SQP type method. As this method is neither based on exact Hessians nor on exact constraint Jacobians it can be regarded as an inexact SQP method. Note that the approximation of the Hessian and the constraint Jacobian has the advantage that no derivatives of the function H with respect to β are needed, which is especially beneficial in the case that we have a large number of algebraic states in the DAE. Moreover, the above SQP formulation can in an obvious way be transferred to the case that we also have inequalities.*

The final step of this section is to discuss the local convergence properties of the suggested inexact SQP method:

Theorem 9.2: *Let F and G be twice continuously differentiable while H satisfies the properties (i) and (ii) of Proposition 9.1. If the discretized DAE optimization problem* (9.4.1) *has a strict local minimum at the primal dual KKT point w^* while A is a regular function, then there exists an open neighborhood $\mathcal{N}(w^*)$ around $w^* \in \mathcal{N}(w^*)$ such that for all initial points $w^0 \in \mathcal{N}(w^*)$ the sequence (w^k) converges q-quadratically to w^*.*

Proof: The proof is based on the standard argumentation for Newton methods: Let w^* be a point which satisfies $\Phi(w^*) = 0$. Now, we have for all $k \in \mathbb{N}$

$$\begin{aligned} w^{k+1} - w^* &= w^k - w^* - A(w^k)^{-1}\Phi(w^k) \\ &= w^k - w^* - A(w^k)^{-1} \int_0^1 \Phi'(w^* + s\left(w^k - w^*\right))\left(w^k - w^*\right) \mathrm{d}s \\ &= A(w^k)^{-1} \int_0^1 \left[A(w^k) - \Phi'(w^* + s\left(w^k - w^*\right))\right]\left(w^k - w^*\right) \mathrm{d}s \ . \end{aligned} \tag{9.4.15}$$

We know from Lemma 9.4 that Φ' is locally Lipschitz continuous and $\Phi'(w^*) = A(w^*)$, i.e., there exists a constant $L < \infty$ such that for all $s \in [0,1]$ the inequality

$$\|A(w^k)^{-1}\left[A(w^k) - \Phi'(w^* + s\left(w^k - w^*\right))\right]\left(w^k - w^*\right)\| \ \leq \ L\,\|w^k - w^*\|^2$$

is satisfied. Thus, we find with equation (9.4.15) that

$$\|w^{k+1} - w^*\| \quad \leq \quad L\,\|w^k - w^*\|^2 \tag{9.4.16}$$

which shows the q-quadratic convergence of the method and leads immediately to the statement of the theorem. □

Remark 9.6:

(i) The above theorem assumes that the matrix valued function A is regular. However, it would also be enough to require that $A(w^*)$ is regular as this implies that there is a neighborhood of w^* in which A is regular. Moreover, it can easily be seen that $A(w^*)$

must be regular if the original unrelaxed optimization problem (9.2.4) satisfies the second order sufficient condition for a minimum at w^* while the constraints satisfy the linear independence constraint qualification (LICQ). (This follows immediately from the equivalence relations $\hat{H}(\cdot) \equiv H(\cdot, \beta^*)$, $\frac{\partial}{\partial \alpha}\hat{H}(\cdot) \equiv H_\alpha(\cdot, \beta^*)$, and $\frac{\partial^2}{\partial \alpha^2}\hat{H}(\cdot) \equiv H_{\alpha.\alpha}(\cdot, \beta^*)$.)

(ii) The above theorem requires that the functions F and G are twice continuously differentiable while H satisfies the properties (i) and (ii) of Proposition 9.1. Note that these requirements are satisfied, if F, the original DAE right-hand side functions f and g, as well as the constraint function r are twice continuously differentiable while the relaxation function ϑ satisfies the properties (i) and (iii) of Lemma 9.2.

(iii) Note that the proposed inexact SQP algorithm does not require the computation of any derivatives of the functions X_i with respect to the algebraic node variables β_i. Hence, the proposed inexact SQP algorithm is beneficial if we have many algebraic states.

(iv) One of the main advantages of the inexact SQP method which is proposed here is that it is easy to implement: we can simply take an existing SQP method combining it with our favorite DAE integrator by replacing the algebraic conditions with the relaxed version and stacking the consistency conditions to the equality constraints. At the place where we would usually call the integrator to compute the derivatives of the differential states with respect to the variables γ_i, we pretend that we have never introduced any relaxation ignoring this dependency. The above Theorem guarantees that this way of dealing with DAEs does not destroy the q-quadratic convergence of the SQP method.

(v) Note that several variations of the above method are possible: for example, instead of computing the Hessians exactly, we could replace them by an approximation based on BFGS updates.

(vi) Finally, we compare the above inexact SQP algorithm with the Partially Reduced Sequential Quadratic Programming method (PRSQP) for DAEs, developed by Leineweber [150]. The PRSQP method is an exact SQP method, but the computation of the state sensitivities and the SQP algorithm are deeply intertwined, such that at the end only

$$n_x + n_u + n_p + 1$$

forward directions are needed in every step of the optimization algorithm. The inexact SQP method proposed here requires the computation of $n_x + n_u + n_p$ forward

derivatives of the state trajectory in every step – i.e., we need 1 direction less, which is only a minor advantage. The main advantage of the inexact SQP method that is proposed here, is that it can easily be implemented. In fact, there exist a lot of open as well as commercial SQP implementations and there is also a number of DAE integrators available. As the PRSQP method requires a deep intertwining of the SQP routine and the sensitivity generation, it is not so easily possible to couple or exchange existing software modules. In contrast, the inexact SQP method proposed here can deal with existing integrators and SQP methods, as it is explained above.

(vii) Note that the above method can easily be combined with Lifted Newton methods [3] for the case that we have both many differential and many algebraic states but only a moderate number of control inputs and parameters.

9.5 Numerical Test Examples

In this section, we address the question whether the proposed inexact SQP method does also work in practice. In the previous section, we have proven the theoretical properties of the proposed method summarized in Theorem 9.2, which states that we can expect q-quadratic convergence of the method. However, we should also ask some critical questions that have only partially been addressed by our theoretical results: how should we choose the design parameters a and b associated with the relaxation function ϑ

$$\vartheta(\gamma,\tau) := \begin{cases} \gamma\ (1-\tau)^{\frac{a+b\|\gamma\|}{\|\gamma\|}} & \text{if } \|\gamma\| \neq 0 \\ \\ 0 & \text{otherwise} \end{cases},$$

which has been defined in equation (9.3.1)? Does the choice of a influence the step size control of an integration routine? How does the inexact SQP algorithm behave if we choose a too small? In order to understand the relevance of these questions, recall the estimate (9.3.18) and (9.3.19). From these estimates we do only know that the Lipschitz constant of the functions X_i' at a KKT point is small if a is large, while the slope of the function ϑ with respect to the time t is expected to be small for small a. Intuitively, we expect that the Lipschitz constant of the functions X_i' will influence the convergence behavior of the SQP method while the slope of the function ϑ might have an influence on the step size control of the integration routine.

In order to give at least an empirical answer to these questions we will test the inexact SQP method by applying it first to a small toy example in order to understand the behavior

of the method. And second, we apply the method to a large real-world optimal control problem for a distillation column.

A small toy example

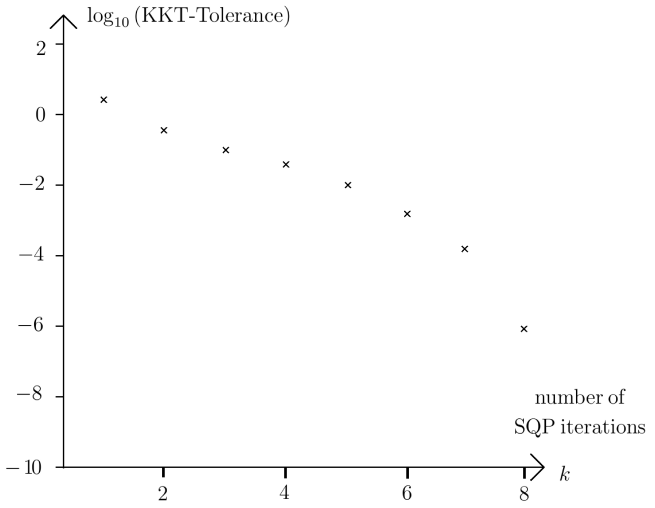
As a simple example for a DAE optimization, we consider the following optimal control problem

$$\boxed{\begin{array}{lrcl} \underset{x(\cdot),z(\cdot),u(\cdot)}{\text{minimize}} & & & \int_0^5 x(t)^2 + 3u(t)^2 \,\mathrm{d}t \\ \text{subject to:} & & & \\ \forall t \in [0,5]: & \dot{x}(t) & = & x(t)\left(\frac{1}{2}x(t) - 1\right) + u(t) + \frac{1}{2}z(t) \\ \forall t \in [0,5]: & 0 & = & z(t) + e^{z(t)} + x(t) - 1 \\ & 0 & = & x(0) - 1 \end{array}} \tag{9.5.1}$$

In this example we have only 1 differential and 1 algebraic state. However, the differential as well as the algebraic equation are nonlinear. For our numerical test, we choose 10 multiple shooting intervals. Moreover, we use the ACADO BDF integrator [131], which is based on a backward differentiation formula (BDF) combined with a higher order diagonal implicit Runge-Kutta starter based on the algorithmic ideas in [15]. We also use the ACADO implementation of multiple shooting SQP methods.

Using the notation from the previous section, we start the Newton- or SQP method with $w^0 = 0$, i.e., all states and controls are simply initialized with 0. As a stopping criterion we require the KKT tolerance to be less than 10^{-6}. Now, we use $a = \frac{1}{2}$ and $b = 1$ within the relaxation function (9.3.1). For this these settings, the inexact SQP method converges rapidly within 8 iterations:

The corresponding minimum value of the objective is 0.49581. Here, it should be remarked that we did not use any globalization technique, i.e., the full-step method was convergent in the above example, although the starting guess $w^0 = 0$ is actually quite far from the optimal solution. For the case that the method is not convergent in full-step mode it is e.g. possible to apply trust region methods. However, such globalization techniques for inexact SQP methods are not in the scope of this chapter and we refer to [234] for further reading.



k	1	2	3	...	7	8
KKT-Tol:	$3.67 \cdot 10^{0}$	$3.46 \cdot 10^{-1}$	$6.00 \cdot 10^{-2}$	...	$1.79 \cdot 10^{-4}$	$7.59 \cdot 10^{-7}$

Figure 9.2: The KKT tolerance associated with the iterations $w^{k+1} = w^k - A(w^k)^{-1}\Phi(w^k)$ of the inexact SQP method applied to the DAE optimal control problem (9.5.1).

We are interested in a discussion on how the method behaves with regard to the design parameter a in the relaxation function. Thus, we test the inexact SQP method for several values of a:

In Table 9.5 the total CPU time of the inexact SQP iterations until convergence is listed. This CPU time is in our example dominated by the time that is needed for the simulation of the DAE as well as for first and second order sensitivity generation, while the solution of the sub-QPs takes usually less than 1 ms. If we choose a too large, the slope (9.3.19) of the relaxation is too high such that the BDF integration routine takes many steps at the start of each interval, which is equivalent to a phase 1 step satisfying the algebraic consistency conditions first before the step size can be increased. However, as soon as we choose a less than 10, we observe that the overall CPU time improves, while we need the same number of SQP iterations. The CPU time achieves its minimum if we choose for a approximately 0.5. If we reduce a further, the approximations of the Jacobian and the Hessian become less accurate such the CPU time increases again - due to the fact that

a	# SQP iterations	total CPU time
50	7	204 ms
10	7	203 ms
5	7	181 ms
1	7	155 ms
0.5	8	137 ms
0.1	10	149 ms
0.05	11	170 ms
0	12	181 ms

Table 9.1: The number of SQP iterations and the associated total CPU time of the inexact SQP method for several values of the design parameter a.

we need more SQP iterations. It is interesting to observe that even for $a = 0$ the method is still convergent. In fact, even if we choose a too small the method is still faster than an exact SQP method, which uses an expensive phase 1 step in the integration routine (here realized via a large relaxation parameter a).

Optimal control of a distillation column

In this section we consider a DAE model for a distillation column taken from the literature [68]. This DAE model has 82 differential states, 122 algebraic states as well as 2 time dependent control inputs. We will not restate all the equations that are needed to build up the model of the distillation column, but we just mention that the first 42 differential states represent molar Methanol concentrations in the reboiler and condenser of the distillation column while the remaining 40 differential states are the molar tray holdups. The liquid and vapor molar fluxes (40 each) together with the 42 temperatures of reboiler, the 40 trays, and the condenser are the algebraic state vector, i.e., we have $n_z = 40 + 40 + 42 = 122$ algebraic states. However, for the details of the model we refer to [68]. The optimal control problem of our interest is also taken from [68]. We consider

a least square problem of the form

$$\boxed{\begin{aligned} &\underset{x(\cdot),z(\cdot),u(\cdot)}{\text{minimize}} \quad \int_0^T \left\| \widetilde{T} z(t) - T_{\text{ref}} \right\|^2 + \| R\,(u(t) - u_{\text{S}}) \|^2 \;\mathrm{d}t \\ &\text{subject to:} \\ &\forall t \in [0,T]: \quad \dot{x}(t) = f(t, x(t), z(t), u(t)) \\ &\forall t \in [0,T]: \quad 0 = g(t, x(t), z(t), u(t)) \\ &\qquad\qquad\quad x(0) = x_0 \end{aligned}}\,, \tag{9.5.2}$$

where T_{ref} is the reference temperature, at which we would like to operate the system, $\widetilde{T}$ a projection matrix which filters the temperatures out of the algebraic state vector that we would like to penalize, u_{S} a control set-point, and x_0 a given initial state (e.g. from a measurement). Note that this problem is exactly coinciding with the optimization problem (7.22) in [68]. Here, we chose $T = 1000\,\text{s}$ to be the time horizon while the 6 multiple shooting discretization intervals are also chosen as suggested in [68].

We employ the same numerical settings as for the toy example from the previous section, i.e., we use the ACADO toolkit BDF integrator and multiple-shooting SQP implementation [131] to implement the inexact SQP method for the distillation column problem (9.5.2). The problem turns out to be only mildly nonlinear, such that the method converges for $a = 0.5$ in 5 iterations, which corresponds to approximately $30\,\text{s}$ of computation time:

k	1	2	3	4	5
KKT-Tol:	$2.252 \cdot 10^{0}$	$4.230 \cdot 10^{-1}$	$7.34 \cdot 10^{-3}$	$2.32 \cdot 10^{-5}$	$1.52 \cdot 10^{-9}$

Table 9.2: The KKT tolerance associated with the iterations of the inexact SQP method applied to a real-world optimal control problem with 82 differential and 122 algebraic states.

In this example the choice of the parameter a did not influence the number of SQP iterations, i.e., we need always 5 SQP iterations. However, for $a = 0.5$ or smaller values of a the BDF integration routine needs approximately 119 integrator steps per simulation and interval while values $a \gg 1$ lead to approximately 145 steps, i.e., we save approximately 20% of computation time in comparison to an exact SQP method with phase 1 step.

Summarizing our observations, the inexact SQP algorithm proposed in this chapter is surprisingly robust with respect to the choice of the design parameter a. Choosing a close to 0.5 turned out to lead to significant savings in terms of computation time.

Chapter 10

Approximate Robust Optimization of a Biochemical Process

In this chapter we present techniques to optimize open-loop stable periodic stationary states of processes that depend on uncertain parameters. We start by recalling approximate robust counterpart formulations but specialize on automatic backward differentiation strategies which are especially beneficial for systems with many uncertain parameters and a small number of inequality constraints. The presented approximate robust programming formulation has an interesting application for stable time-periodic systems where the steady state is affected by uncertainties. In order to demonstrate this, we apply our techniques to a fermentation process optimal in a periodic operation. We discuss this optimal periodic solution and robustify it with respect to unknown model parameters. Note that this chapter is based on joint work with Dr. Filip Logist. The corresponding paper appeared in the proceeding of the 48th Conference on Decision and Control [133].

10.1 Introduction

In this chapter we focus on uncertain dynamic systems where we have on the one hand a large number of uncertain parameters but on the other hand only a small number of inequality constraints that should robustly be satisfied. For this aim, we start in Section 10.2 with an introduction of the basic notation which is needed for robust counterpart formulations for the case that the uncertainty enters via an implicit equation.

Moreover, in Section 10.3, we transfer this formulation and some ideas, which have originally been proposed in [71], to uncertain stable periodic systems where a periodic stationary state has to be optimized. This periodic stationary state is in our consideration depending on both open-loop control inputs and unknown time-invariant parameters. Thus, we are interested in a robust optimization of the cyclic stationary state.

In Section 10.4 we introduce the model of a biochemical fermentation process. We discuss optimal periodic operation modes for this model which can be realized by an application of a time-varying open-loop control input as the system turns out to be asymptotically stable. Here, the average productivity is maximized for a given average amount of substrate feed input. In Section 10.5 we show that this nominally optimal solution needs to be refined if the parameters are not exactly known. The optimal robustified solution for the periodic operation requires a significantly different control input in order to guarantee robustness with respect to the uncertainties.

10.2 Approximate Robust Optimization with Implicit Dependencies

Let us consider an uncertain nonlinear optimization problem of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^{n_x},u\in R^{n_u}} \quad & F_0(x,u) \\ \text{subject to} \quad & G(x,u,w)=0 \\ & F_i(x,u)\leq 0 \quad \text{for all } i\in\{1,\dots,n\} \end{aligned} \tag{10.2.1}$$

with continuously differentiable functions

$$F_0, F_1, \dots, F_n : \mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\to\mathbb{R}$$

depending on an optimization variable $u\in\mathbb{R}^{n_u}$. Here, we assume that the variable x is implicitly defined by the continuously differentiable equality constraint

$$G : \mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\times\mathbb{R}^{n_w}\to\mathbb{R}^{n_x}\ ,$$

where the partial derivative function $\frac{\partial G}{\partial x}$ is assumed to be regular on its domain. This requirement ensures that x can at least locally be eliminated from the optimization problem such that the components of u are the remaining degrees of freedom for the optimization.

In the following we regard the case that G does not only depend on x and u but also on an uncertain parameter $w \in W \subset R^{n_w}$ lying in an uncertainty set W which has ellipsoidal form

$$W := \left\{ w \in R^{n_w} \mid (w - \overline{w})^T \Sigma^{-1} (w - \overline{w}) \leq 1 \right\} . \tag{10.2.2}$$

where $\Sigma \in R^{n_w \times n_w}$ is a positive definite scaling matrix and $\overline{w} \in \mathbb{R}^{n_w}$ a constant. However, for the theoretical part, we will assume that we have $\Sigma = 1$ and $\overline{w} = 0$ which can always be achieved by shifting and rescaling the uncertainty w. Note that in our notation the functions $F_0, F_1, \ldots, F_n$ are not allowed to explicitly depend on w, which is however not a restriction as such a dependence can always be eliminated by a suitable definition of x and G. Note that the only difference to notation in previous chapters of this thesis is that x is implicitly defined by an equality constraint.

In order to incorporate the uncertainty into the optimization problem, we formulate the associated robust counterpart which is again analogous to previous chapters. More precisely, we assume that whatever u the optimizer chooses, the adverse player "nature" chooses the worst possible value $V_i(u)$ defined by

$$V_i(u) := \max_{w,x} F_i(x,u) \quad \text{s.t.} \quad \begin{cases} G(x,u,w) = 0 \\ w \in W \end{cases} . \tag{10.2.3}$$

Our aim is now to solve the associated worst-case minimization problem

$$\begin{aligned} \min_{u \in \mathbb{R}^{n_u}} \quad & V_0(u) \\ \text{subject to} \quad & V_i(u) \leq 0 \quad \text{for all } i \in \{1, \ldots, n\} . \end{aligned} \tag{10.2.4}$$

In this chapter, we do not aim at a rigorous robustification strategy which has been introduced in Chapter 3. Rather, we employ the approximate strategies from [71, 123, 174], where it has been suggested to replace the functions $V_i(u)$ by approximations $\widetilde{V}_i(\overline{x}, u)$ (for all $i \in \{1, \ldots, n\}$) which are obtained by a linearization technique without taking higher order terms into account. For this aim, the functions G and F_i are for all $i \in \{1, \ldots, n\}$ linearized around a reference $\overline{x} \in \mathbb{R}^{n_x}$ satisfying

$$G(\overline{x}, u, 0) = 0 .$$

Now the approximation $\widetilde{V}_i(\overline{x}, u)$ is defined by

$$\begin{aligned}\widetilde{V}_i(\overline{x}, u) &:= \max_{w_i, \xi_i} F_i(\overline{x}, u) + \frac{\partial F_i(\overline{x}, u)}{\partial x}\xi_i \\ &\qquad \text{s.t.} \begin{cases} \frac{\partial G(\overline{x}, u, 0)}{\partial x}\xi_i + \frac{\partial G(\overline{x}, u, 0)}{\partial w} w_i = 0 \\ w_i^T w_i \leq 1 \end{cases} \\ &= F_i(\overline{x}, u) + \left\| \frac{\partial F_i}{\partial x} \left(\frac{\partial G}{\partial x}\right)^{-1} \frac{\partial G}{\partial w} \right\|_2 , \end{aligned} \tag{10.2.5}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. For the last transformation we have explictly solved the linearized maximization problem. In [71] several approaches have been presented on how to numerically deal with the appearance of the inverse $\left(\frac{\partial G}{\partial x}\right)^{-1}$ and in [19, 20] the above explicit solution is discussed under the more general assumption that W is an intersection of a finite number of ellipsoids.

However, we like to specialize on the computation of the derivatives in equation (10.2.5) for the case that n is small, i.e., we have only very few uncertain constraints while the number n_w of uncertain parameters might be very large. In this case it is advisable to use automatic differentiation in the adjoint mode to evaluate the margin terms of the form

$$\left\| \frac{\partial F_i}{\partial x} \left(\frac{\partial G}{\partial x}\right)^{-1} \frac{\partial G}{\partial w} \right\|_2 = \left\| \mu_i^T \frac{\partial G}{\partial w} \right\|_2 \tag{10.2.6}$$

(for $i \in \{0, \ldots, n\}$), where the backward (or adjoint) seed parameters $\mu_0, \ldots, \mu_n \in \mathbb{R}^{n_x}$ are well-defined by linear equations of the form

$$\mu_i^T \frac{\partial G}{\partial x} = \frac{\partial F_i}{\partial x} . \tag{10.2.7}$$

Summarizing $\mu := (\mu_0, \ldots, \mu_n) \in \mathbb{R}^{n_x \times (n+1)}$ we can formulate the approximate robust counterpart problem in the form

$$\begin{aligned} \min_{\overline{x},u,\mu} \quad & F_0(x,u) + \left\| \mu_0^T \frac{\partial G}{\partial w} \right\|_2 \\ \text{s.t.} \quad & G(\overline{x}, u, 0) = 0 \\ & F_i(x,u) + \left\| \mu_i^T \frac{\partial G}{\partial w} \right\|_2 \leq 0 \quad \text{f. a. } i \in \{1, \ldots, n\} \\ & \mu^T \frac{\partial G}{\partial x} - \frac{\partial F}{\partial x} = 0 \,. \end{aligned} \tag{10.2.8}$$

In this general form, the above nonlinear optimization problem can be interpreted as a nonlinear second order cone program (SOCP). However, we should be aware of the fact that we consider only a linear approximation here, i.e., for the case that e.g. $\frac{\partial G}{\partial w}$ is vanishing in the optimal solution, our approximation is obviously too optimistic as higher order terms might dominate the linear approximation even for small uncertainties - this is a known general drawback of linear approximation techniques. On the other hand, if the terms of the form $\mu_i^T \frac{\partial G}{\partial w}$ are for all $i \in \{0, \ldots, n\}$ not equal to zero we can at least guarantee that the approximation is valid for sufficiently small uncertainty sets W. In this case the norms in the above formulation are also differentiable in a neighborhood of the optimal solution.

10.3 Robustified Optimal Control for Periodic Processes

In this section we apply the considerations from the previous section for the case that we like to optimize the stationary state of a stable time periodic dynamic system. Let $y : \mathbb{R} \to \mathbb{R}^{n_y}$ be the differential state of the system

$$\begin{aligned} \forall t \in [0, \infty) : \qquad & \dot{y}(t) = g(y(t), v(t), w) \\ & y(0) = y_0 \end{aligned} \tag{10.3.1}$$

where $v : \mathbb{R} \to \mathbb{R}^{n_u}$ is a time dependent but periodic control input satisfying $v(t) = v(t+T)$ for a cycle duration $T > 0$ and $w \in \mathbb{R}^{n_w}$ a time-constant but unknown parameter.

Now we assume that we have the additional knowledge about the system that the state y converges (independent of the initialization y_0) for $t \to \infty$ to a time-periodic limit cycle

$z : \mathbb{R} \to \mathbb{R}^{n_y}$ satisfying

$$\begin{aligned} \forall t \in [0,T] : \qquad & \dot{z}(t) = g(z(t), v(t), w) \\ & z(0) = z(T) \ . \end{aligned} \tag{10.3.2}$$

We are now interested in the behavior of this limit cycle in dependence on the periodic open-loop control input v but also on the uncertain parameter w.

In order to transfer the ideas from the previous section, we consider an uncertain periodic optimal control problem of the following form:

$$\boxed{\begin{array}{ll} \underset{z(\cdot),v(\cdot)}{\text{minimize}} & f_0(z(T)) \\ \text{subject to:} & \\ \forall t \in [0,T] : & \dot{z}(t) = g(z(t), v(t), w) \\ \forall i \in \{1,\ldots,n\} : & 0 \geq f_i(z(T)) \\ & z(0) = z(T) \end{array}} \tag{10.3.3}$$

To discretize this problem we replace the function v by a piecewise constant approximation

$$\widetilde{v}(t) := \sum_{i=0}^{N-1} u_i I_{[t_i,t_{i+1}]}(t) \ ,$$

where $I_{[a,b]}(t)$ is equal to 1 if $t \in [a,b]$ and equal to 0 otherwise. The time sequence $0 = t_0 < t_1 < \ldots < t_N = T$ can e.g. be equidistant. In the following we summarize

$$u := \left(u_0^T, \ldots, u_{N-1}^T\right)^T$$

to achieve a convenient notation.

In the next step, we regard $z(T) = Z(u, w, x)$ as a function depending on the control input u, the uncertain parameter w, as well as the initial value $z(0) = x$. In other words, Z is the solution operator of the differential equation, which can numerically be evaluated by using an integrator.

As we have $x = z(0) = z(T)$ we define $F_i(x) := f_i(z(T))$ for all $i \in \{0,\ldots,n\}$. Finally, the periodic boundary condition can be written as

$$G(x,u,w) := Z(u,w,x) - x = 0 \ . \tag{10.3.4}$$

Obviously, the functions $F_0, \ldots, F_n$ and G are now defined in such a way that the discretized version of problem (10.3.3) takes the form (10.2.1). Thus, also the associated robust counterpart formulation (10.2.8) transfers immediately.

Note that for the computation of the terms $\mu^T \frac{\partial G}{\partial x}$ and $\mu^T \frac{\partial G}{\partial w}$ automatic differentiation in backward mode can be used:

$$\mu^T \left(\frac{\partial G}{\partial x} \, , \, \frac{\partial G}{\partial w} \right) = \mu^T \frac{\partial Z}{\partial (x, w)}(u, w, x) - \mu^T \left(\mathbb{I} \, , \, 0 \right) \tag{10.3.5}$$

For the numerical evaluation of this expression we can use an integrator which is able to store intermediate values during the forward evaluation of Z such that the associated adjoint variational equation can later be solved by a backward run. However, for the details of adjoint differentiation for differential equations we refer the reader to [2].

10.4 Periodic Optimal Control of a Biochemical Process

In this section we apply the approximate robust formulation of periodic optimal control problems to a biochemical process. More precisely, we consider the following model of continuous culture fermentation which is often used in the literature [1, 185, 202]:

$$\begin{aligned} \dot{X}(t) &= -DX(t) + \mu(t)X(t) \\ \dot{S}(t) &= D(S_{\mathsf{f}}(t) - S(t)) - \frac{\mu(t)X(t)}{Y_{x/s}} \\ \dot{P}(t) &= -DP + (\alpha\mu(t) + \beta)X(t) \end{aligned} \tag{10.4.1}$$

This model consists of 3 states: here, X denotes the biomass concentration, S the substrate concentration, and P the product concentration of a continuous fermentation process. Furthermore, the process can be controlled by the input $S_{\mathsf{f}} : \mathbb{R} \to \mathbb{R}$ representing the feed substrate concentration. While the dilution rate D, the biomass yield $Y_{x/s}$, and the product yield parameters α and β are assumed to be constant and thus independent of the actual operating condition, the specific growth rate $\mu : \mathbb{R} \to \mathbb{R}$ of the biomass is a function of the states:

$$\mu(t) = \mu_{\mathsf{m}} \frac{\left(1 - \frac{P(t)}{P_{\mathsf{m}}}\right) S(t)}{K_{\mathsf{m}} + S(t) + \frac{S(t)^2}{K_{\mathsf{i}}}} \tag{10.4.2}$$

This specific growth rate equation is constructed to allow a description of both the product and the substrate inhibition. For the product an associated saturation constant P_{m} has been introduced while K_{m} denotes a saturation constant associated with the substrate. The constant K_{i} is the substrate inhibition constant and μ_{m} can be interpreted as the maximum specific growth rate.

In the next step we consider the following optimal control task for our fermentation model: our aim is to maximize the average productivity

$$Q := \frac{1}{T}\int_0^T DP(\tau)\,\mathrm{d}\tau$$

for a given amount of substrate $\overline{S}_{\mathsf{f}}$. It has already been suggested in [185, 202] that this aim can efficiently be achieved by operating the system in a periodic mode. The corresponding optimal control problem takes the form

$$\boxed{\begin{array}{ll} \min\limits_{z(\cdot),S_{\mathsf{f}}(\cdot)} & \frac{1}{T}\int_0^T DP(\tau)\,\mathrm{d}\tau \\ \text{subject to:} & \\ \forall t\in[0,T]: & \dot{z}(t) = g(z(t),S_{\mathsf{f}}(t),w) \\ & \frac{1}{T}\int_0^T S_{\mathsf{f}}(\tau)\,\mathrm{d}\tau = \overline{S}_{\mathsf{f}} \\ & z(0) = z(T) \\ & \dot{X}(0) = 0 \\ \forall t\in[0,T]: & S_{\mathsf{f}}^{\mathsf{min}} \le S_{\mathsf{f}}(t) \le S_{\mathsf{f}}^{\mathsf{max}} \\ & \frac{1}{T}\int_0^T X(\tau)\,\mathrm{d}\tau \le \overline{X}^{\mathsf{max}} \end{array}}\,. \tag{10.4.3}$$

Here, we have summarized the states into one differential state vector $z:\mathbb{R}\to\mathbb{R}^3$ given by

$$z := (X,S,P)^T$$

while the corresponding right-hand side of the differential equation (10.4.1) has been denoted by g. Moreover, the parameters are summarized in a vector $w\in\mathbb{R}^8$ given by

$$w := \Big(D, K_{\mathsf{i}}, K_{\mathsf{m}}, P_{\mathsf{m}}, Y_{x/s}, \alpha, \beta, \mu_{\mathsf{m}}\Big)^T\,, \tag{10.4.4}$$

while the corresponding nominal values for w, which are used in this section, are listed in Table 10.1.

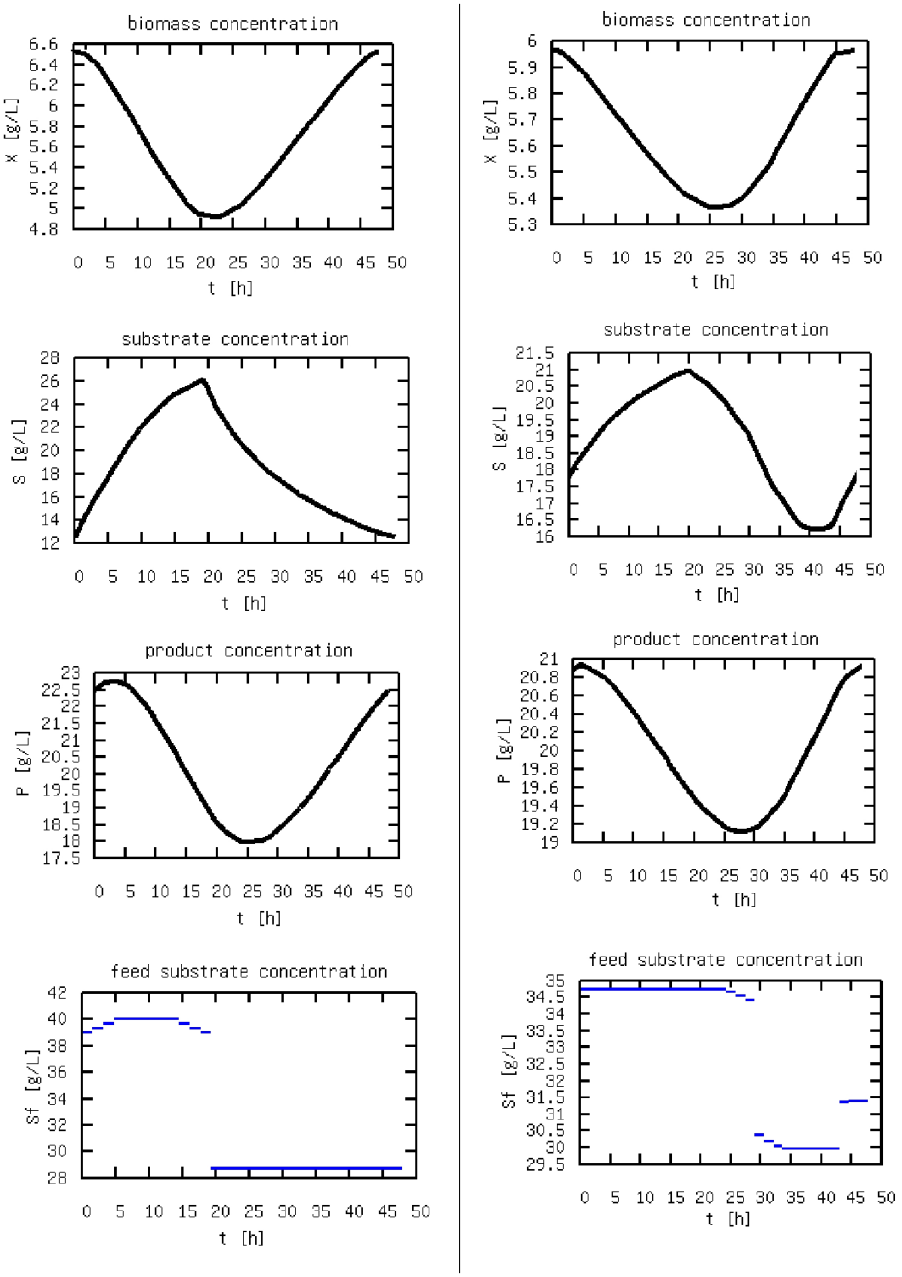


Figure 10.1: Left: A locally optimal result for the three states of the optimal control problem (10.4.3). Right: The approximately robust optimal periodic result for the three states together with the optimized control input.

Table 10.1: Nominal fermentation process parameters

Name	Symbol	Value
dilution rate	D	$0.15\,\mathsf{h}^{-1}$
substrate inhibition constant	K_i	$22\,\frac{\mathsf{g}}{\mathsf{L}}$
substrate saturation constant	K_m	$1.2\,\frac{\mathsf{g}}{\mathsf{L}}$
product saturation constant	P_m	$50\,\frac{\mathsf{g}}{\mathsf{L}}$
yield of the biomass	$Y_{x/s}$	0.4
first product yield constant	α	2.2
second product yield constant	β	$0.2\,\mathsf{h}^{-1}$
specific growth rate scale	μ_m	$0.48\,\mathsf{h}^{-1}$
average feed substrate	$\overline{S}_\mathsf{f}$	$32.9\,\frac{\mathsf{g}}{\mathsf{L}}$
minimum feed substrate	S_f^{min}	$28.7\,\frac{\mathsf{g}}{\mathsf{L}}$
maximum feed substrate	S_f^{max}	$40.0\,\frac{\mathsf{g}}{\mathsf{L}}$
maximum average biomass concentration	$\overline{X}^{\mathsf{max}}$	$5.8\,\frac{\mathsf{g}}{\mathsf{L}}$

Note that the optimization problem (10.4.3) additionally regards a maximum and a minimum bound ($S_\mathsf{f}^{\mathsf{min}}$ and $S_\mathsf{f}^{\mathsf{max}}$) on the input S_f as well as a constraint on the maximum average of the biomass concentration $\overline{X} := \frac{1}{T}\int_0^T X(\tau)\,\mathrm{d}\tau$ over one periodic cycle. Moreover, the equation $\frac{\mathrm{d}}{\mathrm{d}t}X(0) = 0$ has been introduced to remove the indefiniteness with regard to phase shifts. For the numerical solution of the periodic optimal control problem (10.4.3) we use the single shooting method with a piecewise constant parameterization (here 30 pieces) of the control input in combination with an Sequential Quadratic Programming (SQP) algorithm which has been implemented in the automatic control and dynamic optimization software ACADO (cf. Chapter 7). A corresponding locally optimal solution is shown in the left part of Figure 10.1.

It can be seen that the optimal result shows indeed a periodic behavior. In this example the time horizon was fixed to $T = 48\,\mathsf{h}$. The result for the objective in the optimal solution, which is shown in the left part of Figure 10.1, is given by

$$\frac{1}{T}\int_0^T DP(\tau)\,\mathrm{d}\tau = 3.11\frac{\mathsf{g}}{\mathsf{L\ h}}\ . \tag{10.4.5}$$

This value for the objective is clearly larger than the average productivity of $3.00\frac{\mathsf{g}}{\mathsf{L\ h}}$ which would be obtained in a time-constant steady state operation mode.

Moreover, the periodic process is open-loop stable as the spectral radius ρ of the monodromy matrix associated with the periodic process is $\rho \approx 0.003 < 1$ in the optimal solution. I.e., it is possible to start the fermentation process close to the shown periodic solution applying the optimal control $S_f(t)$ blindly without needing any feedback. This is not surprising as this is clearly what we would expect from such a process - independent of the specific control. Finally, we observe that the inequality constraint on the average biomass concentration

$$\frac{1}{T}\int_0^T DX(\tau)\,\mathrm{d}\tau = 5.73\frac{\mathrm{g}}{\mathrm{L}} \leq 5.8\frac{\mathrm{g}}{\mathrm{L}} \tag{10.4.6}$$

is not active in the optimal solution. In contrast to that, the inequality constraints for the input are almost all active: the optimal solution for S_f shows partially a bang-bang structure.

In the first phase, where the feed substrate is close or equal to the upper bound, we observe a substrate accumulation while the biomass concentration as well as the product concentration decrease. In the second phase, where the lower bound for the input is active, a growth of the biomass and, with a small delay, a growth of the product concentration can be seen.

10.5 Robust Optimization of a Biochemical Process

In the next step we are interested in the question what happens if the eight parameters stacked in the vector w are not exactly known but bounded by an ellipsoidal set W given by equation 10.2.2. Here, we choose a diagonal scaling matrix $\Sigma \in \mathbb{R}^{8\times 8}$, whose diagonal elements are given as:

$$\Sigma_{i,i} := \left|\frac{1}{20}\overline{w}_i\right|^2 \tag{10.5.1}$$

i.e., we regard 5% of uncertainty for each parameter. For the nominal parameter $\overline{w} \in \mathbb{R}^8$ we use the values from Table 10.1.

Now, we can solve the robustified optimal control problem of the form (10.2.8) which is associated with the periodic optimal control problem (10.4.3) from the previous section. Note that only one inequality constraint as well as the objective needs to be robustified in this example as the inequality bounds on the control input S_f are not affected by the

uncertainty. As we have eight uncertain parameters we are exactly in the situation where the adjoint mode of automatic differentiation is beneficial.

We use again the ACADO toolkit (cf. Chapter 7) to solve the robustified problem. For the integration an explicit Runge-Kutta integrator of order 7 (with step-size control of order 8) has been used to discretize the dynamic system. This integrator coming with ACADO toolkit is also suitable to compute first and second order derivatives in forward and backward mode with high accuracies. The corresponding numerical optimization results for the robustified problem are shown in the right part of Figure 10.1. In comparison to the nominal results, the biomass concentration X is, due to the robustified constraint, lower but shows still some cyclic behavior. The substrate concentration S as well as the product concentration P have in the robust solution a smaller amplitude. Finally, for S_f there are no active constraints anymore, but the solution shows still phases of accumulation.

The price that needs to be paid for the robustification can be discussed by an evaluation of the nominal average productivity in the optimal robustified solution:

$$Q^* \approx \frac{1}{T}\int_0^T DP(\tau)\,\mathrm{d}\tau = 2.98\frac{\mathsf{g}}{\mathsf{L\ h}}\ . \tag{10.5.2}$$

Comparing this result with the nominal result from equation (10.4.5) we find that we need to pay approximately $4-5\%$ of productivity if we regard the nominal amounts. Finally, we write the result for the robustified objective in the form

$$Q \approx (2.98 \pm 0.19)\,\frac{\mathsf{g}}{\mathsf{L\ h}}\ , \tag{10.5.3}$$

where the size of the worst case interval is given by the linear approximation $\|\frac{\mathrm{d}Q}{\mathrm{d}w}\Sigma^{\frac{1}{2}}\|_2 \approx 0.19\frac{\mathsf{g}}{\mathsf{L\ h}}$.

The main reason for the fact that the robustified optimal solution is, compared with the results from the previous section, significantly different, is that we have to keep a certain security distance with respect to the inequality constraint if the parameters are uncertain. Indeed, the inequality constraint of the form

$$\overline{X} + \left\| \frac{\mathrm{d}\overline{X}}{\mathrm{d}w}\Sigma^{\frac{1}{2}} \right\|_2 \leq 5.8\frac{\mathsf{g}}{\mathsf{L}} \tag{10.5.4}$$

was active in our example.

Finally, we note that in this small example the computation times are not critical: with the adjoint sensitivity generation the computations took approximately 2.0 ms per SQP

iteration, if we use a modern Desktop PC (Intel Pentium, 1.5GHz). In most situations, between 3 and 10 SQP iterations were necessary until an accuracy in the order of 10^{-6} is achieved depending on how close the initialization of the algorithm is to the optimal solution. Just to check that the adjoint mode is not only from a theoretical point of view advisable in our example we have also computed the sensitivities by using the forward mode of automatic differentiation, of course obtaining the same solution, but with the forward mode we need approximately 6.8 ms per SQP iteration. The reason for this difference in the computation times is that $n_w = 8$ forward directions need to be computed in contrast to only $n + 1 = 2$ backward directions that were needed for an evaluation of the model using the adjoint formulation. In any case, these computation times for our small example show that the method has a large potential to be scaled up for larger dynamic systems that are, e.g., arising in the field of chemical and biochemical engineering.

Chapter 11

Conclusions

This thesis led to three main contributions: first, we have developed formulations and algorithms which can deal with nonlinear min-max problems arising in the context of general robust optimization problems. Second, we have contributed with numerical techniques which can compute conservative estimates for the influence of uncertainty on nonlinear dynamic system. Here, numerical strategies for robust optimal control problems as well as stability optimization methods for periodic systems have been investigated. The third contribution is the implementation of optimal control algorithms within the freely available open-source software ACADO, which has successfully been tested with various numerical examples from the field of optimal control, model predictive control, optimization of differential algebraic equations, and robust optimization.

11.1 An Interpretation of the Developed Robust Optimization Methods

This thesis has introduced worst-case formulations and numerical solution strategies for optimization and control problems which are affected by unpredictable external disturbances, model errors, and other uncertainties. Recall that these numerical techniques were always based on the assumption that we succeed in modeling both: the system or the dynamic process of our interest as well as our knowledge about the uncertainties which are affecting it. However, in this thesis we have also learned that even if we succeed in finding a proper model for the process and the uncertainties, there is another challenge to be taken into

account: only for a small class of non-convex min-max optimization problems we know efficient numerical algorithms which are able to solve the problem globally and with a high numerical accuracy. This makes the modeling of uncertain systems challenging: on the one hand, we have to take care about a realistic mathematical representation of the real-world process and, on the other hand, we have to keep an eye on the numerical tractability of the corresponding robust optimization problem.

An important point, which has extensively been discussed in this thesis, is that if a non-convex robust optimization problem is not tractable in its exact version, it might nevertheless be solvable in a conservative approximation. Thus, if we want to model uncertain systems and successfully apply robust optimization techniques in practice, we have to be aware of the whole range of possibilities to employ trade-offs between numerical accuracy and computational tractability. At this point, the contribution of this thesis can be integrated and summarized as follows: we have developed formulations, new algorithms, and tools which have been designed for solving non-convex min-max optimization and min-max optimal control problems either exactly if this is possible with a reasonable amount of computation power or approximately – yet with mathematical guarantees – by exploiting suitable conservative approximations. In this sense, the contribution of this thesis has led to an extension of the scope and practical applicability of robust optimization in general.

A Review of Part I

Looking back at Chapter 2, ellipsoids have been found to be an important candidate for modeling uncertainty sets. Here, the use of ellipsoids or more general quadratic forms has been motivated in Section 2.2 where the S-procedure has been reviewed as a useful tool in convex robust optimization. Already at this early stage in the thesis, we encountered the outlined trade-off between accurate modeling and computational tractability: Theorem 2.1 states that the reformulation based on the S-procedure is exact if the uncertainty set is modeled by a single ellipsoid. However, for the case that we want to represent the uncertainty set more accurately, for example as an intersection of many ellipsoids, the corresponding Lagrangian based reformulation leads in general to a duality gap. Note that the review of existing techniques in Chapter 2 must be interpreted as a foundation for the more advanced Lagrangian based dual reformulation strategies in following chapters in Part I. In addition, the ellipsoid based set approximation strategies from Section 2.3 turned out to be the foundation for many of the computational techniques in Part II. Here, we

have in particular the results on the inner and outer approximations of sums of ellipsoids from Theorem 2.4 and 2.6 in mind.

Concerning the developments in Chapter 3, a general class of non-convex min-max problems has been considered. This is extending our possibilities for modeling uncertain systems. However, in contrast to the exact convexity based reformulation techniques from Chapter 2, the focus was rather on conservative approximations. In this context, an important observation was formulated in Lemma 3.2 where it has been shown that the proposed Lagrangian based dual reformulation strategy can in general be expected to yield better results than existing Taylor expansion based linearization strategies. Moreover, in Theorem 3.1 we have derived an upper bound on the duality gap, i.e., the level of conservatism which is introduced by reformulating the lower level maximization problems. This is an important result as it helps us to assess whether our overestimate of the impact of the uncertainties can be expected to be sufficiently accurate. In the Sections 3.3 and 3.4 of Chapter 3, we have discussed existing first and second order optimality conditions for nonlinear min-max problems as well as the relation to mathematical programming with complementarity constraints. This can be interpreted as a technical preliminary step which is needed to understand the structures in semi-infinite programming problems which have been exploited in the algorithms from Chapter 4.

Part I ends with an important contribution: the development of the sequential convex bilevel programming algorithm. Note that this algorithm has been motivated by comparing it to other possible numerical algorithms which could be based on standard sequential quadratic programming techniques. The main advantage of the algorithm is that it exploits the particular structure of nonlinear semi-infinite optimization problems, as the min-max nature of the problem is kept in the convex sub-problems which have to be solved sequentially. In Theorem 4.1 it has been shown that the algorithm can be expected to converge quadratically requiring the evaluation of first and second order derivatives only. This is in contrast to an application of exact Hessian sequential quadratic programming algorithms to a linearization based approximate formulation which would require at least third order derivatives of the model functions in order to obtain quadratic convergence. Moreover, the sub-problem in the sequential convex bilevel programming algorithms is convex as long as we work with positive semi-definite upper level Hessian approximations. The global convergence properties of the proposed algorithm have been investigated in Theorem 4.2. In addition, the applicability and performance of the method have been tested and illustrated in Section 4.5 where a numerical example is considered.

The Contributions of Part II

In Chapter 5 the propagation of uncertainty in dynamic systems has been analyzed aiming at numerical methods for computing robust positive invariance tubes. For linear systems, a main result was stated in Theorem 5.1, where we have discussed methods for ellipsoidal uncertainty tubes. This result can be interpreted as an extension of existing strategies which have originally been developed by Schweppe and Glover [102, 209] as well as by Kurzhanski and Varaiya [146, 144]. However, a main contribution of this chapter is discussed in Section 5.3, where techniques for the computation of robust positive invariant tubes for nonlinear dynamic systems are developed. These techniques are relevant, as first principle models of real-world dynamic processes are typically nonlinear. The corresponding analysis can be seen as a compromise of both: linear approximation of the system dynamics around a nominal or central trajectory and the careful treatment of nonlinear terms whose influence is overestimated. Here, the main result was stated in Theorem 5.3 where the computational technique has been elaborated. Note that the applicability of the approach has been illustrated with many examples including the extensive case study of the tubular reactor in Section 6.2.

Chapter 6 must be considered as one of the core chapters of this thesis, where the technical methods for computing robust positive invariant tubes have been employed to find tractable formulations of nonlinear robust optimal control problems which are based on a set valued notation. The main result of this chapter has been stated in Theorem 6.1, where guarantees on the conservatism of the proposed approximation approach are proven. Furthermore, the techniques have been extended to periodic systems. In Section 6.3 the relations between periodic robust positive invariant tubes and the existence of Lyapunov functions for periodic systems have been discussed. These considerations have led to Theorem 6.3, where a formulation of robust optimization problems for periodic systems as well as a technique for obtaining guarantees on the region of attraction of open-loop stable dynamic systems have been investigated. Part II ends in Section 6.4, where an application of the developed techniques to an open-loop controlled inverted spring pendulum has been shown.

The Implementation and Applications of Part III

The main contribution of Part III is the development of the optimal control software `ACADO Toolkit`. This implementation has been the basis for all the numerical results in

this thesis. Here, we recall that Part II has discussed how to cast robust optimal control problems in form of smooth standard optimal control problems which have to be solved numerically. The details of the implementation of this software have been presented in Chapter 7, where the scope and class structure of the tool are elaborated. Moreover, we discussed tutorial examples explaining the convenient symbolic syntax which is one of the basic features of ACADO and which enables us to use automatic differentiation, structure detection, symbolic optimization of mathematical expressions, and many other features which help the user to set up and solve optimal control problems efficiently. Here, we recall that the optimal control algorithms in ACADO are based on direct multiple shooting methods combined with various SQP algorithms.

One highlight of ACADO is its efficiency for small scale nonlinear model predictive control applications. In Chapter 8 we presented the implementation of automatic code export techniques. The numerical tests have shown a promising performance of the code being able to perform real-time Gauss-Newton iterations in much less than a millisecond. This has opened a new range of fast applications for nonlinear model predictive control – especially as the exported code can easily be installed on embedded hardware.

In Chapter 9 we presented an algorithm and extension of ACADO which exploits the structure of differential algebraic equations making use of automatic differentiation techniques and tailored algebraic relaxation techniques. The inexact SQP method which has been developed for this class of problems can be used to reduce the computational load of the sensitivity generation which is often the most expensive part in optimal control algorithms for large scale DAE systems. Note that the method has successfully been applied to the optimization of a distillation column with 82 differential and 122 algebraic states illustrating the performance of the method and its implementation.

Finally, Part III ends with an application of robust optimization techniques to a biochemical reaction. The main point of Chapter 10 was that the periodic stationarity state of an uncertain dynamic system can efficiently be optimized in a linear approximation if automatic backward differentiation techniques are employed to evaluate the sensitivities of the constraint functions with respect to a possibly large number of unknown parameters. The benefit of the technique has been illustrated by applying it to a periodically operated biochemical fermentation process.

11.2 Future Research Directions

The algorithms and implementation of the robust optimization and control techniques in this thesis are likely to have much potential in real-world applications. In this thesis, we have only discussed some representative applications for illustrating and testing the methods. However, given the fact that uncertainty plays a role in almost all engineering processes there are many interesting applications of the methods waiting to be discovered.

Besides the applications, the techniques in this thesis also leave space for theoretical developments. Especially a transfer of the open-loop robust optimization to feedback controlled dynamic systems appears to be a natural step for future research. In addition, there are still open questions concerning a priori bounds on the level of conservatism of the presented approximation strategies for robust optimal control. In this thesis, we have shown that the level of conservatism remains small in many applications. A more consistent analysis could be based on an application of the ellipsoidal inner approximation techniques form Chapter 2 to the computation of robust positive invariant tubes for uncertain nonlinear dynamic systems.

Finally, the `ACADO Toolkit` has a great potential to be extended and applied in many fields of engineering. Especially, the code export features show a promising performance and many fast real-world processes, as for example arising in the field of mechatronics, are natural candidates for an application of the developed tools.

Bibliography

[1] P. Agarwal, G. Koshy, and M. Ramirez. An Algorithm for Operating a Fed-batch Fermentor at Optimum Specific Growth Rate. *Biotechnol. Bioeng.*, 33:115–125, 1989.

[2] J. Albersmeyer and H.G. Bock. Sensitivity Generation in an Adaptive BDF-Method. In H.G. Bock, E. Kostina, X.H. Phu, and R. Rannacher, editors, *Modeling, Simulation and Optimization of Complex Processes: Proceedings of the International Conference on High Performance Scientific Computing, March 6-10, 2006, Hanoi, Vietnam*, pages 15–24. Springer, 2008.

[3] J. Albersmeyer and M. Diehl. The Lifted Newton Method and its Application in Optimization. *SIAM Journal on Optimization*, 20(3):1655–1684, 2010.

[4] F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, and S.J. Wright. Nonlinear Predictive Control and Moving Horizon Estimation – An Introductory Overview. In P. M. Frank, editor, *Advances in Control, Highlights of ECC'99*, pages 391–449. Springer, 1999.

[5] F. Allgöwer and A. Zheng. *Nonlinear Predictive Control*, volume 26 of *Progress in Systems Theory*. Birkhäuser, Basel Boston Berlin, 2000.

[6] H. Amann. *Gewöhnliche Differentialgleichungen*. de Gruyter, Berlin; New York, 1983.

[7] M. Anitescu. Nonlinear Programs with Unbounded Lagrange Multiplier Sets. Technical report, Preprint ANL/MCS-P796-0200, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 2000.

[8] M. Anitescu. Global convergence of an elastic mode approach for a class of mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 16:120–145, 2005.

[9] P. Apkarian and D. Noll. Nonsmooth Optimization for Multiband Frequency Domain Control Design. *Automatica*, 43(4):724–731, 2007.

[10] A. Arinstein and M. Gitterman. Inverted spring pendulum driven by a periodic force: linear versus nonlinear analysis. *European Journal of Physics*, 29:385–392, 2008.

[11] U.M. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential–Algebraic Equations*. SIAM, Philadelphia, 1998.

[12] J.P. Aubin. *Viability Theory*. Birkhäuser Boston, 1991.

[13] J.F. Bard. *Practical Bilevel Optimization: Algorithms and Applications.* Kluwer Academic Publishers, Boston MA, 1999.

[14] A.C. Bartlett, C.V. Hollot, and H. Lin. Root Location of an Entire Polytope of Polynomials: It Suffices to Check the Edges. *Mathematics of Control, Signals and Systems*, 1:61–71, 1988.

[15] I. Bauer. *Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik*. PhD thesis, Universität Heidelberg, 1999.

[16] A. Ben-Tal, S. Boyd, and A. Nemirovski. Extending Scope of Robust Optimization: Comprehensive Robust Counterparts of Uncertain Problems. 2005.

[17] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[18] A. Ben-Tal and A. Nemirovski. Robust Truss Topology Design via Semidefinite Programming. *SIAM Journal on Optimization*, 7:991–1016, 1997.

[19] A. Ben-Tal and A. Nemirovski. Robust Convex Optimization. *Math. Oper. Res.*, 23:769–805, 1998.

[20] A. Ben-Tal and A. Nemirovski. Robust Solutions of Uncertain Linear Programs. *Operations Research*, 25:1–13, 1999.

[21] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS-SIAM Series on Optimization. MPS-SIAM, Philadelphia, 2001.

[22] A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

[23] H.Y. Benson, A. Sen, D.F. Shanno, and R.J. Vanderbei. Interior-Point Algorithms, Penalty Methods and Equilibrium Problems. *Computational Optimization and Applications*, 34:155–182, 2006.

[24] D.P. Bertsekas and I.B. Rhodes. On the minimax reachability of target sets and target tubes. *Automatica*, 7:233–247, 1971.

[25] D.P. Bertsekas and I.B. Rhodes. Recursive state estimation for a set-membership description of uncertainty. 16:117–128, 1971.

[26] J.T. Betts. *Practical Methods for Optimal control and Estimation Using nonlinear Programming*. SIAM, 2nd edition, 2010.

[27] B. Bhattacharjee, P. Lemonidis, W.H. Green, and P.I. Barton. Global solution of semi-infinite programs. *Mathmatical Programming (Series B)*, 103(2):283–307, 2005.

[28] S.P. Bhattacharyya, H. Chappelat, and L.H. Keel. Robust Control: The Parametric Approach. *Prentice Hall, Englewood Cliffs, N.J.*, 1995.

[29] Lorenz T. Biegler. *Nonlinear Programming*. MOS-SIAM Series on Optimization. SIAM, 2010.

[30] L.T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering*, 8:243–248, 1984.

[31] L.T. Biegler. An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing*, 46:1043–1053, 2007.

[32] L.T. Biegler and J.B Rawlings. Optimization approaches to nonlinear model predictive control. In W.H. Ray and Y. Arkun, editors, *Proc. 4th International Conference on Chemical Process Control - CPC IV*, pages 543–571. AIChE, CACHE, 1991.

[33] C.H. Bischof, A. Carle, G. Corliss, A. Griewank, and P. Hovland. ADIFOR Generating derivative codes from Fortran programs. *Scientific Programming*, 1:11–29, 1992.

[34] S. Bittanti, P. Colaneri, and G. De Nicolao. The periodic Riccati Equation. In Willems Bittanti, Laub, editor, *The Riccati Equation*. Springer Verlag, 1991.

[35] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

[36] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Birkhäuser, 2008.

[37] H.G. Bock. Numerical Solution of Nonlinear Multipoint Boundary Value Problems with Applications to Optimal Control. *Zeitschrift für Angewandte Mathematik und Mechanik*, 58:407, 1978.

[38] H.G. Bock. Recent advances in parameter identification techniques for ODE. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*. Birkhäuser, Boston, 1983.

[39] H.G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften*. Universität Bonn, Bonn, 1987.

[40] H.G. Bock, I. Bauer, D.B. Leineweber, and J.P. Schlöder. Direct Multiple Shooting Methods for Control and Optimization of DAE in Chemical Engineering. In F. Keil, W. Mackens, H. Voß, and J. Werther, editors, *Scientific Computing in Chemical Engineering II*, volume 2, pages 2–18, Berlin, 1999. Springer.

[41] H.G. Bock, M. Diehl, D.B. Leineweber, and J.P. Schlöder. A direct multiple shooting method for real-time optimization of nonlinear DAE processes. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control*, volume 26 of *Progress in Systems Theory*, pages 246–267, Basel Boston Berlin, 2000. Birkhäuser.

[42] H.G. Bock, E. Eich, and J.P. Schlöder. Numerical Solution of Constrained Least Squares Boundary Value Problems in Differential-Algebraic Equations. In K. Strehmel, editor, *Numerical Treatment of Differential Equations*. Teubner, Leipzig, 1988.

[43] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC World Congress Budapest*, pages 243–247. Pergamon Press, 1984.

[44] P.T. Boggs and J.W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, pages 1–51, 1995.

[45] F.F. Bonsall. *Lectures on some fixed point theorems of functional analysis*. Notes by K.D. Vedak, Tata Institute of Fundamental Research, Bombay, 1962.

[46] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

[47] M.L. Brockman and M. Corless. Quadratic boundedness of nominally linear systems. *International Journal of Control*, 71(6):1105–1117, 1998.

[48] C.G. Broyden. The convergence of a class of double rank minimization algorithms, part I and II. *J. Inst. Maths. Applns.*, 6:76–90 and 222–231, 1970.

[49] A.E. Bryson and Y. Ho. *Applied optimal control: optimization, estimation, and control*. Blaisdell, Waltham, MA, 1969.

[50] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Wiley, New York, 1975.

[51] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton. HIFOO - A MATLAB package for fixed-order controller design and H-infinity Optimization. In *Proceedings of ROCOND 2006*, Toulouse, France, 2006.

[52] J.V. Burke, D. Henrion, A.S. Lewis, and M.L. Overton. Stabilization via Nonsmooth, Nonconvex Optimization. *IEEE Transactions on Automatic Control*, 51(11):1760–1769, 2006.

[53] J.V. Burke, A.S. Lewis, and M.L. Overton. Optimization and Pseudospectra, with Applications to Robust Stability. *SIAM J. Matrix Anal. Appl.*, 25(1):pp. 80–104, 2003.

[54] C. Büskens and H. Maurer. SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. *Journal of Computational and Applied Mathematics*, 120:85–108, 2000.

[55] A. Charnes, W.W. Cooper, and G.H. Symonds. Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958.

[56] B. Chen and T. Harker. A non-interior-point continuation method for linear complementarity problems. *SIAM Journal on Matrix Analysis*, pages 1168–1190, 1993.

[57] H. Chen. *Stability and Robustness Considerations in Nonlinear Model Predictive Control*. Fortschr.-Ber. VDI Reihe 8 Nr. 674. VDI Verlag, Düsseldorf, 1997.

[58] L. Chen and D. Goldfarb. An active set method for mathematical programs with linear complementarity constraint. *SIAM Journal on Optimization*, 2007. (submitted).

[59] P. Colaneri. Continous-time periodic systems in H_2 and H_∞, Part I: Theoretical aspects. *Kybernetika*, vol. 36, no. 2:pp. 211–242, 2000.

[60] A.R. Conn, N. Gould, and P.L. Toint. *Trust-Region Methods*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, USA, 2000.

[61] C.F. Curtiss and J.O. Hirschfelder. Integration of stiff equations. *Proc. Nat. Acad. Sci*, 38:235–243, 1952.

[62] J.E. Cuthrell and L.T. Biegler. Simultaneous optimization and solution methods for batch reactor profiles. *Computers and Chemical Engineering*, 13(1/2):49–62, 1989.

[63] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

[64] T.A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[65] K. Deimling. *Multivalued Differential Equations*. Walter de Gruyter & Co, D-1000 Berlin, 1992.

[66] J.E. Dennis and J. J. Moré. A characterisation of superlinear convergence and its application to quasi–Newton methods. *Mathemathics of Computation*, 28:549–560, 1974.

[67] J.E. Dennis and J. J. Moré. Quasi-Newton Methods, Motivation and Theory. *SIAM Review*, 19(1):46–89, January 1977.

[68] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, Universität Heidelberg, 2001. http://www.ub.uni-heidelberg.de/archiv/1659/.

[69] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*, volume 920 of *Fortschr.-Ber. VDI Reihe 8, Meß-, Steuerungs- und Regelungstechnik*. VDI Verlag, Düsseldorf, 2002. Download also at: http://www.ub.uni-heidelberg.de/archiv/1659/.

[70] M. Diehl and J. Björnberg. Robust Dynamic Programming for Min-Max Model Predictive Control of Constrained Uncertain Systems. *IEEE Transactions on Automatic Control*, 49(12):2253–2257, December 2004.

[71] M. Diehl, H.G. Bock, and E. Kostina. An approximation technique for robust nonlinear optimization. *Mathematical Programming*, 107:213–230, 2006.

[72] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.

[73] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *J. Proc. Contr.*, 12(4):577–585, 2002.

[74] M. Diehl, H. J. Ferreau, and N. Haverbeke. *Nonlinear model predictive control*, volume 384 of *Lecture Notes in Control and Information Sciences*, chapter Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, pages 391–417. Springer, 2009.

[75] M. Diehl, R. Findeisen, and F. Allgöwer. A Stabilizing Real-time Implementation of Nonlinear Model Predictive Control. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-Time and Online PDE-Constrained Optimization*, pages 23–52. SIAM, 2007.

[76] M. Diehl, R. Findeisen, F. Allgöwer, H.G. Bock, and J.P. Schlöder. Nominal Stability of the Real-Time Iteration Scheme for Nonlinear Model Predictive Control. *IEE Proc.-Control Theory Appl.*, 152(3):296–308, 2005.

[77] M. Diehl, F. Jarre, and C. Vogelbusch. Loss of superlinear convergence for an SQP-type method with conic constraints. *SIAM Journal on Optimization*, 16(4):1201–1210, 2006.

[78] M. Diehl, K. Mombaur, and D. Noll. Stability optimization of hybrid periodic systems via a smooth criterion. *IEEE Transactions on Automatic Control*, 54(8):1875–1880, 2009.

[79] M. Diehl, I. Uslu, R. Findeisen, S. Schwarzkopf, F. Allgöwer, H.G. Bock, T. Bürner, E.D. Gilles, A. Kienle, J.P. Schlöder, and E. Stein. Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation

Column. In M. Grötschel, S. O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*, pages 363–384. Springer, 2001. download at: http://www.zib.de/dfg-echtzeit/Publikationen/Preprints/Preprint-01-16.html.

[80] M. Diehl, A. Walther, H. G. Bock, and E. Kostina. An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization. *Optimization Methods and Software*, 25:531–552, 2010.

[81] J.C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23, 1978.

[82] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-Space Solutions to Standard H_2 and H_∞ Control Problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[83] G.E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Springer, New York, 1999.

[84] H.G. Eggleston. *Convexity*. Camebridge University Press, 1969.

[85] L. El-Ghaoui and H. Lebret. Robust Solutions to Least-Square Problems to Uncertain Data Matrices. *SIAM Journal on Matrix Analysis*, 18:1035–1064, 1997.

[86] L.C. Evans and P.E. Souganidis. Differential Games and Representation Formulas for Solutions of Hamilton-Jacobi-Isaacs equations. *Indiana University Mathematics Journals*, 33(5):773–797, 1984.

[87] B.C. Fabien. dsoa: The implementation of a dynamic system optimization algorithm. *Optimal Control Applications and Methods*, 31:231–247, 2010.

[88] F. Facchinei, H. Jiang, and L. Qi. A smoothing method for mathematical programs with equilibrium constraints. *Mathematical Programming*, 85:107–134, 1999.

[89] H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[90] H.J. Ferreau, B. Houska, K. Geebelen, and M. Diehl. Real-time control of a kite-carousel using an auto-generated nonlinear MPC algorithm. In *Proceedings of the IFAC World Congress*, 2011. (accepted).

[91] H.J. Ferreau, B. Houska, T. Kraus, and M. Diehl. Numerical Methods for Embedded Optimisation and their Implementation within the ACADO Toolkit. In W. Mitkowski R. Tadeusiewicz, A. Ligeza and M. Szymkat, editors, *7th Conference - Computer Methods and Systems (CMS'09)*, Krakow, Poland, November 2009. Oprogramowanie Naukowo-Techniczne.

[92] A. Fischer. A special Newton-type optimization method. *Optimization*, 24:269–284, 1992.

[93] M.L. Flegel and C. Kanzow. A Fritz John approach to first order optimality conditions for mathematical programs with equilibrium constraints. *Optimization*, 52:277–286, 2003.

[94] M.L. Flegel and C. Kanzow. On the Guignard constraint qualification for mathematical programs with equilibrium constraints. *Optimization*, 54:517–537, 2005.

[95] R. Fletcher. A new approach to variable metric algorithms. *Computer J.*, 13:317–322, 1970.

[96] R. Fletcher. *Practical Methods of Optimization*. Wiley, Chichester, 2nd edition, 1987.

[97] R. Fletcher, N.I.M. Gould, S. Leyffer, P.L. Toint, and A. Wächter. Global Convergence of a Trust-Region SQP-Filter Algorithm for General Nonlinear Programming. *SIAM Journal on Optimization*, 13(3):635–659, 2002.

[98] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local Convergence of SQP Methods for Mathematical Programs with Equilibrium Constraints. *SIAM Journal on Optimization*, 17:259–286, 2006.

[99] C.A. Floudas. *Deterministic Global Optimization: Theory, Methods, and Applications*. Kluwer Academic Publishers, 1999.

[100] C.A. Floudas and O. Stein. The Adaptative Convexification Algorithm: a Feasible Point Method for Semi-Infinite Programming. *SIAM Journal on Optimization*, 18(4):1187–1208, 2007.

[101] A.L. Fradkov and V.A. Yakubovich. The S-procedure and duality realations in nonconvex problems of quadratic programming. *Vestnik Leningrad Univ. Math.*, 5:101–109, 1973.

[102] J.D. Glover and F.C. Schweppe. Control of Linear Dynamic Systems with Set Constrainted Disturbances. *IEEE Transactions on Automatic Control*, 16:411–423, 1971.

[103] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for Maximum Cut and satisfiability problems using semidefinite programming. *Journal of ACM*, 42:1115–1145, 1995.

[104] D. Goldfarb. A family of variable metric methods derived by variational means. *Maths. Comp.*, 17:739–764, 1970.

[105] P. J. Goulart, E.C. Kerrigan, and J.M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42:523–533, 2006.

[106] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control. *Lecture Notes in Control and Information Sciences, Springer*, pages 95–110, 2008.

[107] M. Grant and S. Boyd. CVX webage. http://cvxr.com/cvx, 2011.

[108] A. Griewank. On Automatic Differentiation. In *Mathematical Programming: Recent Developments and Applications*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1989.

[109] A. Griewank. *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, 2000.

[110] A. Griewank, D. Juedes, H. Mitev, J. Utke, O. Vogel, and A. Walther. ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++. Technical report, Technical University of Dresden, Institute of Scientific Computing and Institute of Geometry, 1999. Updated version of the paper published in *ACM Trans. Math. Software* 22, 1996, 131–167.

[111] A. Griewank and Ph.L. Toint. Partitioned variable metric updates for large structured optimization problems. *Numerische Mathematik*, 39:119–137, 1982.

[112] A. Griewank and A. Walther. *Evaluating Derivatives*. SIAM, 2008.

[113] M. Guignard. Generalized Kuhn-Tucker conditions for mathematical programming problems in a Banach space,. *SIAM Journal on Control*, 7:232–241, 1969.

[114] E. Hairer and G. Wanner. RADAU5 - an implicit Runge-Kutta code. Report, Université de Genève, Dept. de mathématiques, Genève, 1988.

[115] S. P. Han. A Globally Convergent Method for Nonlinear Programming. *JOTA*, 22:297–310, 1977.

[116] D. Henrion and J.B. Lasserre. GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi. *ACM Transactions on Mathematical Software*, 29(2):165–194, 2003.

[117] D. Henrion, S. Tarbouriech, and D. Arzelier. LMI Approximations for the Radius of the Intersection of Ellipsoids: A Survey. *Journal of Optimization Theory and Applications*, 108(1):1–28, 2001.

[118] R. Hettich and H.T. Jongen. *Semi-infinite programming: Conditions of optimality and applications*. Optimization Techniques, Part 2, Lecture Notes in Control and Inform. Sci. 7, J. Stoer, Springer, 1978.

[119] R. Hettich and K. Kortanek. *Semi infinite programming:Theory, Methods, and Application*, volume 35. SIAM Review, 1993.

[120] R. Hettich and G. Still. Second order optimality conditions for generalized semi-infinite programming problems. *Optimization*, 34:195–211, 1995.

[121] H. Hinsberger. *Ein direktes Mehrzielverfahren zur Lösung von Optimalsteuerungsproblemen mit grossen, differential–algebraischen Gleichungssystemen und Anwendungen aus der Verfahrenstechnik*. PhD thesis, Technische Universität Clausthal, 1998.

[122] H. Hinsberger, S. Miesbach, and H.J. Pesch. Optimal temperature control of semibatch polymerization reactors. In F. Keil, W. Mackens, H. Voss, and J. Werther, editors, *Scientific Computing in Chemical Engineering*, Heidelberg, 1996. Springer.

[123] B. Houska. Robustness and Stability Optimization of Open-Loop Controlled Power Generating Kites. Master's thesis, University of Heidelberg, 2007.

[124] B. Houska and M. Diehl. Robust nonlinear optimal control of dynamic systems with affine uncertainties. In *Proceedings of the 48th Conference on Decision and Control*, pages 2274–2279, Shanghai, China, 2009.

[125] B. Houska and M. Diehl. Nonlinear Robust Optimization of Uncertainty Affine Dynamic Systems under the L-infinity Norm. In *In Proceedings of the IEEE Multi - Conference on Systems and Control*, pages 1091–1096, Yokohama, Japan, 2010.

[126] B. Houska and M. Diehl. Robustness and Stability Optimization of Power Generating Kite Systems in a Periodic Pumping Mode. In *Proceedings of the IEEE Multi - Conference on Systems and Control*, pages 2172–2177, Yokohama, Japan, 2010.

[127] B. Houska and M. Diehl. Nonlinear Robust Optimization via Sequential Convex Bilevel Programming. *Mathmatical Programming*, 2011. (submitted).

[128] B. Houska and M. Diehl. A quadratically convergent inexact SQP method for optimal control of differential algebraic equations. *Optimal Control Applications & Methods, John Wiley & Sons*, pages 1–23, 2011. (submitted).

[129] B. Houska and M. Diehl. Robust design of linear control laws for constrained nonlinear dynamic systems. In *Proc. of the 18th IFAC World Congress*, Milan, Italy, September 2011.

[130] B. Houska and H.J. Ferreau. ACADO Toolkit User's Manual. http://www.acadotoolkit.org, 2009–2011.

[131] B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[132] B. Houska, H.J. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 2011. (accepted).

[133] B. Houska, F. Logist, J. Van Impe, and M. Diehl. Approximate robust optimization of time-periodic stationary states with application to biochemical processes. In *Proceedings of the 48th Conference on Decision and Control*, pages 6280–6285, Shanghai, China, 2009.

[134] R. Isaacs. Differential Games. John Wiley and Sons, 1965.

[135] H.T. Jongen, J.J. Rückmann, and O. Stein. Generalized semi-infinite optimization: A first order optimality condition and examples. *Mathematical Programming*, pages 145–158, 1998.

[136] P.T. Kabamba, S.M. Meerkov, and E.K. Poh. Stability robustness in closed loop vibrational control. *International Journal of Robust and Nonlinear Control*, 8:1101–1111, 1998.

[137] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.

[138] R.E. Kalman. Lyapunov functions for the problem of Lur'e in automatic control. *Proc. Nat. Acad. Sci. USA*, 49:pp. 201–205, 1963.

[139] E.C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. PhD thesis, University of Cambridge, UK, 2000.

[140] E.C. Kerrigan and J.M. Maciejowski. Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution. *International Journal on Robust and Nonlinear Control*, 14:395–413, 2004.

[141] K.-U. Klatt and S. Engell. Rührkesselreaktor mit Parallel- und Folgereaktion. In S. Engell, editor, *Nichtlineare Regelung – Methoden, Werkzeuge, Anwendungen. VDI-Berichte Nr. 1026*, pages 101–108. VDI-Verlag, Düsseldorf, 1993.

[142] I. Kolmanovsky and E.G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Math. Probl. Eng.*, 4(4):317–367, 1998.

[143] D. Kraft. On converting optimal control problems into nonlinear programming problems. In K. Schittkowski, editor, *Computational Mathematical Programming*, volume F15 of *NATO ASI*, pages 261–280. Springer, 1985.

[144] A.A. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *Communications in Information and Systems*, 6:179–192, 2006.

[145] A.B. Kurzhanski and P. Valyi. Ellipsoidal Calculus for Estimation and Control. Birkhäuser Boston, 1997.

[146] A.B. Kurzhanski and P. Varaiya. Reachability analysis for uncertain systems - the ellipsoidal technique. *Dynamics of Continuous, Discrete and Impulsive Systems, Ser. B*, 9:347–367, 2002.

[147] W. Langson, S.V. Rakovic I. Chryssochoos, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[148] J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optimization*, 11(3):796–817, 2001.

[149] J.B. Lasserre. *Moments, Positve Polynomials and Their Applications*. Imperial College Press, 2009.

[150] D.B. Leineweber. *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, volume 613 of *Fortschritt-Berichte VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 1999.

[151] D.B. Leineweber, I. Bauer, H.G. Bock, and J.P. Schlöder. An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization. Part I: Theoretical Aspects. *Computers and Chemical Engineering*, 27:157–166, 2003.

[152] D.B. Leineweber, A.A.S. Schäfer, H.G. Bock, and J.P. Schlöder. An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization. Part II: Software Aspects and Applications. *Computers and Chemical Engineering*, 27:167–174, 2003.

[153] Y. Lin, E.D. Sontag, and Y. Wang. A smooth converse lyapunov theorem for robust stability. *SIAM J. Control Optim.*, 34(1):1–33, 1996.

[154] G.S. Liu and J.Z. Zhang. A new branch and bound algorithm for solving quadratic programs with linear complementarity constraints. *Journal on Computational and Applied Mathematics*, 146:77–87, 2002.

[155] J. Llibre. Open problems on the algebraic limit cycles of planar polynomial vector fields. *Bul. Acad. Ştiinţe Repub. Mold. Mat.*, 1:19–26, 2008.

[156] J. Löfberg. Approximations of closed-loop MPC. In Proceedings of the 42nd IEEE Conference on Decision and Control, pp 1438–1442, 2003.

[157] F. Logist, B. Houska, M. Diehl, and J. Van Impe. Fast Pareto set generation for nonlinear optimal control problems with multiple objectives. *Structural and Multidisciplinary Optimization*, 42(4):591–603, 2010.

[158] F. Logist, B. Houska, M. Diehl, and J. Van Impe. A toolkit for multi-objective optimal control in bioprocess engineering. Proceedings of the 11th symposium on Computer Applications in Biotechnology, pp. 269–274, Leuven, Belgium, 2010.

[159] F. Logist, B. Houska, M. Diehl, and J. Van Impe. Robust multi-objective optimal control of uncertain biochemical processes. *Chemical Engineering Science*, 2011. (accepted).

[160] F. Logist, B. Houska, M. Diehl, and J. Van Impe. Robust optimal control of a biochemical reactor with multiple objectives. European Symposium on Computer Aided Process Engineering (ESCAPE). Chalkidiki, Greece, May 2011.

[161] F. Logist, I. Smets, and J. Van Impe. Derivation of generic optimal reference temperature profiles for steady-state exothermic jacketed tubular reactors. *Journal of Process Control*, 18:92–104, 2008.

[162] Z. Luo, J. S. Pang, and D. Ralph. Piecewise sequential quadratic programming for mathematical programs with nonlinear complementarity constraints. *A. Migdalas, P. M. Pardalos and P. Värbrand, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands*, pages 209–229, 1998.

[163] M.A. Lyapunov. Probléme general de la stabilité du mouvement. *Ann. Fac. Sci. Toulouse Math.*, 5(9):pp. 203–474, 1907.

[164] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachibility specifications for hybrid systems. *Automatica*, 35:pp. 349–370, 1999.

[165] C.M. Maes. *A Regularized Acitve-Set Method for Sparse Convex Quadratic Programming*. PhD thesis, Stanford University, 2010.

[166] J. Mattingley and S. Boyd. *Convex Optimization in Signal Processing and Communications*, chapter Automatic Code Generation for Real-Time Convex Optimization. Cambridge University Press, 2009.

[167] D.Q. Mayne, S.V. Rakovic, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42:1217–1222, 2006.

[168] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publisher, Boston, 1999.

[169] L.B. Miller and H. Wagner. Chance-constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965.

[170] I.M. Mitchell, A.M. Bayen, and C.J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[171] K. Mombaur. *Stability Optimization of Open-Loop Controlled Walking Robots*. PhD thesis, Universität Heidelberg, 2001.

[172] D. Mustafa and K. Glover. *Minimum Entropy H_∞ Control*. Lecture Notes in Control and Information Sciences. Springer, 1990.

[173] Z.K. Nagy and R.D. Braatz. Robust nonlinear model predictive control of batch processes. *AIChE Journal*, 49(7):1776–1786, 2003.

[174] Z.K. Nagy and R.D. Braatz. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *Journal of Process Control*, 14:411–422, 2004.

[175] Z.K. Nagy and R.D. Braatz. Distributional uncertainty analysis using power series and polynomial chaos expansions. *Journal of Process Control*, 17:229–240, 2007.

[176] A. Nemirovski, C. Roos, and T. Terlaky. On maximization of quadratic form over intersection of ellipsoids with common center. *Mathmatical Programming*, 86(3):463–473, 1999.

[177] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[178] A. Nemirovski and A. Shapiro. Scenario approximations of chance constraints. In G. Calafiore and F. Dabbene, eds. Probabilistic and Randomized Methods for Design under Uncertainty, Springer, 2006.

[179] Y. Nesterov. Semidefinite relaxation and non-convex quadratic optimization. *Optimization Methods and Software*, 12:1–20, 1997.

[180] Y. Nesterov and B.T. Polyak. Cubic regularization of Newton's method and its global performance. *Mathmatical Programming*, 112(1):177–205, 2006.

[181] A. Neumaier. *Complete Search in Continuous Global Optimization and Constraint Satisfaction*, pages 271–369. Cambridge University Press, 2004.

[182] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

[183] T. Ohtsuka. A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control. *Automatica*, 40(4):563–574, 2004.

[184] P.A. Parrilo. Polynomial games and sum of squares optimization. In *Proceedings of the 45th IEEE Conference on Decision & Control*, pages 2855–2860, San Diego, CA, USA, December 2006.

[185] S. J. Parulekar. Analysis of forced periodic operations of continuous bioprocesses - single input variations. *Chemical Engineering Science*, 53(14):2481–2502, 1998.

[186] L.R. Petzold. DASSL. http://pitagora.dm.uniba.it/ testset/solvers/dassl.php, June 1991.

[187] K.J. Plitt. Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen. Master's thesis, Universität Bonn, 1981.

[188] I. Polik and T. Terkaly. A Survey of the S-Lemma. *SIAM Review*, 49(3):371–418, 2007.

[189] L.S. Pontryagin, V.G. Boltyanski, R.V. Gamkrelidze, and E.F. Miscenko. *The Mathematical Theory of Optimal Processes*. Wiley, Chichester, 1962.

[190] M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G.A. Watson, editor, *Numerical Analysis, Dundee 1977*, volume 630 of *Lecture Notes in Mathematics*, Berlin, 1978. Springer.

[191] M.J.D. Powell. *The convergence of variable metric methods for nonlinearly constrained optimization calculations*. Academic Press, New York, 1978.

[192] M.J.D. Powell. *The performance of two subroutines for constrained optimization on some difficult test problems*, pages 160–177. In Paul T. Boggs, Richard H. Byrd, Robert B. Schnabel (eds), Numerical Optimization, SIAM, 1984.

[193] A. Prèkopa. On probabilistic constrained programming. Proceedings of the Princeton Symposium on Mathematical Programming, Princeton University Press, pp. 113–138, 1970.

[194] S.V. Rakovic and M. Fiacchini. Approximate reachability analysis for linear discrete time systems using homothety and invariance. In *Proceedings of the 17th World Congress on Automatic Control, July 6-11*, Seoul, Korea, 2008.

[195] S.V. Rakovic and K.I. Kouramas. The Minimal Robust Positively Invariant Set for Linear Discrete Time Systems: Approximation Methods and Control Applications. In *Proceedings of the 45th Conference on Decision and Control*, 2006.

[196] D. Ralph. Sequential quadratic programming for mathematical programs with linear complementarity constraints. *Computational Techniques and Applications*, 1996.

[197] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, 2009.

[198] W.C. Rheinboldt. Differential-algebraic systems as differential equations on manifolds. *Math. of Comput.*, 43:473–482, 1984.

[199] S. M. Robinson. First Order Conditions for General Nonlinear Optimization. *SIAM Journal on Applied Mathematics, Vol. 30, No. 4 (Jun., 1976), pp. 597-607*, 30:597–607, 1976.

[200] S. M. Robinson. Strongly Regular Generalized Equations. *Mathematics of Operations Research, Vol. 5, No. 1 (Feb., 1980), pp. 43-62*, 5:43–62, 1980.

[201] A. Romanenko, N. Pedrosa, J. Leal, and L. Santos. *Seminario de Aplicaciones Industriales de Control Avanzado*, chapter A Linux Based Nonlinear Model Predictive Control Framework, pages 229–236. 2007.

[202] L. Ruan and X.D. Chen. Comparison of Several Periodic Operations of a Continous Fermentation Process. *Biotechnol. Prog.*, 12:286–288, 1996.

[203] J.J. Rückmann and O. Stein. On linear and linearized generalized semi-infinite optimization problems. *Ann. Oper. Res.*, pages 191–208, 2001.

[204] R.W.H. Sargent and G.R. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Proceedings of the 8th IFIP Conference on Optimization Techniques (1977), Part 2*, Heidelberg, 1978. Springer.

[205] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. *Math. Oper. Res.*, 25:1–22, 2000.

[206] C.W. Scherer. Special issue on: Linear matrix inequalities in control. *European Journal of Control*, 12:3–29, 2006.

[207] J.P. Schlöder. *Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung*, volume 187 of *Bonner Mathematische Schriften*. Universität Bonn, Bonn, 1988.

[208] V.H. Schulz. *Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots*. PhD thesis, Universität Heidelberg, 1996.

[209] F.C. Schweppe. *Uncertain Dynamic Systems*. Prentice Hall, 1973.

[210] H. Seguchi and T. Ohtsuka. Nonlinear Receding Horizon Control of an Underactuated Hovercraft. *International Journal of Robust and Nonlinear Control*, 13(3–4):381–398, 2003.

[211] D.F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Maths. Comp.*, 24:647–656, 1970.

[212] H.D. Sherali and C.H Tuncbilek. A Reformulation-Convexification Approach for Solving Nonconvex Quadratic Programming Problems. *Journal of Global Optimization*, 7:1–31, 1995.

[213] H.D. Sherali and C.H. Tuncbilek. New Reformulation Linearization/Convexification Relaxations for Univariate and Multivariate Polynomial Programming Problems. *Operations Research Letters*, 21:1–9, 1997.

[214] K. Shimizu and M. Lu. A Global Optimization Method for the Stackelberg Problem with Convex Functions via Problem Transformation and Concave Programming. *IEEE Transactions on Systems, Man., and Cybernetics*, 25(23):1635–1640, December 1995.

[215] N.Z. Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

[216] L.L. Simon, Z.K. Nagy, and K. Hungerbuehler. *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, chapter Swelling Constrained Control of an Industrial Batch Reactor Using a Dedicated NMPC Environment: OptCon, pages 531–539. Springer, 2009.

[217] S. Skogestad and I. Postlethwaite. *Multivariable feedback control - Analysis and design*. Wiley, 2nd edition, 2005.

[218] L. Sonneborn and F. Van Vleck. The Bang-Bang Principle for Linear Control Systems. *SIAM Journal on Control*, 2:151–159, 1965.

[219] A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, pages 1154–1157, 1973.

[220] O. Stein. The feasible set in generalized semi-infinite programming. In M. Lassonde, editor, *Approximation, Optimization, and Mathematical Economics*, pages 313–331, Heidelberg, 2001. Physica-Verlag.

[221] O. Stein. *Bilevel Strategies in Semi Infinite Optimization*. Kluwer, Boston, 2003.

[222] O. Stein and G. Still. On Optimality Conditions for Generalized Semi-Infinite Programming Problems. *Journal of Optimization Theory and Applications*, 104(2):443–458, 2000.

[223] O. Stein and G. Still. Solving Semi-infinite Optimization Problems with Interior Point Techniques. *SIAM Journal on Control and Optimization*, 42(3):769–788, 2003.

[224] A. Stephenson. On a new type of dynamical stability. *Mem. Proc. Manch. Lit. Phil. Soc.*, 52:1–10, 1908.

[225] G. Still. Discretization in semi-infinite programming: The rate of convergence. *Mathematical Programming*, 91:53–69, 2001.

[226] G. Still. Generalized semi-infinite programming: Numerical aspects. *Optimization*, 49:223–242, 2001.

[227] P. Toint. On sparse and symmetric matrix updating subject to a linear equation. *Mathemathics of Computation*, 31(140):954–961, 1977.

[228] N. Trefethen and M. Embree. Spectra and Pseudospectra - The behavior of nonormal matrices. Princeton University Press, 2005.

[229] J. Vanbiervliet, B. Vandereycken, W. Michiels, S. Vandewalle, and M. Diehl. The smoothed spectral abscissa for robust stability optimization. *SIAM Journal on Optimization*, 20(1):156–171, 2009.

[230] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[231] T.V. Voorhis. A Global Optimization Algorithm using Lagrangian Underestimates and the Interval Newton Method. *Journal of Global Optimization*, 24:349–370, 2002.

[232] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD thesis, Carnegie Mellon University, 2002.

[233] A. Wächter and L. Biegler. IPOPT - an Interior Point OPTimizer. https://projects.coin-or.org/Ipopt, 2009.

[234] A. Walther and L. Biegler. Numerical experiments with an inexact Jacobian trust-region algorithm. *Computational Optimization and Applications*, DOI 10.1007, 2007.

[235] Y. Wang and S. Boyd. Performance bounds and suboptimal policies for linear stochastic control via LMIs. *International Journal of Robust and Nonlinear Control*, DOI 10.1002, 2010. (accepted)

[236] G.W. Weber. Generalized Semi-Infinite Optimization and Related Topics. Habilitation Thesis, Darmstadt University of Technology, Darmstadt, Germany, 1999.

[237] L. Wirsching. An SQP Algorithm with Inexact Derivatives for a Direct Multiple Shooting Method for Optimal Control Problems. Master's thesis, University of Heidelberg, 2006.

[238] G. Wolf. *Mathieu Functions and Hill's equation*. Cambridge University Press, 2010.

[239] S.J. Wright. Modifying SQP for degenerate problems. *SIAM Journal on Optimization*, 13:470–497, 2002.

[240] V.A. Yakubovich. S-procedure in nonlinear control theory. *Vestnik Leningrad University*, 4:73–93, 1977.

[241] G. Zames. Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26:301–320, 1981.

[242] V. M. Zavala and L.T. Biegler. The Advanced Step NMPC Controller: Optimality, Stability and Robustness. *Automatica*, 45:86–93, 2009.

[243] J. Zhang and G. Liu. A New Extreme Point Algorithm and Its Application in PSQP Algorithms for Solving Mathematical Programs with Linear Complementarity Constraints. *Journal of Global Optimization*, 19(4):345–361, 2001.

[244] K. Zhou, J.C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, Englewood Cliffs, NJ, 1996.

Links and Software Homepages

[245] ACADO Toolkit Homepage. http://www.acadotoolkit.org, 2009–2011.

[246] qpOASES Homepage. http://www.qpOASES.org, 2007–2011-hp.

[247] Sundials - SUite of Nonlinear and DIfferential/ALgebraic equation Solvers (web page and software). https://computation.llnl.gov/casc/sundials, 2009.

[248] Tomlab Optimization. PROPT: Matlab Optimal Control Software (ODE,DAE). http://tomdyn.com, 2009–2011.

Curriculum Vitae

Personalia

- Name: Boris Houska
- Date of birth: 18 December 1982
- Nationality: German

Higher Education

2003–2007: Diploma in mathematics at the university of Heidelberg, Germany.
Thesis: *Robustness and Stability Optimization of Open-Loop Controlled Power Generating Kites*.

2008–2011: PhD student at the K.U. Leuven, Leuven, Belgium.
Thesis: *Robust Optimzation of Dynamic Systems*.

Sep-Dec 2010: Research stay at the mathematics department of the Jiao Tong university, Shanghai, China.

Publications

Articles in internationally reviewed scientific journals

1. B. Houska, H.J. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, (8 pages), 2011. (accepted, in print)

2. B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32, pp:298-312, 2011. DOI: 10.1002/oca.939

3. B. Houska and M. Diehl. Nonlinear Robust Optimization via Sequential Convex Bilevel Programming. *Mathematical Programming*, (36 pages), 2011. (submitted in June 2010)

4. B. Houska and M. Diehl. A quadratically convergent inexact SQP method for optimal control of differential algebraic equations. *Optimal Control Applications & Methods, John Wiley & Sons*, (23 pages), 2011. (The original version was submitted in December 2009. A revised version has been submitted in October 2010 after including reviewer comments. Chapter 9 is based on this improved version).

5. F. Logist, B. Houska, M. Diehl, and J. Van Impe. Fast Pareto set generation for nonlinear optimal control problems with multiple objectives. *Structural and Multidisciplinary Optimization*, 42(4), pp:591-603, 2010.

6. F. Logist, B. Houska, M. Diehl, and J. Van Impe. Robust multi-objective optimal control of uncertain biochemical processes. *Chemical Engineering Science*, (38 pages), 2011. (accepted)

7. A. Ilzhoefer, B. Houska, and M. Diehl. Nonlinear MPC of kites under varying wind conditions for a new class of large scale wind power generators. *International Journal of Robust and Nonlinear Control*, 17(17), pp:1590–1599, 2011.

Articles in book chapters

1. B. Houska, F. Logist, M. Diehl, and J. Van Impe. A Tutorial on Numerical Methods for State and Parameter Estimation in Nonlinear Dynamic Systems. Accepted for publication as a chapter (22 pages) in the Springer book "Identification for

Automotive System" by Daniel Alberer, Håkan Hjalmarsson, and Luigi del Re (editors). The book shall appear in 2011.

Papers at international conferences, published in full in proceedings

1. B. Houska and M. Diehl. Robust design of linear control laws for constrained nonlinear dynamic systems. In *Proc. of the 18th IFAC World Congress*, Milan, Italy, (6 pages) September 2011. (accepted, to appear in September 2011)

2. B. Houska and M. Diehl. Nonlinear Robust Optimization of Uncertainty Affine Dynamic Systems under the L-infinity Norm. In *In Proceedings of the IEEE Multi - Conference on Systems and Control*, Yokohama, Japan, pp:1091–1096, 2010.

3. B. Houska and M. Diehl. Robustness and Stability Optimization of Power Generating Kite Systems in a Periodic Pumping Mode. In *Proceedings of the IEEE Multi - Conference on Systems and Control*, Yokohama, Japan, pp:2172–2177, 2010.

4. B. Houska and M. Diehl. Robust nonlinear optimal control of dynamic systems with affine uncertainties. In *Proceedings of the 48th Conference on Decision and Control*, Shanghai, China, pp:2274–2279, 2009.

5. B. Houska, F. Logist, J. Van Impe, and M. Diehl. Approximate robust optimization of time-periodic stationary states with application to biochemical processes. In *Proceedings of the 48th Conference on Decision and Control*, Shanghai, China, pp:6280–6285, 2009.

6. B. Houska and M. Diehl. Optimal Control for Power Generating Kites. In *Proceedings of the 9th European Control Conference*, Kos, Greece, pp:3560–3567, 2007.

7. B. Houska and M. Diehl. Optimal control of towing kites. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, USA, pp:2693–2697, 2007.

8. H.J. Ferreau, B. Houska, K. Geebelen, and M. Diehl. Real-time control of a kite-model using an auto-generated nonlinear MPC algorithm. In *Proceedings of the IFAC World Congress*, Milano, Italy, 2011. (accepted, to appear in September 2011)

9. H.J. Ferreau, B. Houska, and M. Diehl. Numerical methods for embedded optimisation and their implementation with the ACADO toolkit. In Tadeusiewicz, R. (Ed.), Ligeza, A. (Ed.), Mitkowski, W. (Ed.), Szymkat, M. (Ed.), *Proceedings of the 7th Conference – Computer Methods and Systems (CMS'09)*, Krakow, Poland, pp:13–29, 2009.

10. F. Logist, B. Houska, M. Diehl, and J. Van Impe. Robust optimal control of a biochemical reactor with multiple objectives. In *European Symposium on Computer Aided Process Engineering (ESCAPE).*, Chalkidiki, Greece, 2011. (to appear)

11. F. Logist, B. Houska, M. Diehl, and J. Van Impe. A Toolkit for Multi-Objective Optimal Control in Bioprocess Engineering. In *Proceedings of the 11th symposium on Computer Applications in Biotechnology.*, Leuven (Belgium), pp:269–274, 2010.

12. F. Logist, B. Houska, M. Diehl, and J. Van Impe. A toolkit for efficiently generating Pareto sets in (bio)chemical multi-objective optimal control problems. In *Proceedings of the 20th European Symposium on Computer Aided Process Engineering (ESCAPE-20)*, Ischia, Italy, pp:481–486, 2010.

Arenberg Doctoral School of Science, Engineering & Technology
Faculty Engineering.
Department of Electrical Engineering
Research group SCD
Kasteelpark Arenberg 10 (bus 2446), B-3001 Leuven



