Reachability and robust design in dynamic systems

by

Stuart Maxwell Harwood

B.S., Northwestern University (2009)

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2015

Author .... Signature redacted
Department of Chemical Engineering
26 June 2015

Certified by................................. Signature redacted
Paul I. Barton
Lammot du Pont Professor of Chemical Engineering
Thesis Supervisor

Accepted by .... Signature redacted ....
Richard D. Braatz
Edwin R. Gilliland Professor of Chemical Engineering
Chairman, Committee for Graduate Students

Reachability and robust design in dynamic systems

by

Stuart Maxwell Harwood

Submitted to the Department of Chemical Engineering
on 26 June 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

Systems of engineering interest usually evolve in time. Models that capture this dynamic behavior can more accurately describe the system. Dynamic models are especially important in the chemical, oil and gas, and pharmaceutical industries, where processes are intrinsically dynamic, or taking into account dynamic behavior is critical for safety. Especially where safety is concerned, uncertainty in the inputs to these models must be addressed. The problems of forward reachability and robust design provide information about a dynamic system when uncertainty is present.

This thesis develops theory and numerical methods for approaching the problems of reachability and robust design applied to dynamic systems. The main assumption is that the models of interest are initial value problems (IVPs) in ordinary differential equations (ODEs). In the case of reachability analysis, the focus is on efficiently calculated enclosures or "bounds" of the reachable sets, since one motivating application is to (deterministic) global dynamic optimization, which requires such information. The theoretical approach taken is inspired by the theory of differential inequalities, which leads to methods which require the solution of an auxiliary IVP defined by parametric optimization problems. Major contributions of this work include methods and theory for efficiently estimating and handling these auxiliary problems. Along these lines, a method for constructing affine relaxations with special parametric properties is developed. The methods for calculating bounds also are extended to a method for calculating affine relaxations of the solutions of IVPs in parametric ODEs.

Further, the problem of ODEs with linear programs embedded is analyzed. This formulation has further application to dynamic flux balance models, which can apply to bioreactors. These models have properties that can make them difficult to handle numerically, and this thesis provides the first rigorous analysis of this problem as well as a very efficient numerical method for the solution of dynamic flux balance models.

The approach taken to robust design is inspired by design centering and, more generally, generalized semi-infinite programming. Theoretical results for reformulating generalized semi-infinite programs are proposed and discussed. This discussion leads to a method for robust design that has clear numerical benefits over others when the system of interest is dynamic in nature. One major benefit is that much of the computational effort can be performed by established commercial software for global optimization. Another method which has a simple implementation in the context of branch and bound is also developed.

Thesis Supervisor: Paul I. Barton
Title: Lammot du Pont Professor of Chemical Engineering

Acknowledgments

First of all, I would like to thank my advisor Professor Paul Barton. This thesis represents the support of many, and Paul's intellectual support has been foremost. The depth and breadth of his knowledge always astounds me. His connections in both academia and industry have been essential to my professional development. Further, I am deeply grateful for the freedom he afforded me to pursue the topics and research paths of interest to me.

I would also like to thank my thesis committee, Professors William Green and Richard Braatz. Their different perspectives and incredible experience in their fields have been invaluable in rounding out this thesis. I would also like to acknowledge funding from Novartis Pharmaceuticals as part of the Novartis-MIT Center for Continuous Manufacturing. Their input has also directed this thesis toward useful applications.

I would also like to acknowledge the support of Process Systems Engineering Laboratory members past and present. My day-to-day work has been facilitated and enlivened by their questions, answers, and tangential conversations. In particular, Joseph Scott, Kai Höffner, Kamil Khan, Achim Wechsung, Spencer Schaber, Matt Stuber, Jose Gomez, and Garrett Dowdy provided incredible prior work, debugging, tips on best practices, software tools, or interesting discussions at some point during graduate school.

To my fellow MIT chemical engineering classmates, I am grateful for camaraderie. I think all of us appreciated the collective friendship and intelligence of this group at some point in our graduate school experience. To friends and family both near and far, I am grateful for emotional support. I am grateful for the times I was able to visit aunts and uncles and cousins in Concord and Littleton. They lent me furniture and gave me eating utensils when I first moved to Cambridge that I am still using. I am glad to have friends who either moved to Boston, made trips out here, or with whom I was able to chat, back home or on the phone. Their company and friendship help me relax and attain a healthy work-life balance. In particular, I would like to thank Michael, Matt, and Emily for helping to make my time at MIT memorable.

I thank Jessica and Stephen for being hospitable, loving, and overall being family. Wherever we are, I always enjoy spending time with them. To Rosa, I am thankful for her patience and daily support. Graduate school has had its ups and downs, but I am glad to share it with her.

Finally, to my parents, what can I say? I am who I am because of them. Their patience for my many questions as a child fostered an appetite for discovery, and their love and confidence in me gave me the self-esteem and resolve so badly needed at various stages in graduate school.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The general objective of this thesis is to develop theory and numerical methods for handling uncertainty in dynamic systems. In specific, the problems of forward reachability analysis and robust design for dynamic systems are considered. These two problems aim to provide information about a system when there is uncertainty in various parameters or inputs.

The pharmaceutical industry provides current and interesting applications for these problems. Relevant processes are typically dynamic in nature and the ability to predict the effect that perturbations in process inputs will have is vital to quality by design initiatives [212]. Because of uncertainties in measurements, a rigorous enclosure of all possible responses is the most desirable information (this is related to forward reachability analysis). Further, quality by design typically requires a design space, a set of parameters and input variables that produce a predictable, desirable output (this is related to the robust design problem).

The following section discusses these and related problems in greater detail and reviews existing approaches and results. Then §1.2 provides an overview of the contributions of this thesis.

1.1 Background

1.1.1 Forward reachability

The problem of forward reachability analysis as considered here applies to the initial value problem (IVP) in ordinary differential equations (ODEs) subject to uncertain initial conditions and control inputs:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t)), \quad t \in [t_0, t_f], \tag{1.1a}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{1.1b}$$

for $\mathbf{u}$ and $\mathbf{x}_0$ in some set of admissible controls $\mathcal{U}$ and initial conditions X_0, respectively. While we refer to the inputs $\mathbf{u}$ as controls, they can model uncertain time-varying inputs in general. The goal is to analyze or estimate the set of states reachable by solutions of IVP (1.1) for all possible controls and initial conditions, otherwise known as the reachable set. Specifically, we define the reachable set at time t by

$$R(t) \equiv \{\mathbf{x}(t) : \mathbf{x} \text{ is a solution of IVP (1.1) for some } (\mathbf{u}, \mathbf{x}_0) \in \mathcal{U} \times X_0, \}.$$

Computing the reachable set is complicated by the fact that an exact analytical solution of IVP (1.1) is often very difficult to obtain in engineering applications.

When the inputs $\mathbf{u}$ are *parameters*, i.e. uncertain but not time-varying, the dynamic problem reduces to an IVP in parametric ODEs. When some sort of probability distribution is known for these parameters, approaches focus on more probabilistic descriptions of the reachable set, in terms of expectations or moments. One such approach is based on polynomial chaos expansions [132, 211], in which the dependence of the solutions of IVP (1.1) is approximated by an infinite linear combination (series) of polynomials of the parameters.

The approach to reachability analysis taken in this thesis is inspired more by worst-case analysis of IVP (1.1). In chemical engineering applications, IVP (1.1) often models systems that have hard constraints. As the name suggests, hard constraints *must* be satisfied for safety reasons, for instance. Consequently, it is useful to think of the constraints as determining "safe" and "unsafe" regions of state space. The result is that descriptions of the reachable set in terms of expectations are typically unacceptable, as we wish to know with 100% certainty that the system can never reach an unsafe operating region. Furthermore, such approaches based on polynomial chaos expansions require approximations, and error bounds are often unavailable in the case of nonlinear systems (in [174], despite an exhaustive consideration of potential sources of error, a bound on the error due to truncating the infinite expansion was only briefly considered, and overall this seems to be an open ques-

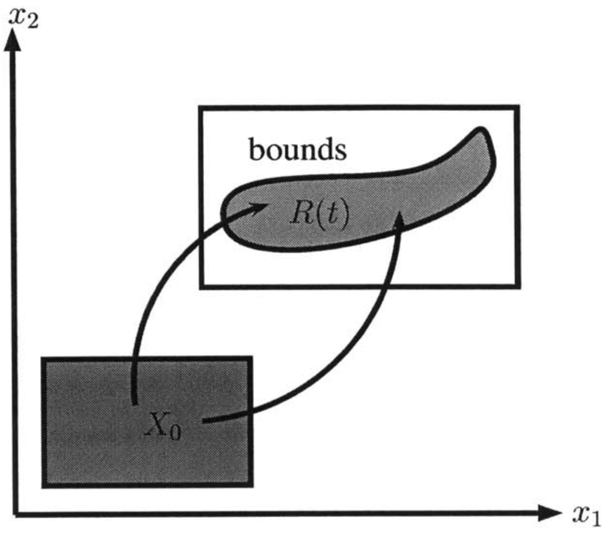


Figure 1-1: A conceptual representation of the forward reachability problem. The set of initial conditions X_0 yields the reachable set $R(t)$ at some time t under some dynamic system. One focus of this thesis is the calculation of enclosures or bounds on the set $R(t)$.

tion). Overall, methods based on polynomial chaos expansions or probabilistic views of the underlying uncertainty typically aim to answer different questions than in a worst-case or safety analysis of a system.

As a consequence, we focus on the case that controls take values in some known set, rather than according to some probability distribution, and the desired estimate of the reachable set is an enclosure of it. We refer to an enclosure as "bounds" on the reachable set. See Fig. 1-1 for a conceptual representation. One area that provides relevant theory for constructing bounds is viability theory [10]. The concepts of viability tubes and kernels are related to the reachability problem, although in their full generality, they are too abstract to be of much use numerically; for basic schemes see [44, §7] and [162]. However, the general theory can be specialized to give bounding theorems based on differential inequalities [72, 97, 152, 168]. These theories focus on constructing interval bounds through the solution of an auxiliary IVP in ODEs. Consequently, sophisticated methods and software for the solution of IVPs in ODEs can be used to approximate the solution of this auxiliary system numerically. The resulting numerical methods for constructing bounds on the reachable set can be implemented very efficiently, as demonstrated in [168, 166]. This thesis follows this general approach; see further discussion in §1.2.1.

Related methods for reachability analysis come from validated integration methods

[26, 107, 133]. Even when no uncertainty is present, these methods provide enclosures of the solutions of IVPs in ODEs that are valid despite integration errors and errors due to round-off (similarly to interval arithmetic [130], the original aim of these methods was to overcome the issues inherent in dealing with real numbers in finite precision arithmetic). However, these methods can handle only a limited amount of parametric uncertainty compared to some methods based on differential inequalities as shown in [152]. A more successful extension of these methods to handle uncertain parameters was introduced in [107]. Subsequent extensions and improvements were made in [160]. These extensions depend on Taylor model arithmetic [114, 136], an extension of interval arithmetic, to handle dependencies on uncertain parameters in a rigorous way. Roughly, a Taylor model of a function is a Taylor polynomial approximation with a rigorous error bound. One disadvantage of these extensions is a rapid increase in the number of terms in a Taylor model/polynomial as the order (and typically, accuracy) of the Taylor polynomial and dimension (number of uncertain parameters and initial conditions) increases; the number of terms in an order q Taylor polynomial with respect to p parameters is approximately $p^q/q!$ [65, Ch. 13]. The efficiency of these methods is constantly improving, however the work in [37, 202] has demonstrated that the quality of the bounds produced by these methods can be improved by using bounding theories based on differential inequalities.

1.1.2 Global dynamic optimization

To introduce further motivation for this thesis' approach to reachability analysis, consider a system modeled by IVP (1.1) and the case that there is a safety constraint on the states at the final time, $g(\mathbf{x}(t_f)) \leq 0$, for any solution $\mathbf{x}$ of IVP (1.1) and any admissible control and initial condition. The problem of ensuring that the reachable set at the final time does not intersect the unsafe region of state space can be formulated as the maximization problem

$$\begin{aligned} g^* = \sup_{\mathbf{u},\mathbf{x}_0} \; & g(\mathbf{x}(t_f)) \qquad (1.2) \\ \text{s.t.}\; & \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t)), \quad t \in [t_0, t_f], \\ & \mathbf{x}(t_0) = \mathbf{x}_0, \\ & (\mathbf{u}, \mathbf{x}_0) \in \mathcal{U} \times X_0. \end{aligned}$$

If the optimal objective value g^* (or an upper bound of g^*) is less than or equal to zero, we can ensure that the system obeys the constraints for all admissible controls and initial conditions. It should be stressed that this requires *global* information about Problem (1.2), especially if safety is concerned. A suboptimal (or local) solution $(\mathbf{u}^\dagger, \mathbf{x}_0^\dagger)$ of Problem (1.2) with corresponding objective value $g^\dagger \leq 0$ does *not* guarantee that the system is safe for all admissible controls and initial conditions. Problem (1.2) is called a global dynamic optimization problem or optimal control problem. This is an extremely important class of problems to many fields and industries and has applications beyond just safety analysis. The books [13, 22, 27, 198] and recent theses [160, 163, 166, 175] are devoted to this problem and attest to its importance.

This connection between reachability analysis and global dynamic optimization is quite important and further motivates the approaches in this thesis. Theory and analysis of optimal control problems have a long history, with topics including dynamic programming and Pontryagin's principle [13]. However, since the focus is on global solution of optimal control problems, and the Pontryagin principle is only a necessary optimality condition and may yield sub-optimal solutions, the most relevant framework is dynamic programming and the related Hamilton-Jacobi-Bellman (HJB) partial differential equation (PDE). In fact, this theory can be adapted to give a direct approach to calculating the reachable set; see [84, 103, 121], and for a better explanation of a related problem, see [31]. Although the solution of this PDE yields an exact representation of the reachable set, analytical solutions are typically not possible. Numerical solution is possible; in [31, 102, 121], numerical solutions are obtained from level-set methods, a class of finite-difference methods (see [141]). Whether the PDE is solved by finite differences, finite volumes, or finite elements, the solution obtained is essentially a finite grid or mesh of points in state space, and at each point is a value that determines whether the mesh point is in the reachable set or not. This lends itself to nice visualization in 3 or fewer dimensions. However, in general, practically using this solution in subsequent analysis or computation would require extra calculation to obtain a more easily represented or manipulated set, such as an interval, ellipsoid, or (convex) polyhedron, which somewhat defeats the purpose of calculating the exact reachable set. If the ultimate goal is to solve an optimal control problem, then there are still issues, as solving the HJB equation in the first place is computationally demanding, as noted in [84], with the additional complication that the equation itself is defined by a potentially nonconvex global optimization problem. This has restricted its application to systems with specific structure, such as systems with linear dynamics or systems which permit an explicit expression to be written for the embedded global optimization problem defining the HJB equation; see [31, 84, 102, 103, 121].

Another class of methods that can solve dynamic optimization problems globally is the class of "direct" methods, and in particular, the sequential or control parameterization approach. Control problems such as (1.2) are infinite dimensional, since the decision space includes $\mathcal{U}$, a subset of a function space. As the name suggests, the control parameterization approach handles this issue by parameterizing the controls with a finite number of parameters, thus transforming the infinite dimensional problem into a finite one. Consequently, a wealth of methods and theory from finite nonlinear programming can be applied to the problem. See [198], in particular Ch. 6, for more on the control parameterization approach, and [27], in particular Ch. 4, for another direct method often called the simultaneous approach. As noted in §1.1 of [163], the control parameterization approach reduces the number of optimization variables compared to the simultaneous approach and thus is better suited to the application of deterministic global optimization methods.

As mentioned earlier, we focus on global optimization as this yields the only acceptable information in applications such as safety analysis, and similarly, we focus on deterministic (as opposed to stochastic) methods, as these yield more rigorous global information (see [137] for a review of global optimization methods). Branch and bound provides a promising framework. This deterministic global optimization method rigorously searches the entire decision space and requires upper and lower bounds on the objective. In "normal" optimization problems, i.e. problems with data given by explicitly defined functions, interval arithmetic and convex and concave relaxations give this global information [57, 117, 130],[137, §16]. Thus, to apply branch and bound to dynamic optimization problems via the control parameterization approach, convex and concave relaxations of the solutions of IVPs in parametric ODEs are required. This has been achieved in [169, 176]. Both of these methods require interval enclosures of the reachable set of a dynamic system. This provides further motivation for the approaches taken in this thesis toward estimating the reachable set. The branch and bound algorithm requires these relaxations many times, and so tight, but efficiently calculated interval enclosures are desired. The methods based on differential inequalities mentioned earlier have yielded promising results and merit further investigation.

1.1.3 Robust design

In a dynamic setting, the problem of robust design is complementary to the forward reachability problem. Consider IVP (1.1) in the case that it depends on parameters (i.e. uncertain but constant inputs). Suppose that a safety analysis has determined that the set of parameters and initial conditions is not acceptable; there is some parameter/initial condition pair that results in a solution entering an unsafe region of state space. A natural recourse is to try to find a set of inputs that *does* guarantee safe system behavior for all resulting solutions. Robust design can help solve this problem.

A general form of the problem is

$$\begin{aligned} &\max_{\mathbf{y}} \text{volume}(D(\mathbf{y})) \\ &\text{s.t. } \widehat{\mathbf{g}}(\mathbf{p}) \leq \mathbf{0}, \quad \forall \mathbf{p} \in D(\mathbf{y}), \\ &\quad \mathbf{y} \in Y, \end{aligned} \tag{1.3}$$

for some decision space Y. In this problem, $D(\mathbf{y})$ is a candidate "design space," a set of parameter values which could potentially be realized in the operation of an uncertain system. This system has design/operational constraints expressed as $\widehat{\mathbf{g}}(\mathbf{p}) \leq \mathbf{0}$; the function $\widehat{\mathbf{g}}$ likely will be expressed in terms a mathematical model of the process (see Problem (1.4) below). Consequently, a design space is feasible if and only if every possible parameter in it satisfies the constraints. The goal is to maximize operational flexibility by choosing the largest design space. Suppose $D(\mathbf{y})$ is expressed as deviation from some central or nominal point; let $\mathbf{y} = (\mathbf{p}_c, \delta)$ and $D(\mathbf{p}_c, \delta) = \{\mathbf{p} : \|\mathbf{p} - \mathbf{p}_c\| \leq \delta\}$. Then if the optimal solution is $(\mathbf{p}_c, \delta)$, $\mathbf{p}_c$ is a parameter corresponding to a robust system; operation of the system at this setpoint affords the most robustness to noise in operation or uncertainty in the model by ensuring that the constraints are satisfied for any other parameter value within δ of $\mathbf{p}_c$ (in some norm). Knowing the value of this acceptable deviation is an important part of the solution; it permits easy monitoring of whether the process is obeying the desired constraints.

As in the forward reachability problem, some approaches to this and related problems assume that probability distributions are available for the uncertain parameters [156, 173, 215]. The resulting formulations have chance constraints; the probability that the constraints $\widehat{\mathbf{g}}(\mathbf{p}) \leq \mathbf{0}$ hold must be greater than some threshold. Again, for the approach taken in this thesis, this is unacceptable.

The approach taken here is inspired more by the work on feasibility and flexibility in [68, 195, 196]. These formulations are based on worst-case or bounded uncertainty analysis. Extensions to more challenging models have been considered in [5, 43, 193]. See also [66, 213] for reviews of related problems. Similarly to these other approaches, the current approach is characterized by the fact that the fundamental problem (Problem (1.3)) is *infinitely constrained* (the constraints $\widehat{\mathbf{g}}(\mathbf{p}) \leq \mathbf{0}$ must hold for *all* $\mathbf{p}$ in the set $D(\mathbf{y})$, which typically will be infinite, and even further, uncountable). In contrast with the dynamic optimization problem (1.2), we focus on the case that Problem (1.3) has a finite number of decision variables $\mathbf{y}$. The result is that Problem (1.3) is a type of generalized semi-infinite program (GSIP) [67, 182, 187, 188]. The theory and methods for this class of problems will be vital in the approach to robust design taken in this thesis.

Robust design is also related to design centering, a specific case of GSIP. This class of problems has the intuitive geometric interpretation of inscribing the largest set (chosen from some class of sets) into some container set. In terms of (1.3), the constraints define the container set $G = \{\mathbf{p} : \widehat{\mathbf{g}}(\mathbf{p}) \leq \mathbf{0}\}$, into which we aim to fit the largest $D(\mathbf{y})$ for $\mathbf{y} \in Y$. See Fig. 1-2. A classic application is to lapidary cutting problems [138, 210], where the largest gem or gems are to be cut from a given rough stone. In thc dynamic setting, this highlights a conceptual difference between forward reachability and robust design in this thesis; forward reachability seeks an enclosure or superset, robust design seeks a subset.

When the robust design problem is dynamic in nature, $\widehat{\mathbf{g}}$ may be defined in part by the solution of an IVP in parametric ODEs. For instance, the robust design problem might have the form

$$\begin{aligned} &\max_{\mathbf{y}} \text{volume}(D(\mathbf{y})) && (1.4)\\ &\text{s.t. } \mathbf{g}(\mathbf{x}(t_f, \mathbf{p}, \mathbf{x}_0)) \leq \mathbf{0}, \quad \forall (\mathbf{p}, \mathbf{x}_0) \in D(\mathbf{y}), \\ &\qquad \dot{\mathbf{x}}(t, \mathbf{p}, \mathbf{x}_0) = \mathbf{f}(t, \mathbf{p}, \mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)), \quad t \in [t_0, t_f], \\ &\qquad \mathbf{x}(t_0, \mathbf{p}, \mathbf{x}_0) = \mathbf{x}_0, \\ &\qquad \mathbf{y} \in Y. \end{aligned}$$

Here $\widehat{\mathbf{g}}(\mathbf{p}, \mathbf{x}_0) = \mathbf{g}(\mathbf{x}(t_f, \mathbf{p}, \mathbf{x}_0))$. The problem of "backward reachability" is related; in its

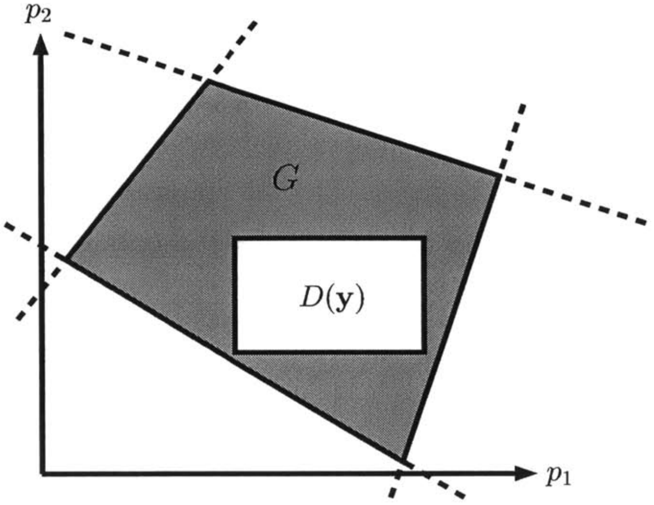


Figure 1-2: A geometric interpretation of the robust design problem. The container set G could represent parameter values which yield safe system behavior. The set $D(\mathbf{y})$ is a feasible design space since it is a subset of G; the goal is to calculate a design space which affords the most flexibility.

basic form, this problem aims to determine the set of initial conditions from which a solution of an IVP in ODEs can reach a given target set. We could treat parameters as states with zero time derivative and adapt this approach to solve Problem (1.4); we then determine the backward reachable set corresponding to the set of states $\{\mathbf{z} : \mathbf{g}(\mathbf{z}) \leq \mathbf{0}\}$. Methods from optimal control involving the HJB equation mentioned earlier could be used, but in addition to the aforementioned numerical difficulties, there is the drawback that the solution obtained does not necessarily define an easily represented, regular set such as an interval or ellipse. As discussed above, the solution of a robust design problem should yield an acceptable deviation (e.g. δ). Inner approximations by ellipsoids have been proposed [38, 142], but these are specific to linear dynamics. Meanwhile, the theory and methods of GSIP, combined with techniques from global dynamic optimization and applied to Problem (1.4), promise a flexible framework for addressing the backward reachability problem.

1.2 Contributions

The overall structure of this thesis and its contributions are discussed in the following sections.

1.2.1 Forward reachability

Chapters 5, 6, and 7 build up the approaches to the forward reachability problem. Chapter 5 develops a numerical method for the construction of interval bounds on the reachable set of IVP (1.1). The theory and numerical methods are then generalized to the construction of polyhedral bounds for dynamic systems in Ch. 6. Although methods for constructing polyhedral bounds exist, this thesis broadens the class of problems for which polyhedral bounds can be constructed, and addresses a number of practical numerical issues that limit the efficiency of previous methods. For instance, previous work focuses on linear dynamics [84], partitions the state space and approximates the dynamics on these regions [9], requires that nonconvex global optimization problems be approximated [39], or manually implements the time steps of the numerical integration [63, 64]. Another contribution of this thesis is the development of a special procedure for constructing parameterized affine relaxations of general functions (see §1.2.3 below). The result is that the bounding methods developed in this thesis depend on parametric linear programs, can be used with established methods for numerical integration, and can be applied to general nonlinear systems. The result is expanded applicability and efficient implementation.

Chapter 6 also explores problem formulations and structure that can be exploited to nontrivially improve bounds without a significant increase in computational cost. This relates to using constraint information or *a priori* enclosures of the reachable set of the system. This information is often available for models of chemical reactors, and this thesis has identified a way to extend the results to continuously-stirred tank reactors that previously only applied to batch reactors [167].

Chapter 7 develops a general bounding theory and explores the connections to many existing theorems. This theory takes into consideration state constraints, which has applications to path constraints in dynamic optimization, measurements in state estimation, and differential-algebraic equations. In addition, this leads to a theory for constructing relaxations of the solutions of parametric ordinary differential equations, which is vital to deterministic global dynamic optimization. The new theory is flexible and numerical experiments show that an implementation is fast and the resulting relaxations are tighter than some comparable previous methods.

1.2.2 Robust design

Chapters 8 and 9 consider generalized semi-infinite programming and robust design. This thesis develops theoretical results and numerical analysis of methods for solving certain classes of GSIP which lead to numerical methods for solving robust design problems. The result is an effective method for general, nonlinear models. This is achieved by using Lagrangian duality results to formulate a semi-infinite program (SIP) restriction of Problem (1.3) and subsequently apply a numerical method for SIP. Most of this theoretical and numerical work is the focus of Ch. 8.

In particular, this method is applicable and efficient even when the robust design problem is dynamic in nature, that is, similar in form to Problem (1.4). The numerical method can be tailored to take advantage of the structure of the problem in order to avoid some of the computational expense associated with the dynamic nature of the problem. Although the method is approximate, any design space that the method produces is guaranteed to be feasible and an example shows that it is effective. This is discussed in Ch. 9.

1.2.3 Parametric affine relaxations

In support of the forward reachability problem, two other contributions are made. The first of these is a method for calculating affine relaxations of general functions. In particular, these relaxations are desired in a dynamic setting, and specific parametric regularity properties must hold. This is the subject of Ch. 3.

1.2.4 Ordinary differential equations with linear programs embedded

The second topic explored in connection with the forward reachability problem is discussed in Ch. 4. This relates to the IVP in ODEs with a lexicographic linear program embedded:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{q}(t, \mathbf{x}(t))), \quad t \in [t_0, t_f], \tag{1.5}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{1.6}$$

where

$$\begin{aligned} q_1(t,\mathbf{z}) = \min_{\mathbf{v}\in\mathbb{R}^{n_v}} \; & \mathbf{c}_1^{\mathrm{T}}\mathbf{v} \\ \text{s.t. } & \mathbf{A}\mathbf{v} = \mathbf{b}(t,\mathbf{z}), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{1.7}$$

and for $i \in \{2,\ldots,n_q\}$,

$$\begin{aligned} q_i(t,\mathbf{z}) = \min_{\mathbf{v}\in\mathbb{R}^{n_v}} \; & \mathbf{c}_i^{\mathrm{T}}\mathbf{v} \\ \text{s.t. } & \begin{bmatrix} \mathbf{A} \\ \mathbf{c}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{c}_{i-1}^{\mathrm{T}} \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{b}(t,\mathbf{z}) \\ q_1(t,\mathbf{z}) \\ \vdots \\ q_{i-1}(t,\mathbf{z}) \end{bmatrix}, \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{1.8}$$

Understanding the potential numerical difficulties of this problem is important to the numerical methods developed in Chapters 5 and 6.

In addition, the problem of ODEs with a linear program embedded has merit on its own. Models of bioreactors based on dynamic flux balance analysis (DFBA) take this general form [70, 79, 81]. This thesis develops the first rigorous mathematical analysis of this industrially-relevant modeling framework, and subsequently develops an efficient and robust numerical method for handling models of this form.

The numerical challenges in handling DFBA models include potential nonuniqueness of the solution set of LP (1.7), potential infeasibility of LP (1.7), and overall nonsmoothness of the vector field of the ODE (right-hand side of (1.5)). The numerical method developed approaches the problem using tools from hybrid systems theory and parametric programming to reformulate the system as index-one differential-algebraic equations (DAEs) with discrete modes. Powerful methods for the integration of index-one DAEs and event detection can then be used. This approach addresses the issues of infeasibility and nonsmoothness. Meanwhile, considering lexicographic linear programs allows modelers to overcome the issue of nonuniqueness. A way of efficiently handling the lexicographic linear program in the context of the overall numerical method is developed.

Chapter 2

Preliminaries

This chapter introduces some general notation, technical results, and technical concepts that will be used in this thesis.

2.1 Notation

The following notation and terminology is fairly standard but we include this discussion for completeness. Let $\mathbb{N}$ and $\mathbb{R}$ denote the natural and real numbers, respectively. For $(m,n) \in \mathbb{N}^2$, let $\mathbb{R}^{m \times n}$ denote the set of $m \times n$ real matrices. Vectors and matrices are denoted with lowercase bold letters (e.g. $\mathbf{v}$) and uppercase bold letters (e.g. $\mathbf{M}$), respectively. The transposes of a vector $\mathbf{v}$ and matrix $\mathbf{M}$ are denoted $\mathbf{v}^{\mathrm{T}}$ and $\mathbf{M}^{\mathrm{T}}$, respectively. The exception is that $\mathbf{0}$ may denote either a matrix or vector of zeros, but it should be clear from context what the appropriate dimensions are. Similarly, $\mathbf{1}$ denotes a vector of ones and $\mathbf{I}$ denotes the identity matrix where the dimensions should be clear from context. The j^{th} component of a vector $\mathbf{v}$ is denoted v_j. For $(p,n) \in \mathbb{N}^2$ and a matrix $\mathbf{M} \in \mathbb{R}^{p \times n}$, the notation $\mathbf{M} = [\mathbf{m}_i^{\mathrm{T}}]$ may be used to emphasize that the i^{th} row of $\mathbf{M}$ is $\mathbf{m}_i$, for $i \in \{1, \dots, p\}$. Similarly, $\mathbf{M} = [m_{i,j}]$ emphasizes that the element in the i^{th} row and j^{th} column is $m_{i,j}$.

For $(t_1, t_2) \in \mathbb{R}^2$, with $t_1 \leq t_2$, a nonempty interval is denoted $[t_1, t_2]$. If $t_1 < t_2$, an open interval is denoted (t_1, t_2); it should be clear from context that this is a subset of $\mathbb{R}$ and not a point in $\mathbb{R}^2$. "Half-open intervals" are denoted $[t_1, t_2)$ and $(t_1, t_2]$ (see also Definition 2.17 of [158]).

Inequalities between vectors hold componentwise. For $n \in \mathbb{N}$ and $(\mathbf{v}, \mathbf{w}) \in \mathbb{R}^n \times \mathbb{R}^n$, $[\mathbf{v}, \mathbf{w}] \equiv [v_1, w_1] \times [v_2, w_2] \times \cdots \times [v_n, w_n]$ denotes an interval in $\mathbb{R}^n$. Note that this interval may be empty; $[\mathbf{v}, \mathbf{w}]$ is nonempty if and only if $\mathbf{v} \leq \mathbf{w}$. Denote the midpoint of an interval $X = [\mathbf{v}, \mathbf{w}]$ by $\text{mid}(X) \equiv {}^1\!/\!{}_2(\mathbf{v} + \mathbf{w})$.

For $(m, n) \in \mathbb{N}^2$, a polyhedron is any subset of $\mathbb{R}^n$ that can be expressed as $\{\mathbf{z} \in \mathbb{R}^n : \mathbf{Mz} \leq \mathbf{d}\}$, for some matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ and vector $\mathbf{d} \in \mathbb{R}^m$ (i.e. it is the intersection of a finite number of closed halfspaces). Consequently, polyhedra are always closed, convex sets.

For $n \in \mathbb{N}$, the equivalence of norms on $\mathbb{R}^n$ is used often; when a statement or result holds for any choice of norm, it is denoted $\|\cdot\|$. In some cases, it is useful to reference a specific norm, in which case it is subscripted; for instance, $\|\cdot\|_1$ denotes the 1-norm. The dual norm of a norm $\|\cdot\|$ is denoted $\|\cdot\|_*$.

For sets X, Y, a mapping S from X to the set of subsets of Y is denoted $S : X \rightrightarrows Y$. For sets X, Y, and Z with $X \subset Z$ and $Y \subset Z$, the difference between X and Y (X intersected with the complement of Y in Z) is denoted $X \backslash Y$. In a metric space, a neighborhood of a point x is denoted $N(x)$ and refers to an open ball centered at x with some nonzero radius. If this radius δ is important, it may be emphasized as a subscript, e.g. $N_\delta(x)$. The closure of a set S is denoted $\overline{S}$. The diameter of a nonempty set $S \subset \mathbb{R}^n$ is defined as $\text{diam}(S) = \sup\{\|\mathbf{z}_1 - \mathbf{z}_2\|_\infty : (\mathbf{z}_1, \mathbf{z}_2) \in S \times S\}$. This coincides with the definition of the width of an interval.

Differentiability of functions is understood in the Fréchet sense. For $(m, n) \in \mathbb{N}^2$, open $D \subset \mathbb{R}^n$, and $\mathbf{f} : D \to \mathbb{R}^m$ which is differentiable at $\mathbf{x}_0 \in D$, the Jacobian matrix of $\mathbf{f}$ at $\mathbf{x}_0$ exists and is denoted by $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_0) \in \mathbb{R}^{m \times n}$. The gradient (the transpose of the Jacobian) is denoted $\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}_0)$ or just $\nabla \mathbf{f}(\mathbf{x}_0)$. When $n = 1$, we allow the domain D to be an open, closed, or half-open interval, and denote the derivative at $t \in D$ (if it exists) by $\dot{\mathbf{f}}(t)$, using "left-hand" or "right-hand" derivatives at the boundary as necessary.

For $n \in \mathbb{N}$ and a set $T \subset \mathbb{R}$, denote the Lebesgue space $L^1(T, \mathbb{R}^n) \equiv \{(\mathbf{v} : T \to \mathbb{R}^n) : \int_T |v_i| < +\infty, \forall i\}$. That is, $\mathbf{v} \in L^1(T, \mathbb{R}^n)$ if each component of $\mathbf{v}$ is in $L^1(T) \equiv L^1(T, \mathbb{R})$. Statements that hold at almost every $t \in T$ (i.e. except on a subset of Lebesgue measure zero) are abbreviated *a.e.* $t \in T$.

A vector-valued function $\mathbf{f}$ is called convex if each component f_i is convex, and similarly for concavity.

2.2 Compact analysis

Some concepts dealing with compact sets are discussed. In the following, (X, d) is a metric space (with metric d).

The set of nonempty compact subsets of (X, d) is denoted $\mathbb{K}X$. Define the distance from a point x to a set Y in (X, d) by

$$d(x, Y) \equiv \inf\{d(x, y) : y \in Y\}.$$

The Hausdorff distance d_H between two sets Y, Z in (X, d) is given by

$$d_H(Y, Z) = \max\left\{\sup\{d(y, Z) : y \in Y\}, \sup\{d(z, Y) : z \in Z\}\right\}.$$

If for all $y \in Y$ there exists a $z \in Z$ such that $d(y, z) \leq \delta$ and vice versa, then $d_H(Y, Z) \leq \delta$. Conversely, if Y and Z are compact and $d_H(Y, Z) \leq \delta$, then for all $y \in Y$ there exists $z \in Z$ with $d(y, z) \leq \delta$ and for all $z \in Z$ there exists $y \in Y$ with $d(z, y) \leq \delta$. The Hausdorff distance defines a metric on $\mathbb{K}X$. For more discussion and related topics see §5 of [51].

A space is locally compact if every point has a compact neighborhood (i.e. every point is contained in the interior of a compact set). It follows that $\mathbb{R}^n$ is locally compact. Further, any closed or open subset of a locally compact space is locally compact as well; see [131].

Lemma 2.2.1. *Let (X, d) be a metric space. Let $\mathcal{K}$ be a compact subset of $(\mathbb{K}X, d_H)$. Then $\widehat{K} = \bigcup_{Z \in \mathcal{K}} Z$ is compact.*

Proof. Choose a sequence $\{x_i\} \subset \widehat{K}$. We will show that a subsequence of it converges to an element of $\widehat{K}$. By the definition of $\widehat{K}$, we can construct a corresponding sequence $\{Z_i\} \subset \mathcal{K}$ such that $Z_i \ni x_i$ for each i. Since $\mathcal{K}$ is compact, there exists a subsequence $\{Z_{i_j}\}$ which converges (with respect to the Hausdorff metric) to some $Z^* \in \mathcal{K}$. Using the definition of the Hausdorff metric and the fact that Z_i and Z^* are compact, we have

$$\forall \epsilon > 0, \exists J > 0 \text{ such that } \forall j > J, \exists z_j \in Z^* \text{ such that } d(x_{i_j}, z_j) \leq \epsilon.$$

It follows that we can construct a subsequence of $\{x_{i_j}\}$, which we will denote $\{x_\ell\}$, and $\{z_\ell\} \subset Z^*$ such that $\forall \epsilon > 0, \exists L > 0$ such that $\forall \ell > L, d(x_\ell, z_\ell) \leq \epsilon$. But since Z^* is compact, a subsequence $\{z_{\ell_m}\}$ converges to some $z \in Z^*$. Using the triangle inequality, we have $\forall \epsilon > 0, \exists M > 0$ such that $\forall m > M$, $d(x_{\ell_m}, z) \leq d(x_{\ell_m}, z_{\ell_m}) + d(z_{\ell_m}, z) < \epsilon$. Thus, $\{x_{\ell_m}\}$ converges to $z \in \widehat{K}$, and so $\widehat{K}$ is compact. $\square$

2.3 Local Lipschitz continuity

The concepts of Lipschitz continuity and local Lipschitz continuity are central to many results in this thesis.

Definition 2.3.1. Let (X, d_X) and (Y, d_Y) be metric spaces. A mapping $f : X \to Y$ is Lipschitz continuous if there exists $L > 0$ such that

$$d_Y(f(x_1), f(x_2)) \leq L d_X(x_1, x_2)$$

for all $(x_1, x_2) \in X \times X$.

Definition 2.3.2. Let (X, d_X) and (Y, d_Y) be metric spaces. A mapping $f : X \to Y$ is locally Lipschitz continuous if for all $x \in X$ there exists a neighborhood $N(x)$ of x and $L(x) > 0$ such that

$$d_Y(f(x_1), f(x_2)) \leq L(x) d_X(x_1, x_2)$$

for all $(x_1, x_2) \in N(x) \times N(x)$.

Compactness and local compactness allow us to infer Lipschitz continuity on subsets from local Lipschitz continuity.

Lemma 2.3.1. *If f is locally Lipschitz continuous on a metric space (X, d), then f is Lipschitz continuous on any compact subset of X. If (X, d) is locally compact and f is Lipschitz continuous on every compact subset of X, then f is locally Lipschitz continuous on X.*

The following establishes that the composition of locally Lipschitz continuous mappings is also locally Lipschitz continuous. See Theorem 2.5.6 in [166] for its proof.

Lemma 2.3.2. *Let (X, d_X), (Y, d_Y), and (Z, d_Z) be metric spaces and let $f : X \to Y$ and $g : Y \to Z$ be locally Lipschitz continuous. Then $g \circ f : X \to Z$ is locally Lipschitz continuous.*

This composition result is very useful and permits us to infer the local Lipschitz continuity of the (finite) sum, (finite) product, maximum, minimum, etc., of locally Lipschitz continuous mappings.

2.4 Parametric programming

A few results regarding parametric optimization problems, in particular parametric linear programs (LPs), are considered. A general resource for related results is [12].

Lemma 2.4.1. *Let (X, d_X) and (Y, d_Y) be metric spaces. Assume $f : X \times Y \to \mathbb{R}$ and $M : Y \rightrightarrows X$ are mappings such that*

1. *M is nonempty-valued,*
2. *$f(\cdot, y)$ attains its infimum on $M(y)$ for each $y \in Y$,*
3. *there exists $L_f > 0$ such that for all (x_1, y_1) and $(x_2, y_2) \in X \times Y$, f satisfies*

$$|f(x_1, y_1) - f(x_2, y_2)| \leq L_f(d_X(x_1, x_2) + d_Y(y_1, y_2)),$$

4. *there exists $L_M > 0$ such that for all $(y_1, y_2) \in Y \times Y$ and for all $x_1 \in M(y_1)$, there exists $x_2 \in M(y_2)$ such that $d_X(x_1, x_2) \leq L_M d_Y(y_1, y_2)$.*

Then $f_{\min} : Y \ni y \mapsto \min\{f(x, y) : x \in M(y)\}$ is Lipschitz continuous.

Proof. Choose $(y_1, y_2) \in Y \times Y$. By assumption, $f(\cdot, y_1)$ achieves its minimum on $M(y_1)$, thus $f_{\min}(y_1) = f(x_1, y_1)$ for some $x_1 \in M(y_1)$. By assumption there is a $L_M > 0$ and $\widetilde{x}_2 \in M(y_2)$ such that $d_X(x_1, \widetilde{x}_2) \leq L_M d_Y(y_1, y_2)$. Because f is Lipschitz continuous, there is a $L_f > 0$ such that

$$|f(x_1, y_1) - f(\widetilde{x}_2, y_2)| \leq L_f\left(d_X(x_1, \widetilde{x}_2) + d_Y(y_1, y_2)\right) \leq L_f(L_M + 1) d_Y(y_1, y_2).$$

This implies that

$$f(\widetilde{x}_2, y_2) \leq L_f(L_M + 1) d_Y(y_1, y_2) + f_{\min}(y_1),$$

which implies that

$$f_{\min}(y_2) \leq f(\widetilde{x}_2, y_2) \leq L_f(L_M + 1) d_Y(y_1, y_2) + f_{\min}(y_1),$$

thus

$$f_{\min}(y_2) - f_{\min}(y_1) \leq L_f(L_M + 1)d_Y(y_1, y_2). \tag{2.1}$$

Similarly, $f(\cdot, y_2)$ achieves its minimum on $M(y_2)$, thus $f_{\min}(y_2) = f(x_2, y_2)$ for some $x_2 \in M(y_2)$ and there is a $\widetilde{x}_1 \in M(y_1)$ such that $d_X(\widetilde{x}_1, x_2) \leq L_M d_Y(y_1, y_2)$. By reasoning similar to above, $f(\widetilde{x}_1, y_1) \leq L_f(L_G + 1)d_Y(y_1, y_2) + f(x_2, y_2)$ thus

$$f_{\min}(y_1) - f_{\min}(y_2) \leq L_f(L_G + 1)d_Y(y_1, y_2)$$

which combined with Eqn. (2.1) gives

$$|f_{\min}(y_1) - f_{\min}(y_2)| \leq L_f(L_G + 1)d_Y(y_1, y_2)$$

as desired. □

The following result establishes "Lipschitz continuity" of the feasible sets, solution sets, and optimal objective value of an LP parameterized by the "right-hand side" of its constraints. This result is from the literature; see for instance Theorem 2.4 of [115].

Lemma 2.4.2. *Assume $(m, n) \in \mathbb{N}^2$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$. Consider the linear program*

$$q(\mathbf{b}) = \sup\{\mathbf{c}^{\mathrm{T}}\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}\}.$$

Let $P(\mathbf{b}) = \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}\}$ (the feasible set), $S(\mathbf{b}) = \{\mathbf{z} \in P(\mathbf{b}) : \mathbf{c}^{\mathrm{T}}\mathbf{z} = q(\mathbf{b})\}$ (the solution set), $F = \{\mathbf{b} : P(\mathbf{b}) \neq \varnothing\}$, and $F_S = \{\mathbf{b} : S(\mathbf{b}) \neq \varnothing\}$. Then for any choice of norms $\|\cdot\|_\alpha$, $\|\cdot\|_\beta$,

1. *there exists $L > 0$ such that for all $(\mathbf{b}_1, \mathbf{b}_2) \in F \times F$ and for any $\mathbf{z}_1 \in P(\mathbf{b}_1)$, there exists a $\mathbf{z}_2 \in P(\mathbf{b}_2)$ with*

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_\alpha \leq L \|\mathbf{b}_1 - \mathbf{b}_2\|_\beta,$$

2. *there exists $L_S > 0$ such that for all $(\mathbf{b}_1, \mathbf{b}_2) \in F_S \times F_S$ and for any $\mathbf{z}_1 \in S(\mathbf{b}_1)$, there exists a $\mathbf{z}_2 \in S(\mathbf{b}_2)$ with*

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_\alpha \leq L_S \|\mathbf{b}_1 - \mathbf{b}_2\|_\beta,$$

3. and $q : F_S \rightarrow \mathbb{R}$ is Lipschitz continuous.

We prove a related result that establishes local Lipschitz continuity of the optimal objective value of an optimization problem parameterized by its objective and feasible set.

Lemma 2.4.3. *Assume $(m, n, p) \in \mathbb{N}^3$. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $I = \{1, \dots, p\}$. Let $P : \mathbb{R}^m \ni \mathbf{b} \mapsto \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}\}$ and $F = \{\mathbf{b} : P(\mathbf{b}) \neq \varnothing\}$. Assume that $P(\mathbf{b})$ is bounded for all $\mathbf{b} \in F$. Define $q : F \times \mathbb{R}^{pn} \times \mathbb{R}^p \rightarrow \mathbb{R}$ by*

$$q : (\mathbf{b}, \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_p, \mathbf{h}) \mapsto \min_{\mathbf{z} \in \mathbb{R}^n} \max_{i \in I} \{\mathbf{c}_i^{\mathrm{T}} \mathbf{z} + h_i\} \tag{2.2}$$
$$\text{s.t. } \mathbf{A}\mathbf{z} \leq \mathbf{b}.$$

Then q is locally Lipschitz continuous.

Proof. An important fact is that F is closed, see §4.7 of [25]. It follows that $F \times \mathbb{R}^{pn} \times \mathbb{R}^p$ is locally compact, and so by Lemma 2.3.1 it suffices to show that q is Lipschitz continuous on any compact subset. So choose compact $K \subset F \times \mathbb{R}^{pn} \times \mathbb{R}^p$. Then there exist compact $K_d \subset F$, $K_c \subset \mathbb{R}^{pn}$, and $K_h \subset \mathbb{R}^p$ such that $K \subset K_d \times K_c \times K_h$.

To apply Lemma 2.4.1, we need to extend the domain of P so that we consider it a function of $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_p) \in K_c$ and $\mathbf{h} \in K_h$ as well. In an abuse of notation, denote this function $P : K_d \times K_c \times K_h \rightrightarrows \mathbb{R}^n$. Since P is compact-valued, Lemma 2.4.2 implies that $P : K_d \times K_c \times K_h \rightarrow \mathbb{KR}^n$ is Lipschitz continuous. Thus the image of $K_d \times K_c \times K_h$ under P, denoted $\mathcal{K}$, is compact in $\mathbb{KR}^n$. By Lemma 2.2.1, $K_v \equiv \bigcup_{Z \in \mathcal{K}} Z$ is compact. Since $f : (\mathbf{z}, \mathbf{b}, \mathbf{c}_1, \dots, \mathbf{c}_p, \mathbf{h}) \mapsto \max_{i \in I}\{\mathbf{c}_i^{\mathrm{T}} \mathbf{z} + h_i\}$ is locally Lipschitz continuous on all of $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{pn} \times \mathbb{R}^p$, it is Lipschitz continuous on $K_v \times K_d \times K_c \times K_h$.

By assumption, $P(\mathbf{b}, \mathbf{c}, \mathbf{h})$ is closed, bounded, nonempty, and a subset of K_v for all $(\mathbf{b}, \mathbf{c}, \mathbf{h}) \in K_d \times K_c \times K_h$, and by Lemma 2.4.2 is Lipschitz continuous in the sense required by Lemma 2.4.1. Thus, $f(\cdot, \mathbf{b}, \mathbf{c}, \mathbf{h})$ achieves its minimum on $P(\mathbf{b}, \mathbf{c}, \mathbf{h})$ for each $(\mathbf{b}, \mathbf{c}, \mathbf{h}) \in K_d \times K_c \times K_h$. So, we can apply Lemma 2.4.1 and obtain that q is Lipschitz continuous on $K_d \times K_c \times K_h$, and thus on K. □

In the special case that the index set $I = \{1\}$, it is clear that optimization problem (2.2) is a linear program parametrized by both its cost vector and right-hand side. In this case, results from, for instance, [208] show that the optimal objective value is locally Lipschitz continuous. In the general case, the objective function of (2.2) is a convex piecewise affine function; consequently, it can be reformulated as the linear program

$$\begin{aligned} q(\mathbf{b}, \mathbf{c}_1, \ldots, \mathbf{c}_p, \mathbf{h}) = \min_{(\mathbf{z},s) \in \mathbb{R}^{n+1}} \; & s \\ \text{s.t. } & \mathbf{A}\mathbf{z} \le \mathbf{b}, \\ & \mathbf{c}_i^{\mathrm{T}}\mathbf{z} + h_i \le s, \, \forall i \in I. \end{aligned}$$

However, in this form, the parameterization is now influencing the constraints of the LP, which in general is less well behaved (see for instance [208]). Thus, rather than obscure the nice parametric properties just established, parametric optimization problems of the form (2.2) are kept in that form and loosely referred to as "linear programs."

2.5 Ordinary differential equations

Initial value problems in ordinary differential equations (IVPs in ODEs) will also be important in this thesis. For $n \in \mathbb{N}$, $D \subset \mathbb{R}^n$, $T \subset \mathbb{R}$, D nonempty, $T = [t_0, t_f] \neq \varnothing$, $\mathbf{x}_0 \in D$, and $\mathbf{f} : T \times D \to \mathbb{R}^n$, consider the IVP in ODEs

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad a.e.\ t \in T, \quad \mathbf{x}(t_0) = \mathbf{x}_0. \tag{2.3}$$

An important property of this ODE system relates to the Lipschitz continuity of its dynamics (or right-hand side) $\mathbf{f}$. The specific property is given in the following definition, borrowing terminology from Ch. 3 of [163].

Definition 2.5.1. Let (T, d_T), (X, d_X), and (Y, d_Y) be metric spaces, and let $f : T \times X \to Y$. Then f is locally Lipschitz continuous on X, uniformly on T, if for all $x \in X$ there exists a neighborhood $N(x)$ of x and $L(x) > 0$ such that

$$d_Y(f(t, x_1), f(t, x_2)) \le L(x) d_X(x_1, x_2)$$

for all $(t, x_1, x_2) \in T \times N(x) \times N(x)$.

In the context of IVP (2.3), we desire $\mathbf{f}$ to be locally Lipschitz continuous on D, uniformly on T. The uniqueness of the solutions of (2.3) rely on this property; see Theorem 1.10

of Ch. II of [116] or Theorem 2 in §1 of [51]. Further, a similar condition establishes that many numerical integration methods, including Runge-Kutta and linear multistep methods, are convergent for problem (2.3); see, for instance, Theorem 1.1 of §1.4 of [105], and Definition 1.6 of §II.1 and the convergence analyses in Sections III.3 and VII.4 of [116].

The next result shows that local Lipschitz continuity on a partially compact domain can yield the Lipschitz property in Definition 2.5.1.

Lemma 2.5.1. *Assume $(m, n, p) \in \mathbb{N}^3$. Let $C \subset \mathbb{R}^m$ be nonempty and compact and $D \subset \mathbb{R}^n$ be nonempty. Let $\mathbf{g} : C \times D \to \mathbb{R}^p$ be locally Lipschitz continuous. Then for all $\mathbf{z} \in D$, there exists a neighborhood $N(\mathbf{z})$ and $L > 0$ such that for all $(\mathbf{y}, \mathbf{z}_1, \mathbf{z}_2)$ in $C \times (N(\mathbf{z}) \cap D) \times (N(\mathbf{z}) \cap D)$*

$$\|\mathbf{g}(\mathbf{y}, \mathbf{z}_1) - \mathbf{g}(\mathbf{y}, \mathbf{z}_2)\| \leq L \|\mathbf{z}_1 - \mathbf{z}_2\| .$$

Proof. Choose $\mathbf{z} \in D$. For each $\mathbf{y} \in C$, let $N(\mathbf{y}, \mathbf{z})$ be a neighborhood of $(\mathbf{y}, \mathbf{z})$ such that $\mathbf{g}$ is Lipschitz continuous on $N(\mathbf{y}, \mathbf{z}) \cap (C \times D)$, with corresponding Lipschitz constant $L(\mathbf{y})$. However, this collection of open sets form an open cover of $C \times \{\mathbf{z}\}$, which is compact, and thus we can choose a finite number of these neighborhoods $\{N(\mathbf{y}_i, \mathbf{z}) : 1 \leq i \leq k\}$, such that their union, $\widetilde{N}$, contains $C \times \{\mathbf{z}\}$. Let L be the (finite) maximum of the corresponding Lipschitz constants (i.e. $L = \max\{L(\mathbf{y}_i) : 1 \leq i \leq k\}$). Note that $\widetilde{N}$ is an open set, and $\mathbf{g}$ is Lipschitz continuous on $\widetilde{N} \cap (C \times D)$ with Lipschitz constant L.

We claim that there exists a $\delta > 0$ such that $C \times N_\delta(\mathbf{z}) \subset \widetilde{N}$ (where $N_\delta(\mathbf{z})$ is viewed as a subset of $\mathbb{R}^n$). This follows from, for instance, Lemma 1 in §5 of [51]. The argument is that the complement of $\widetilde{N}$, $\widetilde{N}^C$, is closed and disjoint from $C \times \{\mathbf{z}\}$, and so there exists a $\delta > 0$ such that the distance between any point in $C \times \{\mathbf{z}\}$ and any point in $\widetilde{N}^C$ is greater than δ. This implies that $C \times N_\delta(\mathbf{z})$ is disjoint from $\widetilde{N}^C$, which in turn implies $C \times N_\delta(\mathbf{z}) \subset \widetilde{N}$. The result follows from Lipschitz continuity on $(C \times N_\delta(\mathbf{z})) \cap (C \times D) = C \times N_\delta(\mathbf{z}) \cap D$. □

Chapter 3

Parametric affine relaxations

3.1 Introduction

As motivation for the discussion in this chapter, consider the following problem: Given $h : \mathbb{R}^n \to \mathbb{R}$, we seek $(\mathbf{h}^{al}, \mathbf{h}^{au}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ and $(h^{bl}, h^{bu}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \times \mathbb{R}$ such that for any interval subset $[\mathbf{v}, \mathbf{w}]$ of $\mathbb{R}^n$, we have

$$(\mathbf{h}^{al}(\mathbf{v}, \mathbf{w}))^{\mathrm{T}}\mathbf{z} + h^{bl}(\mathbf{v}, \mathbf{w}) \leq h(\mathbf{z}) \leq (\mathbf{h}^{au}(\mathbf{v}, \mathbf{w}))^{\mathrm{T}}\mathbf{z} + h^{bu}(\mathbf{v}, \mathbf{w}),$$

for all $\mathbf{z} \in [\mathbf{v}, \mathbf{w}]$. Furthermore, we want $\mathbf{h}^{al}$, $\mathbf{h}^{au}$, h^{bl}, and h^{bu} to be at least continuous (in specific, locally Lipschitz continuous). In other words, we want affine relaxations of h that are continuously parameterized by the set on which these relaxations are valid. Such relaxations will be important in later chapters, specifically in the context of estimating the optimal objective values of parameterized optimization problems in a dynamic setting.

In general, a method is presented for constructing affine relaxations which are continuously parameterized by the underlying set and/or "seed" relaxations. This method has parallels to interval arithmetic [130], and requires the simultaneous calculation of interval bounds. In interval arithmetic, we have a "library" of basic arithmetic operations (such as addition) and elemental functions (such as the exponential or square root). For each of these functions in the library, interval bounds on the range of the function (given some input interval) are available. Then, we can obtain interval bounds on the range of any more complicated function that can be expressed as the finite composition of these library functions.

This affine relaxation method addresses some shortcomings in the literature. Affine relaxations can be obtained from first-order Taylor models [114], subgradients to convex and concave relaxations [124], or other types of "affine arithmetic" [41, 204]. These methods provide affine under and overestimators on some underlying set, usually an interval. However, none of these methods address the issue of how the affine relaxations ($\mathbf{h}^{al}$ and h^{bl} in the example above) depend on the underlying set. In the case of the method in [124], the subgradients are for McCormick relaxations which are potentially nonsmooth. Consequently, these subgradients have potentially discontinuous behavior as the interval set on which the relaxations are valid changes. Similarly, the affine relaxations from [204] do not have the required parametric regularity. Meanwhile, Taylor model arithmetic involves estimating the range of various polynomials. Even if the Taylor model is first-order, this may involve bounding second or higher order polynomials. This can be achieved in various ways (see §5.4.3 of [113] and [136]). However, the default method employed in the implementation of Taylor model arithmetic in MC++ [36], for instance, comes from [107], which involves different cases depending on certain data. As a result, the scalar shifts (h^{bl} and h^{bu}) of the affine relaxations may be discontinuous as the underlying set changes.

The affine arithmetic developed in [41, 190] is very close to what will be described here, but has different motivations and consequently does not address parametric regularity of the affine relaxations. As mentioned, the parametric affine relaxations will be used to estimate the optimal objective values of parameterized optimization problems. These estimates will define the dynamics of an initial value problem in ordinary differential equations; thus, the parametric affine relaxations developed here will need to be calculated many times over the course of numerical integration. Therefore, another aim in the development of the parametric affine relaxations of this chapter is that their calculation is as efficient as possible. Meanwhile, the affine arithmetic from [41, 190] requires the addition of an "error term" each time a nonlinear library function is encountered; the size of the underlying objects consequently increases over the course of the evaluation of the affine relaxations. In other words, although the initial uncertainty (the underlying interval over which relaxations are desired) might only be in two dimensions, the final affine relaxations are over a $2 + p$ dimensional interval, where p is a nonnegative integer. Obtaining relaxations on the original interval is possible, but more than anything this points out the different aims of the affine arithmetic from [41, 190], and thus developing a theory focused on parametric regularity is worthwhile.

3.2 Composition result

The basis of the theory is a composition result for constructing locally Lipschitz continuous affine relaxations of a function given locally Lipschitz continuous affine relaxations of its arguments. The following lemma will be useful; its proof is clear in light of Lemma 2.3.2 and the discussion that follows it.

Lemma 3.2.1. *Let X be a metric space. Let $\mathbf{c}_1$ and $\mathbf{c}_2$ be locally Lipschitz continuous mappings $X \to \mathbb{R}^m$, and s be a locally Lipschitz continuous mapping $X \to \mathbb{R}$. Define $\mathbf{c}_3 : X \to \mathbb{R}^m$ by*

$$\mathbf{c}_3 : \mathbf{x} \mapsto \max\{s(\mathbf{x}), 0\}\mathbf{c}_1(\mathbf{x}) + \min\{s(\mathbf{x}), 0\}\mathbf{c}_2(\mathbf{x}) = \begin{cases} s(\mathbf{x})\mathbf{c}_1(\mathbf{x}) & \textit{if } s(\mathbf{x}) \geq 0, \\ s(\mathbf{x})\mathbf{c}_2(\mathbf{x}) & \textit{otherwise.} \end{cases}$$

Then $\mathbf{c}_3$ is a locally Lipschitz continuous mapping on X.

The composition result follows.

Proposition 3.2.2. *Let $(m,n) \in \mathbb{N}^2$. Let X be a metric space, $Y \subset \mathbb{R}^m$, and $Z \subset \mathbb{R}^n$. Let $f : Y \to \mathbb{R}$ and $\mathbf{g} : Z \to \mathbb{R}^m$. Let $Z_D : X \rightrightarrows Z$. Let $Y_D \subset \{(\mathbf{v}, \mathbf{w}) \in Y \times Y : [\mathbf{v}, \mathbf{w}] \subset Y\}$. For $i \in \{1, \dots, m\}$, let $\mathbf{g}_i^{al}$ and $\mathbf{g}_i^{au}$ be locally Lipschitz continuous mappings $X \to \mathbb{R}^n$ and g_i^{bl}, g_i^{bu}, g_i^L, g_i^U be locally Lipschitz continuous mappings $X \to \mathbb{R}$ which for all $\mathbf{x} \in X$ satisfy*

$$\begin{gathered} \mathbf{g}_i^{al}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + g_i^{bl}(\mathbf{x}) \leq g_i(\mathbf{z}) \leq \mathbf{g}_i^{au}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + g_i^{bu}(\mathbf{x}), \forall \mathbf{z} \in Z_D(\mathbf{x}), \forall i, \\ \mathbf{g}^L(\mathbf{x}) \leq \mathbf{g}(\mathbf{z}) \leq \mathbf{g}^U(\mathbf{x}), \forall \mathbf{z} \in Z_D(\mathbf{x}), \\ (\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \in Y_D. \end{gathered}$$

Let $\mathbf{f}^{al}$ and $\mathbf{f}^{au}$ be locally Lipschitz continuous mappings $Y_D \to \mathbb{R}^m$ and f^{bl}, f^{bu}, f^L, and f^U be locally Lipschitz continuous mappings $Y_D \to \mathbb{R}$ which for all $(\mathbf{v}, \mathbf{w}) \in Y_D$ satisfy

$$\begin{gathered} \mathbf{f}^{al}(\mathbf{v}, \mathbf{w})^{\mathrm{T}}\mathbf{y} + f^{bl}(\mathbf{v}, \mathbf{w}) \leq f(\mathbf{y}) \leq \mathbf{f}^{au}(\mathbf{v}, \mathbf{w})^{\mathrm{T}}\mathbf{y} + f^{bu}(\mathbf{v}, \mathbf{w}), \\ f^L(\mathbf{v}, \mathbf{w}) \leq f(\mathbf{y}) \leq f^U(\mathbf{v}, \mathbf{w}), \end{gathered}$$

for all $\mathbf{y} \in [\mathbf{v}, \mathbf{w}]$.

Let $h : Z \to \mathbb{R}$ *be defined by* $h : \mathbf{z} \mapsto f(\mathbf{g}(\mathbf{z}))$. *For* $i \in \{1, \ldots, m\}$, *let*

$$\mathbf{h}_i^{al} : \mathbf{x} \mapsto \begin{cases} f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))\mathbf{g}_i^{al}(\mathbf{x}) & \textit{if } f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \geq 0, \\ f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))\mathbf{g}_i^{au}(\mathbf{x}) & \textit{otherwise}, \end{cases}$$

$$h_i^{bl} : \mathbf{x} \mapsto \begin{cases} f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))g_i^{bl}(\mathbf{x}) & \textit{if } f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \geq 0, \\ f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))g_i^{bu}(\mathbf{x}) & \textit{otherwise}, \end{cases}$$

$$\mathbf{h}_i^{au} : \mathbf{x} \mapsto \begin{cases} f_i^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))\mathbf{g}_i^{au}(\mathbf{x}) & \textit{if } f_i^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \geq 0, \\ f_i^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))\mathbf{g}_i^{al}(\mathbf{x}) & \textit{otherwise}, \end{cases}$$

$$h_i^{bu} : \mathbf{x} \mapsto \begin{cases} f_i^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))g_i^{bu}(\mathbf{x}) & \textit{if } f_i^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \geq 0, \\ f_i^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))g_i^{bl}(\mathbf{x}) & \textit{otherwise}. \end{cases}$$

Let $\mathbf{h}^{al}$, $\mathbf{h}^{au} : X \to \mathbb{R}^n$ *and* h^{bl}, $h^{bu} : X \to \mathbb{R}$ *be defined by*

$$\begin{aligned} \mathbf{h}^{al} &: \mathbf{x} \mapsto \sum_i \mathbf{h}_i^{al}(\mathbf{x}), & h^{bl} &: \mathbf{x} \mapsto f^{bl}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) + \sum_i h_i^{bl}(\mathbf{x}), \\ \mathbf{h}^{au} &: \mathbf{x} \mapsto \sum_i \mathbf{h}_i^{au}(\mathbf{x}), & h^{bu} &: \mathbf{x} \mapsto f^{bu}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) + \sum_i h_i^{bu}(\mathbf{x}). \end{aligned}$$

Let $h^L : X \ni \mathbf{x} \mapsto f^L(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))$ *and* $h^U : X \ni \mathbf{x} \mapsto f^U(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))$. *Then* $\mathbf{h}^{al}$, $\mathbf{h}^{au}$, h^{bl}, h^{bu}, h^L, h^U *are locally Lipschitz continuous mappings on* X *which for all* $\mathbf{x} \in X$ *satisfy*

$$\begin{gathered} \mathbf{h}^{al}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + h^{bl}(\mathbf{x}) \leq h(\mathbf{z}) \leq \mathbf{h}^{au}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + h^{bu}(\mathbf{x}), \\ h^L(\mathbf{x}) \leq h(\mathbf{z}) \leq h^U(\mathbf{x}), \end{gathered}$$

for all $\mathbf{z} \in Z_D(\mathbf{x})$.

Proof. Local Lipschitz continuity is established first. By assumption, for each $\mathbf{x} \in X$, $(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \in Y_D$. By the local Lipschitz continuity of the composition of functions (Lemma 2.3.2), $\mathbf{x} \mapsto f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))$ and $\mathbf{x} \mapsto f_i^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))$ are locally Lipschitz continuous functions on X, for each i. Similarly, h^L, h^U, $\mathbf{x} \mapsto f^{bl}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))$ and $\mathbf{x} \mapsto f^{bu}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))$ are locally Lipschitz continuous on X. Then by Lemma 3.2.1, $\mathbf{h}_i^{al}$, $\mathbf{h}_i^{au}$, h_i^{bl}, and h_i^{bu} are locally Lipschitz continuous for each i. Finally, noting that the sum of locally Lipschitz continuous functions is locally Lipschitz continuous, we have that $\mathbf{h}^{al}$, $\mathbf{h}^{au}$, h^{bl}, h^{bu} (and h^L and h^U) are locally Lipschitz continuous on X.

Next, the lower and upper estimation properties are established. Choose any $\mathbf{x} \in X$. By assumption, $(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \in Y_D$, and since $\mathbf{g}(\mathbf{z}) \in [\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})]$ for any $\mathbf{z} \in Z_D(\mathbf{x})$, we have

$$\begin{aligned} \mathbf{f}^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))^{\mathrm{T}}\mathbf{g}(\mathbf{z}) + f^{bl}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) &\leq f(\mathbf{g}(\mathbf{z})), \\ \mathbf{f}^{au}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))^{\mathrm{T}}\mathbf{g}(\mathbf{z}) + f^{bu}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) &\geq f(\mathbf{g}(\mathbf{z})), \end{aligned} \tag{3.1}$$

for any $\mathbf{z} \in Z_D(\mathbf{x})$. Consider each term in the inner products. If $f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \geq 0$, for instance, then we have

$$f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \left(\mathbf{g}_i^{al}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + g_i^{bl}(\mathbf{x})\right) \leq f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))g_i(\mathbf{z}),$$

and otherwise

$$f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \left(\mathbf{g}_i^{au}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + g_i^{bu}(\mathbf{x})\right) \leq f_i^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))g_i(\mathbf{z}).$$

Applying the definitions of $\mathbf{h}_i^{al}$ and h_i^{bl}, we have

$$\begin{aligned} \left(\sum_i \mathbf{h}_i^{al}(\mathbf{x})\right)^{\mathrm{T}} \mathbf{z} + \sum_i \left(h_i^{bl}(\mathbf{x})\right) + f^{bl}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})) \leq \\ \mathbf{f}^{al}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x}))^{\mathrm{T}}\mathbf{g}(\mathbf{z}) + f^{bl}(\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})), \end{aligned}$$

which, combined with Inequality (3.1), and using the definitions of $\mathbf{h}^{al}$ and h^{bl} establishes

$$\mathbf{h}^{al}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + h^{bl}(\mathbf{x}) \leq f(\mathbf{g}(\mathbf{z})) = h(\mathbf{z}),$$

for all $\mathbf{z} \in Z_D(\mathbf{x})$. Similar reasoning establishes the case for the affine overestimator $(\mathbf{h}^{au}(\mathbf{x}), h^{bu}(\mathbf{x}))$ and the interval bounds $h^L(\mathbf{x})$ and $h^U(\mathbf{x})$. □

Note that the hypotheses of Proposition 3.2.2 do not preclude the possibility that $Z_D(\mathbf{x})$ is empty for some $\mathbf{x} \in X$. In this case, we allow $[\mathbf{g}^L(\mathbf{x}), \mathbf{g}^U(\mathbf{x})]$ to be empty, and it is for this

Table 3.1: Library functions f, their domain Y, and the domain Y_D of their parametric interval and affine relaxations.

f	Domain Y	Domain Y_D
$s \in \mathbb{R}$, $y \mapsto s$	$\mathbb{R}$	$\mathbb{R}^2$
$s \in \mathbb{R}$, $y \mapsto sy$	$\mathbb{R}$	$\mathbb{R}^2$
$s \in \mathbb{R}$, $y \mapsto y + s$	$\mathbb{R}$	$\mathbb{R}^2$
$(y_1, y_2) \mapsto y_1 + y_2$	$\mathbb{R}^2$	$\mathbb{R}^4$
$(y_1, y_2) \mapsto y_1 - y_2$	$\mathbb{R}^2$	$\mathbb{R}^4$
$(y_1, y_2) \mapsto y_1 y_2$	$\mathbb{R}^2$	$\mathbb{R}^4$
$y \mapsto y^2$	$\mathbb{R}$	$\mathbb{R}^2$
$y \mapsto \lvert y \rvert$	$\mathbb{R}$	$\{(v, w) \in \mathbb{R}^2 : v \neq w\}$
$y \mapsto \exp(y)$	$\mathbb{R}$	$\{(v, w) \in \mathbb{R}^2 : v \neq w\}$
$y \mapsto \ln(y)$	$\{y \in \mathbb{R} : y > 0\}$	$\{(v, w) \in \mathbb{R}^2 : v > 0, w > 0, v \neq w\}$
$y \mapsto \sqrt{y}$	$\{y \in \mathbb{R} : y \geq 0\}$	$\{(v, w) \in \mathbb{R}^2 : v > 0, w > 0\}$
$y \mapsto {}^1\!/_y$	$\{y \in \mathbb{R} : y \neq 0\}$	$\{\mathbf{v} \in \mathbb{R}^2 : \mathbf{v} > \mathbf{0}\} \cup \{\mathbf{v} \in \mathbb{R}^2 : \mathbf{v} < \mathbf{0}\}$

reason that Y_D is defined to let $[\mathbf{v}, \mathbf{w}] = \varnothing$ for some $(\mathbf{v}, \mathbf{w}) \in Y_D$. In this case the conditions

$$\mathbf{f}^{al}(\mathbf{v}, \mathbf{w})^{\mathrm{T}}\mathbf{y} + f^{bl}(\mathbf{v}, \mathbf{w}) \leq f(\mathbf{y}) \leq \mathbf{f}^{au}(\mathbf{v}, \mathbf{w})^{\mathrm{T}}\mathbf{y} + f^{bu}(\mathbf{v}, \mathbf{w}),$$
$$f^L(\mathbf{v}, \mathbf{w}) \leq f(\mathbf{y}) \leq f^U(\mathbf{v}, \mathbf{w}),$$

for all $\mathbf{y} \in [\mathbf{v}, \mathbf{w}]$, are trivially true.

3.3 Function library

In order to apply Proposition 3.2.2 in practice, locally Lipschitz continuous interval and affine relaxations of various functions f comprising a library are required. First, as required by Proposition 3.2.2, the parametric interval and affine relaxations for a given function require a common domain Y_D. These are summarized in Table 3.1. The interval bounds are summarized in Table 3.2. These are standard and come from interval analysis. Parametric affine under and overestimators are summarized in Table 3.3. The reasoning behind these choices and other discussion are in the following subsections.

3.3.1 Simple arithmetic functions

The constant mapping, scalar multiplication, addition of a constant, bivariate addition, and bivariate subtraction all have straightforward affine relaxations that do not depend on the

Table 3.2: Library functions f and their parameterized interval relaxations on $[\mathbf{y}^L, \mathbf{y}^U]$.

f	$f^L(\mathbf{y}^L, \mathbf{y}^U)$	$f^U(\mathbf{y}^L, \mathbf{y}^U)$
$s \in \mathbb{R}$, $y \mapsto s$	s	s
$s \in \mathbb{R}$, $y \mapsto sy$	$\min\{sy^L, sy^U\}$	$\max\{sy^L, sy^U\}$
$s \in \mathbb{R}$, $y \mapsto y + s$	$y^L + s$	$y^U + s$
$(y_1, y_2) \mapsto y_1 + y_2$	$y_1^L + y_2^L$	$y_1^U + y_2^U$
$(y_1, y_2) \mapsto y_1 - y_2$	$y_1^L - y_2^U$	$y_1^U - y_2^L$
$(y_1, y_2) \mapsto y_1 y_2$	$\min\{y_1^L y_2^L, y_1^L y_2^U, y_1^U y_2^L, y_1^U y_2^U\}$	$\max\{y_1^L y_2^L, y_1^L y_2^U, y_1^U y_2^L, y_1^U y_2^U\}$
$y \mapsto y^2$	$(\text{median}\,\{y^L, y^U, 0\})^2$	$\max\{(y^L)^2, (y^U)^2\}$
$y \mapsto \lvert y \rvert$	$\left\lvert \text{median}\,\{y^L, y^U, 0\} \right\rvert$	$\max\{\lvert y^L \rvert, \lvert y^U \rvert\}$
$y \mapsto \exp(y)$	$\exp(y^L)$	$\exp(y^U)$
$y \mapsto \ln(y)$	$\ln(y^L)$	$\ln(y^U)$
$y \mapsto \sqrt{y}$	$\sqrt{y^L}$	$\sqrt{y^U}$
$y \mapsto {}^1/_y$	${}^1/_{y^U}$	${}^1/_{y^L}$

underlying interval.

3.3.2 Bivariate multiplication

The specific expression for the affine relaxations of the bilinear function can be obtained in a few ways. First, a second-order Taylor expansion at some reference $\mathbf{y}^r$ is

$$\begin{aligned} y_1 y_2 &= y_1^r y_2^r + \begin{bmatrix} y_2^r \\ y_1^r \end{bmatrix}^{\mathrm{T}} (\mathbf{y} - \mathbf{y}^r) + {}^1/_2 (\mathbf{y} - \mathbf{y}^r)^{\mathrm{T}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} (\mathbf{y} - \mathbf{y}^r) \\ &= y_1^r y_2^r + y_2^r (y_1 - y_1^r) + y_1^r (y_2 - y_2^r) + (y_1 - y_1^r)(y_2 - y_2^r), \end{aligned}$$

which one can verify is exact. Letting $\mathbf{y}^r$ equal the midpoint of the interval $[\mathbf{y}^L, \mathbf{y}^U]$ and evaluating the second-order term in interval arithmetic yields the affine relaxations in Table 3.3. More generally, the expressions obtained, seen as defining functions of $(y_1^L, y_2^L, y_1^U, y_2^U)$, are indeed locally Lipschitz continuous on all of $\mathbb{R}^4$, and constitute valid affine under and overestimators when $\mathbf{y}^L \leq \mathbf{y}^U$.

Alternatively, we could obtain the affine underestimator in Table 3.3 from the pointwise average of the two affine functions comprising the convex envelope of the bilinear term on $[\mathbf{y}^L, \mathbf{y}^U]$:

$$\max\{y_2^L y_1 + y_1^L y_2 - y_1^L y_2^L, y_2^U y_1 + y_1^U y_2 - y_1^U y_2^U\}.$$

Similarly the affine overestimator comes from the pointwise average of the two affine func-

Table 3.3: Library functions f and their parameterized affine under and overestimators on $[\mathbf{y}^L, \mathbf{y}^U]$. Define $m : (y^L, y^U) \mapsto \frac{y^L+y^U}{2}$.

Underestimators:

f	$\mathbf{f}^{al}(\mathbf{y}^L, \mathbf{y}^U)$	$f^{bl}(\mathbf{y}^L, \mathbf{y}^U)$
$s \in \mathbb{R}$, $y \mapsto s$	0	s
$s \in \mathbb{R}$, $y \mapsto sy$	s	0
$s \in \mathbb{R}$, $y \mapsto y + s$	1	s
$(y_1, y_2) \mapsto y_1 + y_2$	$(1, 1)$	0
$(y_1, y_2) \mapsto y_1 - y_2$	$(1, -1)$	0
$(y_1, y_2) \mapsto y_1 y_2$	$\left(m(y_2^L, y_2^U), m(y_1^L, y_1^U)\right)$	$-{}^1\!/_2(y_1^L y_2^L + y_1^U y_2^U)$
$y \mapsto y^2$	$2m(y^L, y^U)$	$-m(y^L, y^U)^2$
$y \mapsto \lvert y \rvert$	$\frac{\lvert y^U \rvert - \lvert y^L \rvert}{y^U - y^L}$	0
$y \mapsto \exp(y)$	$\exp(m(y^L, y^U))$	$\exp(m(y^L, y^U))(1 - m(y^L, y^U))$
$y \mapsto \ln(y)$	$\frac{\ln(y^U) - \ln(y^L)}{y^U - y^L}$	$-\frac{\ln(y^U) - \ln(y^L)}{y^U - y^L} y^L + \ln(y^L)$
$y \mapsto \sqrt{y}$	$\left(\sqrt{y^L} + \sqrt{y^U}\right)^{-1}$	$-\left(\sqrt{y^L} + \sqrt{y^U}\right)^{-1} y^L + \sqrt{y^L}$
$y \mapsto {}^1\!/_y$	$\begin{cases} -(m(y^L, y^U)^{-2}) & \text{if } (y^L, y^U) > \mathbf{0} \\ -(y^L y^U)^{-1} & \text{if } (y^L, y^U) < \mathbf{0} \end{cases}$	$\begin{cases} 2m(y^L, y^U)^{-1} & \text{if } (y^L, y^U) > \mathbf{0} \\ {}^1\!/_{y^L} + {}^1\!/_{y^U} & \text{if } (y^L, y^U) < \mathbf{0} \end{cases}$

Overestimators:

f	$\mathbf{f}^{au}(\mathbf{y}^L, \mathbf{y}^U)$	$f^{bu}(\mathbf{y}^L, \mathbf{y}^U)$
$s \in \mathbb{R}$, $y \mapsto s$	0	s
$s \in \mathbb{R}$, $y \mapsto sy$	s	0
$s \in \mathbb{R}$, $y \mapsto y + s$	1	s
$(y_1, y_2) \mapsto y_1 + y_2$	$(1, 1)$	0
$(y_1, y_2) \mapsto y_1 - y_2$	$(1, -1)$	0
$(y_1, y_2) \mapsto y_1 y_2$	$\left(m(y_2^L, y_2^U), m(y_1^L, y_1^U)\right)$	$-{}^1\!/_2(y_1^U y_2^L + y_1^L y_2^U)$
$y \mapsto y^2$	$y^L + y^U$	$-y^L y^U$
$y \mapsto \lvert y \rvert$	$\frac{\lvert y^U \rvert - \lvert y^L \rvert}{y^U - y^L}$	$-\frac{\lvert y^U \rvert - \lvert y^L \rvert}{y^U - y^L} y^L + \lvert y^L \rvert$
$y \mapsto \exp(y)$	$\frac{\exp(y^U) - \exp(y^L)}{y^U - y^L}$	$-\frac{\exp(y^U) - \exp(y^L)}{y^U - y^L} y^L + \exp(y^L)$
$y \mapsto \ln(y)$	$m(y^L, y^U)^{-1}$	$\ln(m(y^L, y^U)) - 1$
$y \mapsto \sqrt{y}$	$\left(2\sqrt{m(y^L, y^U)}\right)^{-1}$	${}^1\!/_2\sqrt{m(y^L, y^U)}$
$y \mapsto {}^1\!/_y$	$\begin{cases} -(y^L y^U)^{-1} & \text{if } (y^L, y^U) > \mathbf{0} \\ -(m(y^L, y^U)^{-2}) & \text{if } (y^L, y^U) < \mathbf{0} \end{cases}$	$\begin{cases} {}^1\!/_{y^L} + {}^1\!/_{y^U} & \text{if } (y^L, y^U) > \mathbf{0} \\ 2m(y^L, y^U)^{-1} & \text{if } (y^L, y^U) < \mathbf{0} \end{cases}$

tions comprising the concave envelope on $[\mathbf{y}^L, \mathbf{y}^U]$:

$$\min\{y_2^L y_1 + y_1^U y_2 - y_1^U y_2^L, y_2^U y_1 + y_1^L y_2 - y_1^L y_2^U\}.$$

3.3.3 Square

As with a number of functions in the library, there are various choices for defining the affine relaxations. The motivation behind the relaxations for the "square" function $y \mapsto y^2$ holds for a number of univariate library functions. Since the square function is a convex function, an underestimator comes from a tangent at any point in $[y^L, y^U]$. We might be tempted to choose a linearization point so that the minimum of the affine underestimator coincides with the minimum of $y \mapsto y^2$ on $[y^L, y^U]$. However, the parametric affine relaxations are constructed simultaneously with interval bounds, and the interval bounds are already capturing the exact minimum and maximum of the square function on the interval of interest. Thus, the affine relaxations should complement the interval arithmetic and provide "good" first-order information. In general one would want to use the Chebyshev affine approximation (minimizing the maximum error between the function and the affine underestimator, see §3.3 of [41]). As stated in §3.1, the motivation for these affine relaxations is first and foremost parametric regularity, and simplicity and speed of calculation.

Along these lines, the linearization point for the square function and others is chosen as the midpoint of the interval. Thus, the underestimator for the square function is

$$2\Big(\frac{y^L + y^U}{2}\Big)\Big(y - \frac{y^L + y^U}{2}\Big) + \Big(\frac{y^L + y^U}{2}\Big)^2.$$

which simplifies. Meanwhile, an overestimator comes from the secant over the interval $[y^L, y^U]$,

$$\frac{(y^U)^2 - (y^L)^2}{y^U - y^L}(y - y^L) + (y^L)^2,$$

which simplifies.

3.3.4 Absolute value

Similar to the case of the square function, the overestimator for the absolute value function comes from the secant over the underlying interval; the underestimator is the line parallel to the overestimator with zero intercept.

Note that the domain Y_D in Table 3.1 precludes a degenerate interval. In part, this is because the expression for $\mathbf{f}^{al}$ and $\mathbf{f}^{au}$ would become singular. We might try to define an extension of $\mathbf{f}^{al}$, for instance, by defining

$$\mathbf{f}^{al} : (y^L, y^U) \mapsto \begin{cases} 1 & \text{if } y^L = y^U \geq 0, \\ -1 & \text{if } y^L = y^U < 0. \end{cases}$$

However, note that this is not continuous.

3.3.5 Exponential and natural logarithm

The underestimator of the exponential function and overestimator of the natural logarithm are tangents at the midpoint of the interval; the overestimator of the exponential and underestimator of the natural logarithm are secants over the interval. As in the case of the absolute value function, the expressions for these secants become singular for degenerate intervals. However, since these functions are smooth, it is likely that there are locally Lipschitz continuous extensions of the affine overestimator (the mappings $\mathbf{f}^{au}$ and f^{bu}) for the exponential function, for instance, to all of $\mathbb{R}^2$. This is a subject for future research.

3.3.6 Square root

As in the case of the natural logarithm, the underestimator is a secant while the overestimator is a tangent at the midpoint of the interval; these expressions simplify to those in Table 3.3. While the domain Y of the square root function is the nonnegative reals, it is only locally Lipschitz continuous on the positive reals. Thus, the domain Y_D of the interval and affine relaxations must be restricted to the (strictly) positive orthant of $\mathbb{R}^2$ to be locally Lipschitz continuous.

3.3.7 Reciprocal

On interval subsets of the positive reals, the affine underestimator of the reciprocal function $y \mapsto {}^1\!/_y$ is a tangent at the midpoint of the interval; the affine overestimator is a secant. On interval subsets of the negative reals, this is reversed; the affine under and overestimator are a secant and tangent at the midpoint, respectively. Although the affine relaxations involve different "branches," the domain Y_D is disconnected and so the relaxations are locally

Lipschitz continuous.

3.3.8 Trigonometric functions, integer powers

Trig functions and odd integer powers (e.g. $y \mapsto \sin(y)$, $y \mapsto y^3$) are subjects for future research. Even integer powers could be handled in a manner similar to the square function.

3.4 Examples

Constructing affine relaxations proceeds by recursively applying Proposition 3.2.2, illustrated by the following examples.

Example 3.4.1. Let $h : \mathbb{R} \times \mathbb{R}^2 \ni (p, \mathbf{z}) \mapsto pz_1(1 - z_2)$, where p is an uncertain parameter. The goal is to construct, for any $p \in [p^L, p^U]$ such that $p^L > 0$, affine relaxations of $h(p, \cdot)$ on any interval $[\mathbf{z}^L, \mathbf{z}^U]$ such that $z_1^L > 0$ and $z_2^L > 1$. Furthermore, one desires that these relaxations are locally Lipschitz continuous with respect to $(\mathbf{z}^L, \mathbf{z}^U)$. In terms of the notation in Proposition 3.2.2, we let $X = \mathbb{R}^2 \times \mathbb{R}^2$, $Z = \mathbb{R}^2$, and $Z_D : (\mathbf{z}^L, \mathbf{z}^U) \mapsto [\mathbf{z}^L, \mathbf{z}^U]$.

As mentioned, the process resembles the construction of an interval enclosure of the range of h via interval arithmetic. Evaluation of h is broken down into a sequence of auxiliary variables called "factors," which can be expressed as simple arithmetic operations on previously computed factors. Interval and affine relaxations of each factor can also be computed, and following the rules in Proposition 3.2.2 and Tables 3.2 and 3.3, the affine relaxations will also be locally Lipschitz continuous in the manner desired. See Table 3.4 for the factored expression. Note that factor v_3, corresponding to the parameter p, is initialized with the trivial affine relaxations $\mathbf{0}^{\mathrm{T}}\mathbf{z} + p^L \le p \le \mathbf{0}^{\mathrm{T}}\mathbf{z} + p^U$. This ensures that the final relaxations obtained are valid for all $p \in [p^L, p^U]$, since in addition the calculation of the interval and affine relaxations do not depend on the value of p. Also, note that the restrictions $z_1^L > 0$, $z_2^L > 1$, and $p^L > 0$, simplify the evaluation and preclude the need to consider the different branches when constructing the affine relaxations for factors v_5 and v_6, as indicated in Proposition 3.2.2 (for example, this implies that $1/2(v_4^L + v_4^U) < 0$). Although in general, the different cases must be taken into account.

The final factor, v_6, gives the value of $h(p, \cdot)$, and thus one also has for any $p \in [p^L, p^U]$

$$(\mathbf{v}_6^{al})^{\mathrm{T}}\mathbf{z} + v_6^{bl} \le h(p, \mathbf{z}) \le (\mathbf{v}_6^{au})^{\mathrm{T}}\mathbf{z} + v_6^{bu}$$

Table 3.4: Factored expression, corresponding interval enclosures, and corresponding affine relaxations for Example 3.4.1.

Factor	Value	Lower bound	Upper Bound
v_1	z_1	$v_1^L = z_1^L$	$v_1^U = z_1^U$
v_2	z_2	$v_2^L = z_2^L$	$v_2^U = z_2^U$
v_3	p	$v_3^L = p^L$	$v_3^U = p^U$
v_4	$1 - v_2$	$v_4^L = 1 - v_2^U$	$v_4^U = 1 - v_2^L$
v_5	$v_1 v_4$	$v_5^L =$ $\min\{v_1^L v_4^L, v_1^L v_4^U, v_1^U v_4^L, v_1^U v_4^U\}$	$v_5^U =$ $\max\{v_1^L v_4^L, v_1^L v_4^U, v_1^U v_4^L, v_1^U v_4^U\}$
v_6	$v_3 v_5$	$v_6^L =$ $\min\{v_3^L v_5^L, v_3^L v_5^U, v_3^U v_5^L, v_3^U v_5^U\}$	$v_6^U =$ $\max\{v_3^L v_5^L, v_3^L v_5^U, v_3^U v_5^L, v_3^U v_5^U\}$

Factor	Underestimator	Overestimator
v_1	$\mathbf{v}_1^{al} = (1, 0)$, $v_1^{bl} = 0$	$\mathbf{v}_1^{au} = (1, 0)$, $v_1^{bu} = 0$
v_2	$\mathbf{v}_2^{al} = (0, 1)$, $v_2^{bl} = 0$	$\mathbf{v}_2^{au} = (0, 1)$, $v_2^{bu} = 0$
v_3	$\mathbf{v}_3^{al} = (0, 0)$, $v_3^{bl} = p^L$	$\mathbf{v}_3^{au} = (0, 0)$, $v_3^{bu} = p^U$
v_4	$\mathbf{v}_4^{al} = -\mathbf{v}_2^{au}$, $v_4^{bl} = 1 - v_2^{bu}$	$\mathbf{v}_4^{au} = -\mathbf{v}_2^{al}$, $v_4^{bu} = 1 - v_2^{bl}$
v_5	$\mathbf{v}_5^{al} = {}^1/_2(v_4^L + v_4^U)\mathbf{v}_1^{au}$ $+{}^1/_2(v_1^L + v_1^U)\mathbf{v}_4^{al}$, $v_5^{bl} = -{}^1/_2(v_1^L v_4^L + v_1^U v_4^U)$ $+{}^1/_2(v_4^L + v_4^U)v_1^{bu}$ $+{}^1/_2(v_1^L + v_1^U)v_4^{bl}$	$\mathbf{v}_5^{au} = {}^1/_2(v_4^L + v_4^U)\mathbf{v}_1^{al}$ $+{}^1/_2(v_1^L + v_1^U)\mathbf{v}_4^{au}$, $v_5^{bu} = -{}^1/_2(v_1^U v_4^L + v_1^L v_4^U)$ $+{}^1/_2(v_4^L + v_4^U)v_1^{bl}$ $+{}^1/_2(v_1^L + v_1^U)v_4^{bu}$
v_6	$\mathbf{v}_6^{al} = {}^1/_2(v_5^L + v_5^U)\mathbf{v}_3^{au}$ $+{}^1/_2(v_3^L + v_3^U)\mathbf{v}_5^{al}$, $v_6^{bl} = -{}^1/_2(v_3^L v_5^L + v_3^U v_5^U)$ $+{}^1/_2(v_5^L + v_5^U)v_3^{bu}$ $+{}^1/_2(v_3^L + v_3^U)v_5^{bl}$	$\mathbf{v}_6^{au} = {}^1/_2(v_5^L + v_5^U)\mathbf{v}_3^{al}$ $+{}^1/_2(v_3^L + v_3^U)\mathbf{v}_5^{au}$, $v_6^{bu} = -{}^1/_2(v_3^U v_5^L + v_3^L v_5^U)$ $+{}^1/_2(v_5^L + v_5^U)v_3^{bl}$ $+{}^1/_2(v_3^L + v_3^U)v_5^{bu}$

for all $\mathbf{z} \in [\mathbf{z}^L, \mathbf{z}^U]$. However, by virtue of Proposition 3.2.2, $\mathbf{v}_6^{al}$, $\mathbf{v}_6^{au}$, v_6^{bl}, v_6^{bu} can be considered locally Lipschitz continuous functions with respect to $(\mathbf{z}^L, \mathbf{z}^U)$.

Example 3.4.2. This example demonstrates the proper application of Proposition 3.2.2 when constructing relaxations that are locally Lipschitz continuous with respect to the seed or initial relaxations. Consider trying to construct affine relaxations of $\widetilde{h} : \mathbb{R}^2 \supset P \to \mathbb{R} : \mathbf{p} \mapsto p_1 + p_2 + \widetilde{z}(\mathbf{p})$, where $P = [\mathbf{p}^L, \mathbf{p}^U]$ is a nonempty set and $\widetilde{z} : P \to \mathbb{R}$, where all we know is that

$$(\mathbf{a}^l)^{\mathrm{T}}\mathbf{p} + b^l \leq \widetilde{z}(\mathbf{p}) \leq (\mathbf{a}^u)^{\mathrm{T}}\mathbf{p} + b^u,$$

$$z^L \leq \widetilde{z}(\mathbf{p}) \leq z^U,$$

for all $\mathbf{p} \in P$, for some $(\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$. Construction of affine relaxations of $\widetilde{h}$ on P is straightforward; let

$$\begin{aligned}
\widetilde{\mathbf{h}}^{al} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (1,0) + (0,1) + \mathbf{a}^l, \\
\widetilde{\mathbf{h}}^{au} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (1,0) + (0,1) + \mathbf{a}^u, \\
\widetilde{h}^{bl} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto b^l, \\
\widetilde{h}^{bu} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto b^u, \\
\widetilde{h}^{L} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto p_1^L + p_2^L + z^L, \\
\widetilde{h}^{U} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto p_1^U + p_2^U + z^U.
\end{aligned}$$

Furthermore, it is clear that these are locally Lipschitz continuous mappings.

However, for more complex expressions (and for automation of the constructions) we must use Proposition 3.2.2, and defining Z_D in this case is difficult. For instance, let $X = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$, $\widetilde{Z}_D : \mathbf{x} \mapsto P$, $g_1 : \mathbf{p} \mapsto \widetilde{z}(\mathbf{p})$, $g_1^L : (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto z^L$, and $g_1^U : (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto z^U$. Then the hypotheses of Proposition 3.2.2 cannot be satisfied; for $\mathbf{x} = (\mathbf{0}, \mathbf{0}, 0, 0, 2, 1) \in X$ it does *not* hold that

$$g_1^L(\mathbf{x}) \leq g_1(\mathbf{p}) \leq g_1^U(\mathbf{x}),$$

for all $\mathbf{p} \in \widetilde{Z}_D(\mathbf{x})$.

Proper application of Proposition 3.2.2 proceeds by letting $X = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ as before, but letting $Z = \mathbb{R} \times P$ and

$$Z_D : (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto \{(z, \mathbf{p}) \in [z^L, z^U] \times P : (\mathbf{a}^l)^{\mathrm{T}}\mathbf{p} + b^l \leq z \leq (\mathbf{a}^u)^{\mathrm{T}}\mathbf{p} + b^u\},$$

and instead constructing affine relaxations of $h : (z, \mathbf{p}) \mapsto p_1 + p_2 + z$. In this case, we can apply Proposition 3.2.2 once, by letting $\mathbf{e}_2 = (0,1,0)$, $\mathbf{e}_3 = (0,0,1)$,

$$\begin{aligned}
\mathbf{g} &: (z, \mathbf{p}) \mapsto (p_1, p_2, z) \\
(\mathbf{g}_1^{al}, \mathbf{g}_2^{al}, \mathbf{g}_3^{al}) &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto \big(\mathbf{e}_2, \mathbf{e}_3, (0, a_1^l, a_2^l)\big), \\
(\mathbf{g}_1^{au}, \mathbf{g}_2^{au}, \mathbf{g}_3^{au}) &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto \big(\mathbf{e}_2, \mathbf{e}_3, (0, a_1^u, a_2^u)\big), \\
(g_1^{bl}, g_2^{bl}, g_3^{bl}) &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (0, 0, b^l), \\
(g_1^{bu}, g_2^{bu}, g_3^{bu}) &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (0, 0, b^u), \\
(g_1^{L}, g_2^{L}, g_3^{L}) &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (p_1^L, p_2^L, z^L), \\
(g_1^{U}, g_2^{U}, g_3^{U}) &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (p_1^U, p_2^U, z^U),
\end{aligned}$$

and letting $Y = \mathbb{R}^3$, $Y_D = \mathbb{R}^6$,

$$\begin{aligned}
f &: (y_1, y_2, y_3) \mapsto y_1 + y_2 + y_3, \\
(\mathbf{f}^{al}, \mathbf{f}^{au}) &: (\mathbf{v}, \mathbf{w}) \mapsto \big((1,1,1), (1,1,1)\big), \\
(f^{bl}, f^{bu}) &: (\mathbf{v}, \mathbf{w}) \mapsto (0, 0), \\
(f^L, f^U) &: (\mathbf{v}, \mathbf{w}) \mapsto (v_1 + v_2 + v_3, w_1 + w_2 + w_3).
\end{aligned}$$

The hypotheses hold (perhaps trivially) since, for instance, $Z_D(\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) = \varnothing$ when $z^U < z^L$. The result is that we get

$$\begin{aligned}
\mathbf{h}^{al} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (0, 1 + a_1^l, 1 + a_2^l), \\
\mathbf{h}^{au} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto (0, 1 + a_1^u, 1 + a_2^u), \\
h^{bl} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto b^l, \\
h^{bu} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto b^u, \\
h^{L} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto p_1^L + p_2^L + z^L, \\
h^{U} &: (\mathbf{a}^l, \mathbf{a}^u, b^l, b^u, z^L, z^U) \mapsto p_1^U + p_2^U + z^U,
\end{aligned}$$

which satisfy the conclusion of Proposition 3.2.2. Since the first component of $\mathbf{h}^{al}$ and $\mathbf{h}^{au}$

(corresponding to "z") is zero, the conclusion of Proposition 3.2.2 reduces to: for all $\mathbf{x} \in X$,

$$\widetilde{\mathbf{h}}^{al}(\mathbf{x})^{\mathrm{T}}\mathbf{p} + \widetilde{h}^{bl}(\mathbf{x}) \leq h(z, \mathbf{p}) \leq \widetilde{\mathbf{h}}^{au}(\mathbf{x})^{\mathrm{T}}\mathbf{p} + \widetilde{h}^{bu}(\mathbf{x}),$$
$$\widetilde{h}^{L}(\mathbf{x}) \leq h(z, \mathbf{p}) \leq \widetilde{h}^{U}(\mathbf{x}),$$

for all $(z, \mathbf{p}) \in Z_D(\mathbf{x})$. Thus, this example demonstrates the proper interpretation when constructing affine relaxations that are parameterized by the seed interval and affine relaxations.

3.5 Complexity

We analyze the computational complexity of evaluating affine relaxations via the methods in this chapter. The approach taken is similar to the analysis in §4.4 of [65] for the complexity of calculating interval relaxations, and so a bound on the complexity of evaluating the affine relaxations relative to evaluating the original function is sought. Let the function of interest be $h : \mathbb{R}^n \supset Z \to \mathbb{R}$ and the goal is to construct affine relaxations on some subset of its domain. Assume that the sequence of library functions required for the evaluation of h is $\{f_1, f_2, \dots, f_N\}$ for some finite N. As in §4.4 of [65], the main assumption required for this analysis is that the cost of evaluating h, denoted $cost(h)$, is equal to the sum of the cost of evaluating each library function in the sequence $\{f_k : k \in \{1, \dots, N\}\}$; that is, $cost(h) = \sum_{k=1}^{N} cost(f_k)$ (for this discussion, "cost" is roughly measured in terms of floating-point operations).

Now, analyze one step in the evaluation of h; that is, one evaluation of a library function $f : \mathbb{R}^m \supset Y \to \mathbb{R}$. Using the notation in Proposition 3.2.2, let $(f^L, f^U) : Y_D \to \mathbb{R}^2$ be the parameterized lower and upper bounds of the interval enclosure of f (i.e. $[f^L(\mathbf{v}, \mathbf{w}), f^U(\mathbf{v}, \mathbf{w})] \ni f(\mathbf{y})$, for all $\mathbf{y} \in [\mathbf{v}, \mathbf{w}]$). Assume that the cost of evaluating f^L and f^U is no more than $\alpha cost(f)$, for some $\alpha > 0$, for any possible f in the library of functions. For instance, based on Table 3.2, the most complicated interval evaluation is for bivariate multiplication, which requires four (scalar) multiplications and three comparisons; assuming that comparison and multiplication are roughly the same cost, one could take $\alpha = 7$ for the library considered in Table 3.2. It follows that evaluation of an interval enclosure of h will be no more expensive than $2\alpha cost(h)$. Thus, the cost of evaluating an interval enclosure is bounded above by a scalar multiple of the cost of evaluating the original function. This is consistent with a slightly more detailed argument in §4.4 of [65].

To evaluate the affine relaxations, similar reasoning applies; again assume that evaluation of the affine relaxations $\mathbf{f}^{al}$, $\mathbf{f}^{au}$, f^{bl}, f^{bu} of any library function f is no more expensive than $\beta cost(f)$, for some $\beta > 0$. This is reasonable based on Table 3.3. Let $\{\mathbf{g}_i^{al}, g_i^{bl}, \mathbf{g}_i^{au}, g_i^{bu}\}$ be the values of the affine relaxations corresponding to the previously computed factors $\mathbf{g}$. Since the affine relaxations of h are with respect to each of its arguments, each of $\mathbf{g}_i^{al}$ and $\mathbf{g}_i^{au}$ are in $\mathbb{R}^n$. Assume that the cost of scalar addition ($+$) and multiplication ($\times$) are bounded above by some scalar multiple of the cost of evaluating f, for any f in the library of functions. That is, there exists $\eta > 0$ such that $cost(+) \leq \eta cost(f)$ and $cost(\times) \leq \eta cost(f)$ for all f in the library. Again, this is consistent with a slightly more detailed argument in §4.4 of [65]; also, this is a reasonable assumption since scalar addition and multiplication are two of the cheapest operations. Following the rules in Proposition 3.2.2 and keeping track of the operations required, the cost of propagating the affine relaxations for one step is bounded by $(\Psi + \Upsilon)cost(f)$, for some $\Psi > 0$ and $\Upsilon > 0$. In this case, Ψ only depends on the library of intrinsic functions used, while Υ depends on the dimension of the system, i.e. n. In more detail, let $M \in \mathbb{N}$ be the maximum number of arguments any f in the library can take. In terms of the notation in Proposition 3.2.2, the cost of evaluation of

1. $\{\mathbf{h}_i^{al} : i \in \{1, \dots, m\}\}$ is $cost(\mathbf{f}^{al}) + (mn)cost(\times) \leq (\beta)cost(f) + (Mn\eta)cost(f)$,
2. $\{\mathbf{h}_i^{au} : i \in \{1, \dots, m\}\}$ is $cost(\mathbf{f}^{au}) + (mn)cost(\times) \leq (\beta)cost(f) + (Mn\eta)cost(f)$,
3. $\{h_i^{bl}, h_i^{bu} : i \in \{1, \dots, m\}\}$ is $(2m)cost(\times) \leq (2M\eta)cost(f)$,
4. $\{\mathbf{h}^{al}, \mathbf{h}^{au}\}$ is $(2mn)cost(+) \leq (2Mn\eta)cost(f)$,
5. h^{bl} is $cost(f^{bl}) + (m)cost(+) \leq (\beta)cost(f) + (M\eta)cost(f)$,
6. h^{bu} is $cost(f^{bu}) + (m)cost(+) \leq (\beta)cost(f) + (M\eta)cost(f)$.

The sum of these bounds is $(4\beta + 4M\eta + 4Mn\eta)cost(f)$. Then let $\Psi = 4\beta + 4M\eta$ and $\Upsilon = 4Mn\eta$.

Thus the cost of evaluating the affine relaxations of the overall function h is bounded by $(\Psi + \Upsilon)\sum_{k=1}^{N} cost(f_k) = (\Psi + \Upsilon)cost(h)$. Adding in the cost of evaluating the interval relaxations makes this $(\Psi + \Upsilon + 2\alpha)cost(h)$. Therefore the cost of evaluating a pair of affine under and overestimators is no more expensive than a scalar multiple of the cost of evaluating the original function, although this multiple depends on the number of variables.

3.6 Implementation

As the expression defining the function for which affine relaxations are desired grows more complex, attempting to construct affine relaxations by the methods in this chapter by hand becomes tedious and error prone. Consequently, a C++ code implementing the rules in Proposition 3.2.2 and Tables 3.2 and 3.3 has been developed. This code is based on the `CompGraph` code originally developed by Achim Wechsung, which uses operator overloading to analyze an expression and build up the sequence of factors and library functions required to evaluate the expression (its computational graph, see Ch. 3 of [206]). With this information, it is fairly easy to go through the sequence of factors and apply Proposition 3.2.2 each time. With the aim of making the evaluation of affine relaxations as efficient as possible, C++ code is generated which performs the evaluation.

3.7 Extensions

3.7.1 Piecewise affine relaxations

One possible extension of the theory and methods described so far in this chapter would be the construction of convex piecewise affine underestimators and concave piecewise affine overestimators. That is, for some $(n_l, n_u) \in \mathbb{N}^2$ one would construct $\{(\mathbf{h}_k^{al}, h_k^{bl}) : k \in \{1, \dots, n_l\}\}$ and $\{(\mathbf{h}_k^{au}, h_k^{bu}) : k \in \{1, \dots, n_u\}\}$ such that

$$\max\{(\mathbf{h}_k^{al})^{\mathrm{T}}\mathbf{z} + h_k^{bl} : k \in \{1, \dots, n_l\}\} \le h(\mathbf{z}) \le \min\{(\mathbf{h}_k^{au})^{\mathrm{T}}\mathbf{z} + h_k^{bu} : k \in \{1, \dots, n_u\}\}$$

for all $\mathbf{z}$ in the set of interest. Since interval relaxations h^L and h^U are also constructed, we already have such a situation, with $n_l = n_u = 2$ and the second set of affine relaxations given by $\mathbf{h}_2^{al} = \mathbf{h}_2^{au} = \mathbf{0}$ and $h_2^{bl} = h^L$ and $h_2^{bu} = h^U$.

A fairly simple way to extend this idea would be to define different affine relaxations of the library functions and to repeat the construction using various combinations of the library function relaxations. For instance, as mentioned in §3.3.2, the convex and concave envelopes of the bilinear function on an interval each are defined by two affine functions. Thus, each time bivariate multiplication is required in the evaluation of an expression, four different affine relaxations could be constructed (two affine underestimators combined with two affine overestimators for the blinear function). However, it is clear that this can lead to a potentially very large number (i.e. large n_l and n_u) of possible affine relaxations; for instance, even for the simple function considered in Example 3.4.1, we could define up to $4^2 = 16$ affine under and overestimators. Of course, we could also choose, by some heuristic, to calculate only a subset of this collection of possible affine relaxations.

A prototype implementation of these ideas (specifically using the four different combinations of affine under and overestimators of the bilinear function) has been coded in MATLAB. However, numerical experiments with this code seem to indicate that there is little improvement in the quality of the piecewise affine relaxations compared to the method as already described.

3.7.2 Tighter interval bounds

Another possible extension is to use the affine relaxations to tighten the interval relaxations. Assume that we have

$$\left(\mathbf{h}^{al}(\mathbf{z}^L,\mathbf{z}^U), \mathbf{h}^{au}(\mathbf{z}^L,\mathbf{z}^U), h^{bl}(\mathbf{z}^L,\mathbf{z}^U), h^{bu}(\mathbf{z}^L,\mathbf{z}^U), h^L(\mathbf{z}^L,\mathbf{z}^U), h^U(\mathbf{z}^L,\mathbf{z}^U)\right)$$

which constitute affine and interval relaxations of the final function h on some interval $[\mathbf{z}^L,\mathbf{z}^U]$, then certainly

$$\max\left\{h^L(\mathbf{z}^L,\mathbf{z}^U), \min\{\mathbf{h}^{al}(\mathbf{z}^L,\mathbf{z}^U)^{\mathrm{T}}\mathbf{z} + h^{bl}(\mathbf{z}^L,\mathbf{z}^U) : \mathbf{z} \in [\mathbf{z}^L,\mathbf{z}^U]\}\right\} \leq \\ \min\{h(\mathbf{z}) : \mathbf{z} \in [\mathbf{z}^L,\mathbf{z}^U]\}.$$

However, this can be performed at *each* application of Proposition 3.2.2. In other words, in the conclusion of Proposition 3.2.2, h^L can be redefined as the maximum of the interval lower bound and the minimum of the affine underestimator on the underlying interval, and similarly h^U can be redefined as the minimum of the interval upper bound and the maximum of the affine overestimator. This is stated formally in the following.

Proposition 3.7.1. *Let $(m,n) \in \mathbb{N}^2$. Let $Y \subset \mathbb{R}^m$, $Z \subset \mathbb{R}^n$, and $X = \{(\mathbf{v},\mathbf{w}) \in Z \times Z : [\mathbf{v},\mathbf{w}] \subset Z\}$. Let $f : Y \to \mathbb{R}$ and $\mathbf{g} : Z \to \mathbb{R}^m$. Let $Z_D : X \ni (\mathbf{z}^L,\mathbf{z}^U) \mapsto [\mathbf{z}^L,\mathbf{z}^U]$. Let $Y_D \subset \{(\mathbf{v},\mathbf{w}) \in Y \times Y : [\mathbf{v},\mathbf{w}] \subset Y\}$. For $i \in \{1,\dots,m\}$, let $\mathbf{g}_i^{al}$ and $\mathbf{g}_i^{au}$ be locally Lipschitz continuous mappings $X \to \mathbb{R}^n$ and g_i^{bl}, g_i^{bu}, g_i^L, g_i^U be locally Lipschitz continuous mappings*

$X \to \mathbb{R}$ which for all $\mathbf{x} \in X$ satisfy

$$\mathbf{g}_i^{al}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + g_i^{bl}(\mathbf{x}) \leq g_i(\mathbf{z}) \leq \mathbf{g}_i^{au}(\mathbf{x})^{\mathrm{T}}\mathbf{z} + g_i^{bu}(\mathbf{x}),\ \forall \mathbf{z} \in Z_D(\mathbf{x}),\ \forall i,$$
$$\mathbf{g}^L(\mathbf{x}) \leq \mathbf{g}(\mathbf{z}) \leq \mathbf{g}^U(\mathbf{x}),\ \forall \mathbf{z} \in Z_D(\mathbf{x}),$$
$$(\mathbf{g}^L(\mathbf{x}),\mathbf{g}^U(\mathbf{x})) \in Y_D.$$

Let $\mathbf{f}^{al}$ and $\mathbf{f}^{au}$ be locally Lipschitz continuous mappings $Y_D \to \mathbb{R}^m$ and f^{bl}, f^{bu}, f^L, and f^U be locally Lipschitz continuous mappings $Y_D \to \mathbb{R}$ which for all $(\mathbf{v},\mathbf{w}) \in Y_D$ satisfy

$$\mathbf{f}^{al}(\mathbf{v},\mathbf{w})^{\mathrm{T}}\mathbf{y} + f^{bl}(\mathbf{v},\mathbf{w}) \leq f(\mathbf{y}) \leq \mathbf{f}^{au}(\mathbf{v},\mathbf{w})^{\mathrm{T}}\mathbf{y} + f^{bu}(\mathbf{v},\mathbf{w}),$$
$$f^L(\mathbf{v},\mathbf{w}) \leq f(\mathbf{y}) \leq f^U(\mathbf{v},\mathbf{w}),$$

for all $\mathbf{y} \in [\mathbf{v},\mathbf{w}]$.

Define $\mathbf{h}^{al}$, $\mathbf{h}^{au}$, h^{bl}, h^{bu} as in Proposition 3.2.2. Let $\widetilde{h}^L : X \ni \mathbf{x} \mapsto f^L(\mathbf{g}^L(\mathbf{x}),\mathbf{g}^U(\mathbf{x}))$ and $\widetilde{h}^U : X \ni \mathbf{x} \mapsto f^U(\mathbf{g}^L(\mathbf{x}),\mathbf{g}^U(\mathbf{x}))$. Define

$$h^{a,min} : \mathbf{x} = (\mathbf{z}^L,\mathbf{z}^U) \mapsto h^{bl}(\mathbf{z}^L,\mathbf{z}^U) + \sum_{i=1}^{n} \max\{h_i^{al}(\mathbf{x}),0\}z_i^L + \min\{h_i^{al}(\mathbf{x}),0\}z_i^U,$$
$$h^{a,max} : \mathbf{x} = (\mathbf{z}^L,\mathbf{z}^U) \mapsto h^{bu}(\mathbf{z}^L,\mathbf{z}^U) + \sum_{i=1}^{n} \max\{h_i^{au}(\mathbf{x}),0\}z_i^U + \min\{h_i^{au}(\mathbf{x}),0\}z_i^L,$$

and

$$h^L : \mathbf{x} \mapsto \max\{\widetilde{h}^L(\mathbf{x}), h^{a,min}(\mathbf{x})\},$$
$$h^U : \mathbf{x} \mapsto \min\{\widetilde{h}^U(\mathbf{x}), h^{a,max}(\mathbf{x})\}.$$

Then the conclusion of Proposition 3.2.2 holds with the definitions of h^L and h^U given here.

Proof. The result is clear noting that for all $(\mathbf{z}^L,\mathbf{z}^U) \in X$ such that $[\mathbf{z}^L,\mathbf{z}^U] \neq \varnothing$

$$h^{a,min}(\mathbf{z}^L,\mathbf{z}^U) = \min\{\mathbf{h}^{al}(\mathbf{z}^L,\mathbf{z}^U)^{\mathrm{T}}\mathbf{z} + h^{bl}(\mathbf{z}^L,\mathbf{z}^U) : \mathbf{z} \in [\mathbf{z}^L,\mathbf{z}^U]\},$$
$$h^{a,max}(\mathbf{z}^L,\mathbf{z}^U) = \max\{\mathbf{h}^{au}(\mathbf{z}^L,\mathbf{z}^U)^{\mathrm{T}}\mathbf{z} + h^{bu}(\mathbf{z}^L,\mathbf{z}^U) : \mathbf{z} \in [\mathbf{z}^L,\mathbf{z}^U]\},$$

and that by Lemma 3.2.1, $h^{a,min}$ and $h^{a,max}$ are locally Lipschitz continuous. □

An affine relaxation method based on Proposition 3.7.1 would be more expensive, since it involves the extra calculation of the minimum and maximum of the affine relaxations at each step. However, there is the potential for tighter overall relaxations. Compare this to McCormick relaxations, for instance, where the interval bounds are not (easily) improved by the convex and concave relaxations.

Chapter 4

Efficient solution of ordinary differential equations with a parametric lexicographic linear program embedded

4.1 Introduction

The focus of this chapter is the initial value problem (IVP) in ordinary differential equations (ODEs) with a parametric lexicographic linear program (LP) embedded. The LP is said to be "embedded" because the vector field depends on the solution of the lexicographic LP, which is in turn parametrized by the dynamic states. See §4.2 for a formal problem statement. The consideration of a lexicographic LP affords a lot of modeling flexibility while simultaneously enforcing a well-defined problem. This chapter focuses on the situations in which this problem can be numerically intractable and when this intractability can be difficult to detect *a priori*. The main contribution of this work is to develop a numerical method for the solution of this problem which is accurate, efficient, and robust despite these difficulties.

The situations of interest include applications to the modeling of industrial fermentation processes. This modeling framework is known as dynamic flux balance analysis (DFBA) [70, 79, 81]. In its basic form, differential equations describe the evolution of the concentrations of biomass and various metabolites of interest, such as glucose or ethanol. These equations depend on the metabolism of the microbial population, which is modeled by a parametric LP. The microbes' growth rate and uptake of resources are taken from the solution set of this LP.

One issue is that the LP may not have a singleton solution set. This means that quantities that are needed to define the dynamics of the overall system are not uniquely defined. This may lead some modelers to treat the resulting dynamic model as a differential inclusion instead. However, the ultimate goal of most research in DFBA and the motivation of this work is to obtain a numerical approximation of the solution of the dynamic problem. The idea often followed in related problems is to simulate a specifically chosen measurable selection [44, 69, 161]. The lexicographic LP provides a way to do exactly this by allowing the modeler to minimize or maximize various quantities in a hierarchical (or lexicographic) order over the solution set of the base LP model of the cellular metabolism. By minimizing or maximizing these quantities, a unique value for each is obtained. In essence, a specifc measurable selection is chosen, and the proposed method can calculate this very efficiently. The result is that the method reduces the ambiguity of the simulation results.

Another difficulty in simulating a dynamic system with an LP embedded relates to the fact that the embedded LP can be infeasible, which could induce a closed domain of definition for the dynamic system (referred to as the "domain issue"). For typical numerical integration methods for IVPs in ODEs, this is a serious issue. Certain computations that are performed by the integration method, such as predictor steps, corrector iterations, or the calculation of Jacobian information by finite differences, require the evaluation of the dynamics at states that are near the current computed solution. When the computed solution is near the boundary of this domain of definition, these states might not be in this domain, and the result is that the numerical integrator cannot obtain the necessary information and may fail, or produce incorrect results.

Consequently, our attention goes to hybrid systems theory, where different "modes" are defined on possibly closed domains [14, 55]. Typically the dynamics in those modes are trivially extended outside the domain; as in [14, 55], for instance, the definition of the dynamics on an open set is given as part of the problem statement. The challenge here is defining such an extension. Thus parametric linear programming results become important [54]. This subject is concerned with the computation of the set of values that the right-hand side of the LP constraints can take and still yield a nonempty feasible set. Using results from this literature, an appropriate extension of the domain of definition of the right-hand side of the ODEs is defined. Conceptually this is similar to some parametric programming algorithms, such as those in [123].

Inspired by these results, a method is developed which redefines the system locally as index-one differential-algebraic equations (DAEs) with an open domain. The contribution of this work is the application of the parametric LP results and hybrid systems theory to the problem of ODEs with an LP embedded; this results in a powerful and implementable numerical method which is more flexible, efficient, and accurate than previous methods. Mature methods for the solution of DAEs can be used (adaptive time-stepping and error control can be used, corrector iterations are defined, Jacobians are easy to obtain analytically or by finite differences). Further, the consideration of lexicographic LPs is a novel extension. This work's ability to handle the lexicographic LP in an efficient manner is a nontrivial development.

DFBA is considered in [70, 71, 79, 92, 149], and so these papers deal with a problem similar to the one considered here. The work in [70, 71, 79] deals with experimental validation of these models, but does not consider specific numerical issues. Meanwhile, [149] applies a differential variational inequality (DVI) formulation, and solves it with a uniform discretization in time, similarly to some time-stepping methods. This approach involves the solution of a large optimization problem (a variational inequality or mixed complementarity problem) to determine the solution trajectory all at once [4, 144], and so it is very different from numerical integration methods such as the method proposed. Further, it will be seen (see §4.6) that ODEs with LPs embedded can be extremely stiff, which motivates the proposed developments and the ability to use numerical integration methods with adaptive time steps. The work in [92] reformulates the problem as a DAE system by replacing the embedded LP with its KKT conditions. Because of the potential for a nonunique solution set, the result is that the reformulated DAE is high-index. The subsequent need to use specialized solvers for such systems also motivates the current developments, in which an index-one DAE is obtained. As mentioned, more established numerical integration methods can be used. Finally, the aforementioned references have not explored the domain issue as it relates to DFBA, which is a significant source of numerical intractability of the ODEs with LP embedded problem. The use of a lexicographic LP distinguishes this work as well.

The rest of this chapter is organized as follows. Section 4.2 introduces notation and necessary concepts and formally states the problem. Section 4.3 provides motivating discussion and examples which highlight some of the difficulties inherent in the problem formulation. Section 4.4 considers existence and uniqueness results for the solutions of the ODE. In the context of this work, this serves as more motivation for the numerical developments. Section 4.5 represents the main contribution of this work, and states the proposed algorithm for solving the ODE with LP embedded problem, which includes a specific method for solving the lexicographic LP. Section 4.6 applies the algorithm to models of industrial fermentation processes using DFBA.

4.2 Problem statement and preliminaries

The formal problem statement is as follows. For $(n_x, n_q, m, n_v) \in \mathbb{N}^4$, let $D_t \subset \mathbb{R}$, $D_x \subset \mathbb{R}^{n_x}$ and $D_q \subset \mathbb{R}^{n_q}$ be nonempty open sets. Let $\mathbf{f} : D_t \times D_x \times D_q \to \mathbb{R}^{n_x}$, $\mathbf{b} : D_t \times D_x \to \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times n_v}$, and $\mathbf{c}_i \in \mathbb{R}^{n_v}$ for $i \in \{1, \dots, n_q\}$ be given. First, let $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$. Let $\widehat{\mathbf{q}} : \mathbb{R}^m \to \overline{\mathbb{R}}^{n_q}$ be defined by

$$\begin{aligned} \widehat{q}_1 : \mathbf{d} \mapsto \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \ & \mathbf{c}_1^{\mathrm{T}} \mathbf{v} \\ \text{s.t. } & \mathbf{A}\mathbf{v} = \mathbf{d}, \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{4.1}$$

and for $i \in \{2, \dots, n_q\}$,

$$\begin{aligned} \widehat{q}_i : \mathbf{d} \mapsto \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \ & \mathbf{c}_i^{\mathrm{T}} \mathbf{v} \\ \text{s.t. } & \begin{bmatrix} \mathbf{A} \\ \mathbf{c}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{c}_{i-1}^{\mathrm{T}} \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{d} \\ \widehat{q}_1(\mathbf{d}) \\ \vdots \\ \widehat{q}_{i-1}(\mathbf{d}) \end{bmatrix}, \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{4.2}$$

Subsequently, define

$$F \equiv \left\{\mathbf{d} \in \mathbb{R}^m : -\infty < \widehat{q}_i(\mathbf{d}) < +\infty, \forall i \in \{1, \dots, n_q\}\right\}, \tag{4.3}$$
$$K \equiv \mathbf{b}^{-1}(F).$$

Note that $K \subset D_t \times D_x$.

The focus of this work is an initial value problem in ODEs: given a $t_0 \in D_t$ and $\mathbf{x}_0 \in D_x$, we seek an interval $[t_0, t_f] = I \subset D_t$, and absolutely continuous function $\mathbf{x} : I \to D_x$ which satisfy

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{q}(t, \mathbf{x}(t))\right), \quad a.e.\ t \in I, \tag{4.4a}$$
$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{4.4b}$$

where $\mathbf{q} : K \to \mathbb{R}^{n_q} : (t, \mathbf{z}) \mapsto \widehat{\mathbf{q}}(\mathbf{b}(t, \mathbf{z}))$. Such an I and $\mathbf{x}$ are called a solution of IVP (4.4).

Linear program (4.2) is called the i^{th}-level LP; it is an optimization problem over the solution set of the $(i-1)^{th}$-level LP, where the first-level LP is given by (4.1). Together, these LPs are called a lexicographic linear program, using the terminology from [154] (further background on lexicographic optimization is presented in §4.5.3). Note that any solution of the n_q^{th}-level LP must also be a solution of the i^{th}-level LP, $i \in \{1, \dots, n_q - 1\}$.

Proposition 4.2.1 establishes an important topological property of F, the domain of $\widehat{\mathbf{q}}$.

Proposition 4.2.1. *Assume F defined in* (4.3) *is nonempty. Then*

$$F = \left\{\mathbf{A}\mathbf{v} \in \mathbb{R}^m : \mathbf{v} \geq \mathbf{0}\right\},$$

and thus it is closed.

Proof. Choose any $\mathbf{d} \in F$. It follows that $\widehat{q}_1(\mathbf{d})$ is finite, which implies that the first-level LP is feasible for $\mathbf{d}$; i.e. $\mathbf{A}\mathbf{v} = \mathbf{d}$ for some $\mathbf{v} \geq \mathbf{0}$. Thus $F \subset \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$.

Conversely, since F is nonempty, there exists $\mathbf{d}^* \in \mathbb{R}^m$ such that $\widehat{q}_i(\mathbf{d}^*)$ is finite for each i. Consequently, $\widehat{q}_1(\mathbf{d}^*) = \max\left\{(\mathbf{d}^*)^{\mathrm{T}}\mathbf{w} : \mathbf{A}^{\mathrm{T}}\mathbf{w} \leq \mathbf{c}_1\right\}$; i.e. the dual of the first-level LP is feasible and has a bounded solution. Note that the dual is feasible for any value of $\mathbf{d}$ (its feasible set is invariant). Thus, using duality results such as those in Table 4.2 of [25], $\widehat{q}_1(\mathbf{d})$ is finite for all $\mathbf{d}$ such that the first-level LP is feasible (i.e. for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$).

Next, assume that $\widehat{q}_{i-1}(\mathbf{d})$ is finite for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$. Since $\widehat{q}_i(\mathbf{d}^*)$ is finite, a similar argument establishes that $\widehat{q}_i(\mathbf{d})$ is finite for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$. Proceeding by induction, one has that for each $i \in \{1, \dots, n_q\}$, $\widehat{q}_i(\mathbf{d})$ is finite for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$. Thus $\{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\} \subset F$ and, combined with the inclusion above, equality follows. □

4.3 Domain issues

This section demonstrates how domain issues are manifested as numerical complications by applying other methods to simple instances of (4.4). To understand this from a theoretical view, note that any solution of (4.4) must satisfy $(t, \mathbf{x}(t)) \in K$, *a.e.* $t \in I$, otherwise $\mathbf{q}(t, \mathbf{x}(t))$ is undefined on a set of nonzero measure, and consequently Eqn. (4.4a) does not hold. Consequently, even though $D_t \times D_x$ is nonempty and open, the effective domain of definition of the system, K, may not be either of those.

4.3.1 Direct method

The direct method refers to solving IVP (4.4) by using a standard numerical integrator and calling an LP solver directly from the function evaluation subroutine to determine the dynamics. This approach can be made quite efficient, especially as it can rely on established commercial codes for the numerical integration and LP solution. Unfortunately, it can also be unreliable. Consider the following example:

$$\mathbf{x}(0) = \mathbf{0}, \qquad \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), q(\mathbf{x}(t))\right) = \begin{bmatrix} 1 \\ x_2(t)q(\mathbf{x}(t)) - (x_2(t))^2 + 2x_1(t) \end{bmatrix},$$

where $q(\mathbf{z}) = \min\{v : z_1^2 \leq v \leq z_2\}$.

The first thing to note is that the LP is feasible only if $\mathbf{z} \in K = \{\mathbf{z} : z_1^2 \leq z_2\}$. Although this is a closed set, we can verify that $\mathbf{x}(t) = (t, t^2)$ is a solution; $\mathbf{x}(0) = (0, 0)$, $q(\mathbf{x}(t)) = t^2$, and $\mathbf{f}\left(\mathbf{x}(t), q(\mathbf{x}(t))\right) = (1, 2t) = \dot{\mathbf{x}}(t)$. Consider now what happens when applying an explicit Euler step. Let $\widetilde{\mathbf{x}}(t)$ be the numerical estimate of the solution at t. Then for $h > 0$ and

$\widetilde{\mathbf{x}}(0) = \mathbf{x}(0)$,

$$\begin{aligned}\widetilde{\mathbf{x}}(0+h) &= \widetilde{\mathbf{x}}(0) + h\mathbf{f}\left(\widetilde{\mathbf{x}}(0), q(\widetilde{\mathbf{x}}(0))\right)\\ &= \mathbf{0} + h(1,0) = (h,0).\end{aligned}$$

We see that $\widetilde{\mathbf{x}}(h) \notin K$. Thus when attempting to evaluate $q(\widetilde{\mathbf{x}}(h))$ for the next step, we encounter an infeasible LP, and the numerical method fails.

Although explicit Euler is a very simple method, the explicit Euler step is often a part of more sophisticated integration methods; the second stage derivative of an explicit Runge-Kutta method is evaluated after taking an explicit Euler step, and the initial predictor of many linear multistep predictor-corrector methods is given by an explicit Euler step [105]. Meanwhile, numerical integration methods which do not involve an explicit Euler step will often involve an *implicit* Euler step; this includes the backwards differentiation formulas (BDF) and semi-implicit Runge-Kutta methods [105], where again the first step of a BDF method is an implicit Euler step, and the first stage derivative of a semi-implicit Runge-Kutta method is determined by an implicit Euler step [105]. For the example above, an implicit method may work, but there is nothing intrinsic to an implicit method that avoids the domain issue; see §4.3.2 for a counterexample. In fact, implicit methods have more opportunities to fail when simulating ODEs with an LP embedded. Implicit methods typically must solve nonlinear equations by a fixed-point or Newton iteration. Since $\mathbf{f}$ and thus $\mathbf{q}$ must be evaluated at each point produced by the iteration, the sequence of iterates must be in K, which need not hold in general. Further, obtaining Jacobian information by finite differences provides another point of potential failure, as the perturbed states may not be in K.

4.3.2 DVI time-stepping method

Time-stepping methods refer to a class of numerical methods for solving an initial-value DVI [4, 8, 144, 205]. The solution set of an LP is equivalent to the solution set of its KKT conditions, and the KKT conditions are a type of complementarity problem or variational inequality. Thus ODEs with an LP embedded are a special case of a DVI, and one could potentially apply a time-stepping method to IVP (4.4). However, as the essential step in these methods is the solution of a system of equations with conditions that are equivalent to the embedded LP having an optimal solution, they do not differ in a meaningful way from the direct method previously mentioned. More generally, implicit integration methods also suffer from domain issues.

As a counterexample, consider a problem similar to the one in §4.3.1:

$$\mathbf{x}(0) = \mathbf{0}, \quad \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), q(\mathbf{x}(t))\right) = \begin{bmatrix} 1 \\ x_2(t)q(\mathbf{x}(t)) - (x_2(t))^2 + 2x_1(t) \end{bmatrix}, \tag{4.5}$$

where $q(\mathbf{z}) = \min\{v : z_2 \le v \le z_1^2\}$.

The embedded LP is feasible only if $\mathbf{x} \in K = \{\mathbf{z} : z_2 \le z_1^2\}$, a closed, nonconvex set. Note that $\mathbf{x}(t) = (t, t^2)$ is a solution. Letting $\mathbf{b}(\mathbf{z}) = (z_1^2, -z_2)$ and rewriting q in terms of the embedded LP's dual, we have

$$q(\mathbf{z}) = \max\{\mathbf{b}(\mathbf{z})^{\mathrm{T}}\mathbf{w} : \mathbf{w} \le 0, w_1 - w_2 = 1\}.$$

Letting W denote the feasible set of the above (dual) LP, this is equivalent to finding $\mathbf{w}^* \in W$ such that $(\mathbf{w} - \mathbf{w}^*)^{\mathrm{T}}(-\mathbf{b}(\mathbf{z})) \ge 0$, $\forall \mathbf{w} \in W$. This is a parametric variational inequality and is denoted $\mathrm{VI}(W, -\mathbf{b}(\mathbf{z}))$. This requires the dynamics to be rewritten as

$$\begin{aligned}&\mathbf{f}\left(\mathbf{x}(t), q(\mathbf{x}(t))\right) = \\ &\quad \widehat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t)) = \begin{bmatrix} 1 \\ x_2(t)\left((x_1(t))^2 u_1(t) - x_2(t)u_2(t)\right) - (x_2(t))^2 + 2x_1(t) \end{bmatrix},\end{aligned}$$

where $\mathbf{u}(t)$ is a solution of $\mathrm{VI}\left(W, -\mathbf{b}(\mathbf{x}(t))\right)$. Given $h > 0$, an implicit time-stepping scheme takes the form

$$\begin{aligned}\widetilde{\mathbf{x}}(t+h) &= \widetilde{\mathbf{x}}(t) + h\widehat{\mathbf{f}}\left(\widetilde{\mathbf{x}}(t+h), \widetilde{\mathbf{u}}(t+h)\right), \\ \widetilde{\mathbf{u}}(t+h) &\text{ solves } \mathrm{VI}\left(W, -\mathbf{b}(\widetilde{\mathbf{x}}(t+h))\right).\end{aligned} \tag{4.6}$$

Typically this implicit system is solved as the equivalent variational inequality $\mathrm{VI}(\mathbb{R}^2 \times W, \mathbf{g}^t)$

where

$$\mathbf{g}^t : (\mathbf{z}, \mathbf{v}) \mapsto \begin{bmatrix} z_1 - \widetilde{x}_1(t) - h \\ z_2 - \widetilde{x}_2(t) - h(z_1^2 z_2 v_1 - z_2^2 v_2 - z_2^2 + 2z_1) \\ -z_1^2 \\ z_2 \end{bmatrix}$$

(see for instance [144]). However, again letting $\widetilde{\mathbf{x}}(0) = \mathbf{x}(0) = \mathbf{0}$, the initial variational inequality $\mathrm{VI}(\mathbb{R}^2 \times W, \mathbf{g}^0)$ does *not* have a solution for any choice of h.

To see this, assume, for a contradiction, that a solution exists. Then there is a $(\mathbf{z}^*, \mathbf{v}^*) \in \mathbb{R}^2 \times W$ such that

$$\begin{aligned} &(z_1 - z_1^*)(z_1^* - h) + (z_2 - z_2^*)\left(z_2^* - h((z_1^*)^2 z_2^* v_1^* - (z_2^*)^2 v_2^* - (z_2^*)^2 + 2z_1^*)\right) + \\ &(v_1 - v_1^*)(-(z_1^*)^2) + (v_2 - v_2^*)(z_2^*) \geq 0, \end{aligned} \tag{4.7}$$

for all $(\mathbf{z}, \mathbf{v}) \in \mathbb{R}^2 \times W$. First note that $z_1^* = h$, otherwise we could always find a $z_1 \in \mathbb{R}$ such that the inequality (4.7) did not hold. Similarly, we must have

$$z_2^* = h\left((z_1^*)^2 z_2^* v_1^* - (z_2^*)^2 v_2^* - (z_2^*)^2 + 2z_1^*\right). \tag{4.8}$$

Using this in inequality (4.7), we obtain

$$(v_1 - v_1^*)(-h^2) + (v_2 - v_2^*)(z_2^*) \geq 0,$$

for all $(\mathbf{z}, \mathbf{v}) \in \mathbb{R}^2 \times W$. For any $\mathbf{v} \in W$, we can write $v_2 = v_1 - 1$. Then we get

$$(v_1 - v_1^*)(-h^2) + (v_1 - 1 - (v_1^* - 1))(z_2^*) \geq 0,$$

which yields

$$(v_1 - v_1^*)(z_2^* - h^2) \geq 0,$$

for all $v_1 \leq 0$, where $v_1^* \leq 0$ and $z_2^* \in \mathbb{R}$ satisfy

$$hv_1^*(z_2^*)^2 + (1 - h^3 v_1^*)z_2^* - 2h^2 = 0 \tag{4.9}$$

(which is obtained from Eqn. (4.8) via the substitutions $z_1^* = h$ and $v_2^* = v_1^* - 1$).

We can now analyze three cases:

1. $z_2^* > h^2$: However, if this was the case, then whatever the value of v_1^*, we could always find a $v_1' < v_1^*$ which then implies $(v_1' - v_1^*)(z_2^* - h^2) < 0$, which is a contradiction.
2. $z_2^* < h^2$: However, if this was the case, we must have $v_1^* = 0$, otherwise there exists a v_1' such that $v_1^* < v_1' \leq 0$ which then implies that $(v_1' - v_1^*)(z_2^* - h^2) < 0$. Thus, assuming $v_1^* = 0$, use Eqn. (4.9) to check the value of z_2^*. However, that yields $z_2^* = 2h^2$, which contradicts $z_2^* < h^2$.
3. $z_2^* = h^2$: However, if this was the case, we can use Eqn. (4.9) to check the consistency of values. This yields

$$h^5 v_1^* + h^2 - h^5 v_1^* - 2h^2 = 0 \implies -h^2 = 0,$$

which contradicts $h > 0$.

Thus, it follows that there does not exist a point $(\mathbf{z}^*, \mathbf{v}^*) \in \mathbb{R}^2 \times W$ which solves $\mathrm{VI}(\mathbb{R}^2 \times W, \mathbf{g}^0)$.

Note that the implicit time-stepping scheme (4.6) is equivalent to the direct method applying an implicit Euler step to the original system (4.5). Thus, the failure of the system (4.6) to have a solution indicates that the direct method, even with an implicit integration routine, also fails.

4.4 Existence of solutions

This section presents some results for the existence and uniqueness of solutions of IVP (4.4). First, the idea of an "extended IVP" is introduced. This represents an approach from the perspective of hybrid systems theory, and existence results based on this idea lead to a useful understanding of how to approach the IVP (4.4) numerically. Then an existence result based on viability theory is proved, and it is discussed why this does not lead to a useful numerical method.

4.4.1 Extended solutions

The following theorem presents what is essentially an a posteriori check for existence. In the following λ denotes Lebesgue measure.

Theorem 4.4.1. *Suppose* $\widehat{\mathbf{q}}^E : \mathbb{R}^m \to \mathbb{R}^{n_q}$ *is an extension of* $\widehat{\mathbf{q}}$ *(i.e.* $\widehat{\mathbf{q}}^E$ *is defined on all of* $\mathbb{R}^m$ *and* $\widehat{\mathbf{q}}^E$ *restricted to* F *equals* $\widehat{\mathbf{q}}$*),* $\mathbf{b}(\cdot, \mathbf{z})$ *is measurable for all* $\mathbf{z} \in D_x$*,* $\mathbf{b}(t, \cdot)$ *is continuous for a.e.* $t \in D_t$*, and there exist an interval* $I^E = [t_0, t_f^E]$ *and absolutely continuous function* $\mathbf{x} : I^E \to D_x$ *which are a solution of the IVP*

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(t, \mathbf{x}(t), \widehat{\mathbf{q}}^E\left(\mathbf{b}(t, \mathbf{x}(t))\right)\right), \quad a.e.\ t \in I^E, \tag{4.10a}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0. \tag{4.10b}$$

Letting $S(t) = \{s \in [t_0, t] : (s, \mathbf{x}(s)) \notin K\}$*, if* $(t_0, \mathbf{x}_0) \in K$ *and*

$$t_f = \sup\left\{t \in I^E : \lambda\left(S(t)\right) = 0\right\},$$

then $I = [t_0, t_f]$ *and* $\mathbf{x}$ *restricted to* I *are a solution of IVP* (4.4)*. Furthermore, this is the largest interval on which* $\mathbf{x}$ *is a solution of* (4.4)*.*

Proof. Since $\mathbf{x}$ is continuous, the composite function $\mathbf{b}_{\mathbf{x}} : I^E \to \mathbb{R}^m : t \mapsto \mathbf{b}(t, \mathbf{x}(t))$ is measurable (see Lemma 1 in §1 of [51]). By Proposition 4.2.1, the complement of F, F^C, is open, so $S^E = \mathbf{b}_{\mathbf{x}}^{-1}(F^C)$ is measurable. Then one has $\lambda(S(t)) = \int_{[t_0,t]} \chi_{S^E}(s) ds$, where χ_{S^E} is the indicator function of S^E. This implies that $\lambda(S(\cdot))$ is continuous and increasing.

Thus, $\lambda(S(t_f)) = 0$ and so for almost every $t \in I$, $(t, \mathbf{x}(t)) \in K$ and therefore $\mathbf{b}(t, \mathbf{x}(t)) \in F$. So $\mathbf{q}(t, \mathbf{x}(t)) = \widehat{\mathbf{q}}^E(\mathbf{b}(t, \mathbf{x}(t)))$ for almost every $t \in I$, which combined with Eqn. (4.10a) implies $\mathbf{x}$ satisfies (4.4a) for almost every $t \in I$, and thus is a solution. The second claim follows easily; for $t' > t_f$, $\lambda(S(t')) > 0$ and so Eqn. (4.4a) cannot be satisfied for almost every $t \in [t_0, t']$. □

Refer to IVP (4.10) as the "extended IVP." Note that the interval I in Theorem 4.4.1 could be degenerate, i.e. $t_0 = t_f$. This leads to a somewhat trivial solution. Ruling out this case requires something akin to the sufficient conditions for existence from viability-type results; see §4.4.2.

The characterization of t_f given in Theorem 4.4.1 is not in a particularly useful form. The next result alleviates this under stricter assumptions on $\mathbf{b}$.

Corollary 4.4.2. *Suppose there is a solution* $I^E = [t_0, t_f^E]$*,* $\mathbf{x}$ *of the extended IVP* (4.10)*. Let* $S(t) = \{s \in [t_0, t] : (s, \mathbf{x}(s)) \notin K\}$ *and* $t_f = \sup\{t \in I^E : \lambda(S(t)) = 0\}$*. Assume that for* *any $t \in D_t$, there exists an interval $[t_1, t_2) \subset D_t$ such that $[t_1, t_2) \ni t$ and* $\mathbf{b}$ *is continuous on $[t_1, t_2) \times D_x$. Then $t_f = \inf\{t \in I^E : (t, \mathbf{x}(t)) \notin K\}$.*

Proof. For a contradiction, assume $t_f > \inf\{t \in I^E : (t, \mathbf{x}(t)) \notin K\}$, that is, there exists a $t^* \in I^E$ such that $t^* < t_f$ and $(t^*, \mathbf{x}(t^*)) \notin K$. By assumption, there is an interval $[t_1, t_2) \ni t^*$ on which $\mathbf{b}$ is continuous. Without loss of generality, assume $t_2 < t_f$. Then since $\mathbf{x}$, as a solution of the extended IVP, is continuous, $\mathbf{b}_{\mathbf{x}} : t \mapsto \mathbf{b}(t, \mathbf{x}(t))$ is continuous on $[t^*, t_2)$, and $\mathbf{b}_{\mathbf{x}}(t^*) \notin F$. By Proposition 4.2.1, the complement of F, F^C, is open, so $\mathbf{b}_{\mathbf{x}}^{-1}(F^C)$ is open in $[t^*, t_2)$ and nonempty. Thus there exists $t^{**} \in (t^*, t_2)$ such that $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ for all $t \in [t^*, t^{**})$. This implies that $\lambda(S(t^{**})) > 0$. But as in the proof of Theorem 4.4.1, $\lambda(S(\cdot))$ is increasing on I^E, and so $t_f \leq t^{**}$, which contradicts $t^{**} < t_2 < t_f$.

Now, assume $t_f < \inf\{t \in I^E : (t, \mathbf{x}(t)) \notin K\}$. This implies that there exists a $t^* > t_f$ such that $(t, \mathbf{x}(t)) \in K$ for all $t < t^*$, and so $\lambda(S(t^*)) = 0$. But this contradicts the definition of t_f as a supremum. □

Corollary 4.4.2 says that, under the appropriate conditions on $\mathbf{b}$ (roughly, "continuity from the right"), a solution of the extended IVP ceases to be a solution of the original system (4.4) at the first time the solution trajectory leaves K. Intuitively this makes sense, but this intuition can lead to trouble for the numerical method as demonstrated in §4.3; just because one cannot find a solution of the LP at a specific step in the numerical procedure does not mean that a solution no longer exists. Care must be taken when applying Corollary 4.4.2.

Since $(t, \mathbf{z}) \mapsto \mathbf{f}(t, \mathbf{z}, \widehat{\mathbf{q}}^E(\mathbf{b}(t, \mathbf{z})))$ is defined on the open set $D_t \times D_x$ (assuming $\widehat{\mathbf{q}}^E(\mathbf{b}(t, \mathbf{z}))$ is in D_q for all $(t, \mathbf{z})$), standard existence and uniqueness results can now be applied to the extended IVP. The main concern is whether we can define an appropriate extension $\widehat{\mathbf{q}}^E$. In fact, we can define a Lipschitz continuous extension.

Proposition 4.4.1. *There exists a Lipschitz continuous function $\widehat{\mathbf{q}}^E : \mathbb{R}^m \to \mathbb{R}^{n_q}$ such that $\widehat{\mathbf{q}}^E$ restricted to F equals $\widehat{\mathbf{q}}$, the solution of the lexicographic linear program* (4.1)-(4.2).

Proof. If F is empty the result is trivial. Otherwise, assume without loss of generality that the first $k_1 = \text{rank}(\mathbf{A})$ rows of $\mathbf{A}$ are linearly independent and let $\widetilde{\mathbf{A}}_1 \in \mathbb{R}^{k_1 \times n_v}$ be a matrix formed from the first k_1 rows of $\mathbf{A}$. Define $\mathbf{d}_1^E : \mathbb{R}^m \to \mathbb{R}^{k_1}$ as $\mathbf{d} \mapsto (d_1, d_2, \ldots, d_{k_1})$. Then

$$\widehat{q}_1(\mathbf{d}) = \min\{\mathbf{c}_1^{\mathrm{T}}\mathbf{v} : \widetilde{\mathbf{A}}_1\mathbf{v} = \mathbf{d}_1^E(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\} \tag{4.11}$$

for all $\mathbf{d} \in F$. Since F is nonempty, it follows that the dual of LP (4.11) has a nonempty feasible set. Furthermore, by the discussion in §5.2 of [25], $\{\mathbf{p}_j \in \mathbb{R}^{k_1} : 1 \leq j \leq n_1\}$, the set of vertices of the feasible set of the dual of LP (4.11), is nonempty and $\widehat{q}_1(\mathbf{d}) = \max\{\mathbf{p}_j^{\mathrm{T}}\mathbf{d}_1^E(\mathbf{d}) : 1 \leq j \leq n_1\}$ for $\mathbf{d} \in F$. However, this is perfectly well-defined and Lipschitz continuous for all $\mathbf{d} \in \mathbb{R}^m$, so let

$$\widehat{q}_1^E : \mathbf{d} \mapsto \max\{\mathbf{p}_j^{\mathrm{T}}\mathbf{d}_1^E(\mathbf{d}) : 1 \leq j \leq n_1\}.$$

Then assume full row rank $\widetilde{\mathbf{A}}_i \in \mathbb{R}^{k_i \times n_v}$ and Lipschitz continuous $\widehat{q}_i^E$ and $\mathbf{d}_i^E : \mathbb{R}^m \to \mathbb{R}^{k_i}$ have been constructed such that $\widehat{q}_i^E$ restricted to F equals $\widehat{q}_i$ and $\{\mathbf{v} : \widetilde{\mathbf{A}}_i\mathbf{v} = \mathbf{d}_i^E(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\}$ equals the feasible set of the i^{th}-level LP for all $\mathbf{d} \in F$. If $\mathbf{c}_i$ and the rows of $\widetilde{\mathbf{A}}_i$ are linearly independent, let $k_{i+1} = k_i + 1$, $\widetilde{\mathbf{A}}_{i+1} = \begin{bmatrix} \widetilde{\mathbf{A}}_i \\ \mathbf{c}_i^{\mathrm{T}} \end{bmatrix}$, and $\mathbf{d}_{i+1}^E : \mathbf{d} \mapsto (\mathbf{d}_i^E(\mathbf{d}), \widehat{q}_i^E(\mathbf{d}))$; otherwise let $k_{i+1} = k_i$, $\widetilde{\mathbf{A}}_{i+1} = \widetilde{\mathbf{A}}_i$, and $\mathbf{d}_{i+1}^E : \mathbf{d} \mapsto \mathbf{d}_i^E(\mathbf{d})$. Then

$$\widehat{q}_{i+1}(\mathbf{d}) = \min\{\mathbf{c}_{i+1}^{\mathrm{T}}\mathbf{v} : \widetilde{\mathbf{A}}_{i+1}\mathbf{v} = \mathbf{d}_{i+1}^E(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\} \tag{4.12}$$

for all $\mathbf{d} \in F$. Similarly to the induction basis, let $\{\mathbf{p}_j \in \mathbb{R}^{k_{i+1}} : 1 \leq j \leq n_{i+1}\}$ be the nonempty set of vertices of the feasible set of the dual of LP (4.12); then let

$$\widehat{q}_{i+1}^E : \mathbf{d} \mapsto \max\{\mathbf{p}_j^{\mathrm{T}}\mathbf{d}_{i+1}^E(\mathbf{d}) : 1 \leq j \leq n_{i+1}\}.$$

Then $\widehat{q}_{i+1}^E$ is also Lipschitz continuous, and when restricted to F it equals $\widehat{q}_{i+1}$. Proceeding by induction, we obtain the desired Lipschitz continuous extension $\widehat{\mathbf{q}}^E$. □

For completeness, a local existence and uniqueness result for the extended IVP is stated. Furthermore, the assumptions of the following result provide basic conditions under which the extended IVP is numerically tractable. Weakening the assumptions to allow $\mathbf{f}$ to be measurable with respect to time can be done by following results in Ch. 1 of [51].

Proposition 4.4.2. *Assume*

1. $\widehat{\mathbf{q}}^E(\mathbf{b}(t_0, \mathbf{x}_0)) \in D_q$,
2. *there exists* $t_1 > t_0$ *such that* $\mathbf{b}$ *is continuous on* $[t_0, t_1) \times D_x$ *and* $\mathbf{f}$ *is continuous on* $[t_0, t_1) \times D_x \times D_q$, *and*
3. *there exist open neighborhoods* $N_x \ni \mathbf{x}_0$ *and* $N_q \ni \widehat{\mathbf{q}}^E(\mathbf{b}(t_0, \mathbf{x}_0))$, *constants* $L_b \geq 0$,

$L_f \geq 0$, such that for all $(t, \mathbf{z}_1, \mathbf{z}_2, \mathbf{p}_1, \mathbf{p}_2) \in [t_0, t_1) \times N_x \times N_x \times N_q \times N_q$,

$$\|\mathbf{b}(t, \mathbf{z}_1) - \mathbf{b}(t, \mathbf{z}_2)\| \leq L_b \|\mathbf{z}_1 - \mathbf{z}_2\|,$$

$$\|\mathbf{f}(t, \mathbf{z}_1, \mathbf{p}_1) - \mathbf{f}(t, \mathbf{z}_2, \mathbf{p}_2)\| \leq L_f(\|\mathbf{z}_1 - \mathbf{z}_2\| + \|\mathbf{p}_1 - \mathbf{p}_2\|).$$

Then a unique solution of the IVP (4.10) *exists.*

Proof. By Proposition 4.4.1, we can assume $\widehat{\mathbf{q}}^E$ is Lipschitz continuous with constant L_q, so $\mathbf{q}^E = \widehat{\mathbf{q}}^E \circ \mathbf{b}$ is continuous on $[t_0, t_1) \times D_x$ and satisfies

$$\left\|\mathbf{q}^E(t, \mathbf{z}_1) - \mathbf{q}^E(t, \mathbf{z}_2)\right\| \leq L_q L_b \|\mathbf{z}_1 - \mathbf{z}_2\|.$$

Since $\mathbf{q}^E$ is continuous, we can assume without loss of generality that $\mathbf{q}^E(t, \mathbf{z}) \in N_q$ for all $(t, \mathbf{z}) \in [t_0, t_1) \times N_x$. Thus,

$$\left\|\mathbf{f}(t, \mathbf{z}_1, \mathbf{q}^E(t, \mathbf{z}_1)) - \mathbf{f}(t, \mathbf{z}_2, \mathbf{q}^E(t, \mathbf{z}_2))\right\| \leq L_f(1 + L_q L_b) \|\mathbf{z}_1 - \mathbf{z}_2\|$$

(i.e. $(t, \mathbf{z}) \mapsto \mathbf{f}(t, \mathbf{z}, \mathbf{q}^E(t, \mathbf{z}))$ is locally Lipschitz continuous on N_x, uniformly on $[t_0, t_1)$, as in Definition 2.5.1). Therefore we can apply Theorem 2.3 of Ch. II of [116] to the mapping $(t, \mathbf{z}) \mapsto \mathbf{f}(t, \mathbf{z}, \mathbf{q}^E(t, \mathbf{z}))$ and conclude that there exists a $t_f^E > t_0$ and continuous function $\mathbf{x}$ on $[t_0, t_f^E]$ which are a solution of (4.10). □

4.4.2 Viability-based existence

This section presents a tangential discussion, mostly to contrast with the theory in the previous section. An existence result based on viability theory [10] is proved. While in theory such an existence result would allow us to apply numerical integration schemes that can overcome the fact that K might not be open, we will see that it relies on a condition that is next to impossible to verify in the situations of interest.

The following background is helpful. Recalling the discussion of local compactness from §2.2, we have the additional results (see [131] for proofs and further background): the finite product of locally compact metric spaces is locally compact (i.e. if X and Y are locally compact metric spaces then so is $X \times Y$); for $t_0 < t_1$, $[t_0, t_1) \subset \mathbb{R}$ is locally compact. The contingent cone (sometimes called the Bouligand tangent cone) $T_V(\mathbf{v})$ of a set $V \subset \mathbb{R}^n$ at

$\mathbf{v} \in \overline{V}$ is given by

$$T_V(\mathbf{v}) = \left\{ \mathbf{w} \in \mathbb{R}^n : \liminf_{h \to 0^+} \frac{d(\mathbf{v} + h\mathbf{w}, V)}{h} = 0 \right\}.$$

The following lemmata establish some required properties of the contingent cone.

Lemma 4.4.3. *If $\mathbf{w} \in T_V(\mathbf{v})$, then for any open set $N \ni \mathbf{v}$, $\mathbf{w} \in T_{V \cap N}(\mathbf{v})$.*

Proof. If $\mathbf{w} \in T_V(\mathbf{v})$, an equivalent characterization is that there exist sequences $h_n \to 0$, $h_n > 0$ for all n, and $\mathbf{w}_n \to \mathbf{w}$ such that for all $n \in \mathbb{N}$, $\mathbf{v} + h_n\mathbf{w}_n \in V$ (see §1.1 of [10]). Since N is open, for sufficiently small h, $\mathbf{v} + h\widehat{\mathbf{w}} \in N$ for any $\widehat{\mathbf{w}}$. Thus, there are subsequences h_{n_k} and $\mathbf{w}_{n_k}$ of h_n and $\mathbf{w}_n$, respectively, such that $\mathbf{v} + h_{n_k}\mathbf{w}_{n_k} \in N$ for all $k \in \mathbb{N}$. Consequently, $\mathbf{v} + h_{n_k}\mathbf{w}_{n_k} \in V \cap N$ for all $k \in \mathbb{N}$, and so $\mathbf{v} \in \overline{V \cap N}$ and $\mathbf{w} \in T_{V \cap N}(\mathbf{v})$. □

Lemma 4.4.4. *If $\mathbf{w} \in T_V(\mathbf{v})$, $\mathbf{v} = (v, \widetilde{\mathbf{v}})$ and $\mathbf{w} = (1, \widetilde{\mathbf{w}})$, then $\mathbf{w} \in T_{V \cap R}(\mathbf{v})$ for any $R = [v_a, v_b) \times \widetilde{N}$, where $\widetilde{N}$ is an open set containing $\widetilde{\mathbf{v}}$ and $v_a \leq v < v_b$.*

Proof. The proof proceeds similarly to that of Lemma 4.4.3. There exist sequences $h_n \to 0$, $h_n > 0$, and $(y_n, \widetilde{\mathbf{w}}_n) \to (1, \widetilde{\mathbf{w}})$, such that $(v, \widetilde{\mathbf{v}}) + h_n(y_n, \widetilde{\mathbf{w}}_n) \in V$ for all n. Then, for large enough n, $(v, \widetilde{\mathbf{v}}) + h_n(y_n, \widetilde{\mathbf{w}}_n) \in R$ since $\widetilde{N}$ is open and v is a limit point of $[v_a, v_b)$. Thus there are subsequences such that $(v, \widetilde{\mathbf{v}}) + h_{n_k}(y_{n_k}, \widetilde{\mathbf{w}}_{n_k}) \in V \cap R$, and it follows that $\mathbf{v} \in \overline{V \cap R}$ and $\mathbf{w} \in T_{V \cap R}(\mathbf{v})$. □

The following result is an example of the kind of conditions that would be sufficient to establish existence of a solution of IVP (4.4).

Proposition 4.4.5. *If $(t_0, \mathbf{x}_0) \in K$, $\mathbf{q}(t_0, \mathbf{x}_0) \in D_q$, $\mathbf{f}$ is continuous, $\mathbf{b}$ is continuous, and there exist $t_1 \in D_t$, $t_1 > t_0$, and an open set $N_x \subset D_x$ containing $\mathbf{x}_0$ such that $(1, \mathbf{f}(t, \mathbf{z}, \mathbf{q}(t, \mathbf{z}))) \in T_K(t, \mathbf{z})$ for all $(t, \mathbf{z}) \in K \cap N_0$ where $N_0 = [t_0, t_1) \times N_x$, then a solution of IVP (4.4) exists.*

Proof. By Proposition 4.4.1, $\widehat{\mathbf{q}}$ is continuous on F, and combined with the continuity of $\mathbf{b}$, $\mathbf{q}$ is continuous on K, and so $\mathbf{q}^{-1}(D_q)$ is open in K. By simple topological arguments, this means that $\mathbf{q}^{-1}(D_q) = K \cap N_1$ for some open $N_1 \subset \mathbb{R}^{1+n_x}$, and we can further assume that $N_1 \subset D_t \times D_x$. Note that $(t_0, \mathbf{x}_0) \in \mathbf{q}^{-1}(D_q) \subset N_1$.

Let $\widehat{K} = K \cap N_1 \cap N_0$. Note that $\widehat{K}$ is nonempty, since $(t_0, \mathbf{x}_0)$ is in each of K, N_1, N_0. More importantly, it is locally compact. To see this, first note that $[t_0, t_1)$ and N_x are locally compact, and so N_0 is also locally compact. Then, since $N_1 \cap N_0$ is an open subset of N_0, it too is locally compact. Since K is nonempty, then so is F, and by Proposition 4.2.1 F is closed. Finally, $\widehat{K} = K \cap N_1 \cap N_0 = \mathbf{b}^{-1}(F)$ is a closed subset of $N_1 \cap N_0$, since $\mathbf{b}$ is continuous on $N_1 \cap N_0$ and F is closed. As a closed subset of a locally compact space, $\widehat{K}$ is locally compact.

Note that $\mathbf{q}$ is defined, continuous, and takes values in D_q on $\widehat{K}$, and so $\mathbf{f}(\cdot, \cdot, \mathbf{q}(\cdot, \cdot))$ is defined and continuous on $\widehat{K}$. Now, let

$$\widehat{\mathbf{f}} : \widehat{K} \to \mathbb{R}^{1+n_x} : (t, \mathbf{z}) \mapsto (1, \mathbf{f}(t, \mathbf{z}, \mathbf{q}(t, \mathbf{z}))) .$$

By construction, $\widehat{\mathbf{f}}$ is continuous on $\widehat{K}$. Introduce the dummy variable s and formulate the initial value problem

$$\dot{\mathbf{y}}(s) = \widehat{\mathbf{f}}(\mathbf{y}(s)), \quad \mathbf{y}(s_0) = (t_0, \mathbf{x}_0). \tag{4.13}$$

The value of s is immaterial, so let $s_0 = t_0$. If there exists a solution $\mathbf{y}(s) = (t(s), \mathbf{x}(s))$ of (4.13), then it follows that $\frac{dt}{ds}(s) = 1$ and that $t(s) = s$. Furthermore,

$$\frac{d\mathbf{x}}{ds}(s) = \mathbf{f}(t(s), \mathbf{x}(s), \mathbf{q}(t(s), \mathbf{x}(s))) = \mathbf{f}(s, \mathbf{x}(s), \mathbf{q}(s, \mathbf{x}(s))).$$

Thus there would exist a corresponding solution of IVP (4.4).

Now, choose any $(t, \mathbf{z}) \in \widehat{K}$. Then $(t, \mathbf{z}) \in K \cap N_0$, and by assumption $\widehat{\mathbf{f}}(t, \mathbf{z}) \in T_K(t, \mathbf{z})$. So by Lemma 4.4.3, $\widehat{\mathbf{f}}(t, \mathbf{z}) \in T_{K \cap N_1}(t, \mathbf{z})$. Then, by Lemma 4.4.4, $\widehat{\mathbf{f}}(t, \mathbf{z}) \in T_{K \cap N_1 \cap N_0}(t, \mathbf{z})$. Thus $\widehat{\mathbf{f}}(t, \mathbf{z}) \in T_{\widehat{K}}(t, \mathbf{z})$ for any $(t, \mathbf{z}) \in \widehat{K}$. We say that $\widehat{K}$ is a *viability domain* for $\widehat{\mathbf{f}}$ (see Ch. 1 of [10]). Consequently, by the Nagumo Theorem, see for instance Theorem 1.2.1 of [10], a solution exists for (4.13), which corresponds to a solution of (4.4). □

If K is closed and we can establish *a priori* that a solution must exist, we could apply numerical integration methods tailored to this situation [10, §1.3], [161]. These methods enforce the fact the solution must remain in K by applying a projection operation to the solution estimates produced by some standard integration method.

As a first complication, this assumes more strict conditions on K (e.g. compactness, convexity, or an explicit representation in terms of inequality constraints is available). In the engineering-relevant examples in §4.6, this is certainly not the case. Second, Proposition 4.4.5 indicates that, in order to establish that a solution exists, we must verify the "tangency condition": $(1, \mathbf{f}(t, \mathbf{z}, \mathbf{q}(t, \mathbf{z}))) \in T_K(t, \mathbf{z})$, for all $(t, \mathbf{z}) \in K$. This condition, for all but the simplest cases, is completely abstract; K is defined in terms of a preimage, and the contingent cone typically does not admit a simple representation (the contingent cone of a set defined by smooth inequality constraints is one exception [11, §4.3.2]). In the case that $\mathbf{b}$ satisfies stronger regularity conditions, the condition could be recast in terms of the contingent cone of F, although this incurs its own complications like requiring the calculation of the preimage of the derivative of $\mathbf{b}$ (see Corollary 4.3.4 of [11]). And although we could attempt to verify the tangency condition with numerical information at a single point, this is complicated by the fact that it must hold on (at least) a *neighborhood* of the initial conditions. Consequently, for practical engineering applications, attempting to verify the tangency condition either numerically or analytically is extremely difficult.

4.5 Numerical developments

This section discusses the numerical method that has been developed for the efficient and reliable integration of ODEs with LP embedded. First, notation specific to this section and background from linear programming are introduced in §4.5.1. Then the overall numerical integration routine is introduced in §4.5.2. This method depends on a specific way to solve the lexicographic LP (4.1)-(4.2), which is described in §4.5.3.

4.5.1 Notation and background

The cardinality of a set J is $\operatorname{card}(J)$. Consider a vector $\mathbf{v} \in \mathbb{R}^n$ and a matrix $\mathbf{M} \in \mathbb{R}^{p \times n}$. Denote the j^{th} column of $\mathbf{M}$ by $\mathbf{M}_j$. For an index set $J = \{j_1, \dots, j_{n_J}\} \subset \{1, \dots, n\}$, let $\mathbf{v}_J = (v_{j_1}, \dots, v_{j_{n_J}})$ and similarly $\mathbf{M}_J = \begin{bmatrix} \mathbf{M}_{j_1} & \dots & \mathbf{M}_{j_{n_J}} \end{bmatrix}$. Similar notation applies to vectors and matrices that already have a subscript. For instance, $c_{i,j}$ is the j^{th} component of the vector $\mathbf{c}_i$, and for some index set $J \subset \{1, \dots, n_v\}$, $\mathbf{c}_{i,J}$ is the vector formed from the components of $\mathbf{c}_i$ corresponding to J. In Algorithm 2 matrices $\widehat{\mathbf{A}}_i \in \mathbb{R}^{m_i \times n_i}$ (for some (m_i, n_i)), will be constructed. It will be useful to think of their columns as indexed by some set P_i, with $\operatorname{card}(P_i) = n_i$, rather than $\{1, \dots, n_i\}$. Thus, for $j \in P_i$ and $J \subset P_i$, the "j^{th}" column of $\widehat{\mathbf{A}}_i$ is denoted $\widehat{\mathbf{A}}_{i,j}$, and $\widehat{\mathbf{A}}_{i,J}$ is the matrix formed from the columns of $\widehat{\mathbf{A}}_i$ corresponding to J.

The following linear programming background will be helpful, which draws freely from the first four chapters of [25]. Consider the first-level LP as a prototype for standard-form LPs parameterized by the right-hand side of the constraints:

$$\widehat{q}_1(\mathbf{d}) = \inf\left\{\mathbf{c}_1^{\mathrm{T}}\mathbf{v} : \mathbf{v} \in \mathbb{R}^{n_v}, \mathbf{A}\mathbf{v} = \mathbf{d}, \mathbf{v} \geq \mathbf{0}\right\}. \tag{4.14}$$

The following assumption will hold in this and subsequent sections. It is a standard assumption of the simplex method, upon which the proposed numerical developments are based.

Assumption 4.5.1. *The matrix* $\mathbf{A}$ *is full row rank.*

The concept of a basis is introduced. A basis B is a subset of $\{1, \dots, n_v\}$ with $m = \mathrm{card}(B)$. An optimal basis is one which satisfies

$$\mathbf{A}_B^{-1}\mathbf{d} \geq \mathbf{0}, \tag{4.15}$$

$$\mathbf{c}_1^{\mathrm{T}} - \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{A} \geq \mathbf{0}^{\mathrm{T}}. \tag{4.16}$$

A basis which satisfies (4.15) is primal feasible, while one that satisfies (4.16) is dual feasible. Thus, a basis is optimal if and only if it is primal and dual feasible. The invertible matrix $\mathbf{A}_B$ is the corresponding basis matrix. A basis also serves to describe a vector $\mathbf{v} \in \mathbb{R}^{n_v}$; the components of the vector corresponding to B, $\mathbf{v}_B$, are given by $\mathbf{v}_B = \mathbf{A}_B^{-1}\mathbf{d}$, and the rest are zero, i.e. $v_j = 0$, $j \notin B$. If the basis B is optimal, then the vector $\mathbf{v}$ which it describes is in the optimal solution set of the first-level LP. Thus, $\widehat{q}_1(\mathbf{d}) = \mathbf{c}_1^{\mathrm{T}}\mathbf{v} = \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{v}_B$. The variables $\mathbf{v}_B$ are called the basic variables. The vector $\mathbf{c}_1^{\mathrm{T}} - \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{A}$ is the vector of reduced costs. It is clear that perturbations in $\mathbf{d}$ do not affect dual feasibility of a basis. Thus, a basis is optimal for all $\mathbf{d}$ such that the basic variables are nonnegative. As a basic observation, the existence of either a primal or dual feasible basis implies that $\mathbf{A}$ is full row rank (m columns are linearly independent, which implies that the column rank is at least m, which implies that $\mathbf{A}$ is full row rank).

4.5.2 Solution algorithm

Theorem 4.4.1, Corollary 4.4.2, and Proposition 4.4.2 indicate how we should approach calculating a solution of IVP (4.4): solve the extended IVP (4.10) and detect the earliest time that the solution trajectory leaves K, indicated by the infeasibility of the embedded

LP at a point on the solution trajectory. In general terms, this is the approach taken in the following numerical method. Under the assumptions of Proposition 4.4.2, broad classes of numerical integration methods are convergent for the extended IVP (4.10), including linear multistep and Runge-Kutta methods [105]. However, there is still the issue that we need to detect the earliest time that the solution trajectory leaves K accurately and reliably. As indicated by the examples in §4.3, we cannot merely rely on detecting an infeasible embedded LP, as this could occur during a corrector iteration, for instance. The following method addresses these issues.

The essence of the method is easily understood when $n_q = 1$, in which case the dynamics only depend on the optimal objective value of a single LP parameterized by its right-hand side. If we solve the embedded LP at the initial conditions with any method which finds an optimal basis B, then for as long as B is optimal, we can obtain the optimal basic variables by solving the system $\mathbf{A}_B\mathbf{u}_B(t) = \mathbf{b}(t, \mathbf{x}(t))$ for $\mathbf{u}_B(t)$, from which we obtain $q_1(t, \mathbf{x}(t)) = \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{u}_B(t)$. Meanwhile, B is optimal for as long as the basic variables are nonnegative, i.e. $\mathbf{u}_B(t) \geq \mathbf{0}$. Consequently, the general idea is to reformulate the system as DAEs, where the basic variables $\mathbf{u}_B$ have been added as algebraic variables, and employ event detection to detect when the value of a basic variable crosses zero. Once a basic variable crosses zero, a new optimal basis is found by re-solving the LP, and the procedure is repeated.

For the time being suppose that a δ-optimal basis B is acceptable; that is to say that $\mathbf{A}_B^{-1}\mathbf{b}(t, \mathbf{x}(t)) > -\delta\mathbf{1}$. To guarantee the detection of when B ceases to be δ-optimal, we need to use a feasibility tolerance $\epsilon < \delta$ when solving the embedded LP. Then, the initial values of the basic variables satisfy

$$\mathbf{u}_B(t_0) \geq -\epsilon\mathbf{1} > -\delta\mathbf{1}.$$

Consequently, $\mathbf{u}_B(t_0) + \delta\mathbf{1}$ is strictly positive. If B ceases to be δ-optimal, then for some index j, the value $u_j(t) + \delta$ will cross zero, which can be detected quite accurately with event detection algorithms [145]. The following DAEs, while $\mathbf{u}_B(t) > -\delta\mathbf{1}$, are integrated numerically:

$$\begin{aligned}\dot{\mathbf{x}}(t) - \mathbf{f}(t, \mathbf{x}(t), \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{u}_B(t)) &= \mathbf{0},\\ \mathbf{A}_B\mathbf{u}_B(t) - \mathbf{b}(t, \mathbf{x}(t)) &= \mathbf{0}.\end{aligned}$$

Since $\mathbf{A}_B$ is nonsingular, it is clear from inspection that this is a semi-explicit index-one system of DAEs, and amenable to many numerical integration methods.

Of course, ϵ is a small, but positive, number, and so δ must be as well. Consequently, we have to ask whether it actually is acceptable for the basis B to be merely δ-optimal. Since the goal is to calculate a solution of the extended IVP, we need to ensure that for a δ-optimal basis B, $\widehat{q}_1^B(\mathbf{d}) = \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{d}$ is an accurate approximation of $\widehat{q}_1^E(\mathbf{d})$. Indeed it is. For a dual feasible basis B, let $F_B = \{\mathbf{d} \in \mathbb{R}^m : \mathbf{A}_B^{-1}\mathbf{d} \geq \mathbf{0}\}$, thus F_B is the subset of F on which B is primal feasible and so also optimal. Let $F_{B,\delta} = \{\mathbf{d} \in \mathbb{R}^m : \mathbf{A}_B^{-1}\mathbf{d} \geq -\delta\mathbf{1}\}$, thus $F_{B,\delta}$ is the set on which B is δ-optimal. Now assume $\mathbf{d} \in F_{B,\delta}$ and let $\mathbf{v} = \mathbf{A}_B^{-1}\mathbf{d}$. Construct $\widetilde{\mathbf{v}}$ such that $\widetilde{v}_i = \max\{v_i, 0\}$, thus $\widetilde{\mathbf{v}} \geq \mathbf{0}$. Let $\widetilde{\mathbf{d}} = \mathbf{A}_B\widetilde{\mathbf{v}} \in F_B$. Note that $\|\mathbf{v} - \widetilde{\mathbf{v}}\|_\infty \leq \delta$, thus $\left\|\mathbf{d} - \widetilde{\mathbf{d}}\right\|_\infty \leq \|\mathbf{A}_B\|_\infty \delta$. Since $\widehat{q}_1^B = \widehat{q}_1^E$ on F_B, $\widehat{q}_1^B(\widetilde{\mathbf{d}}) = \widehat{q}_1^E(\widetilde{\mathbf{d}})$. Consequently,

$$\begin{aligned}\left|\widehat{q}_1^B(\mathbf{d}) - \widehat{q}_1^E(\mathbf{d})\right| &\leq \left|\widehat{q}_1^B(\mathbf{d}) - \widehat{q}_1^B(\widetilde{\mathbf{d}})\right| + \left|\widehat{q}_1^B(\widetilde{\mathbf{d}}) - \widehat{q}_1^E(\mathbf{d})\right|\\ &= \left|\widehat{q}_1^B(\mathbf{d}) - \widehat{q}_1^B(\widetilde{\mathbf{d}})\right| + \left|\widehat{q}_1^E(\widetilde{\mathbf{d}}) - \widehat{q}_1^E(\mathbf{d})\right|\\ &\leq \|\mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\|_2 \left\|\mathbf{d} - \widetilde{\mathbf{d}}\right\|_2 + L_q \left\|\mathbf{d} - \widetilde{\mathbf{d}}\right\|_2\\ &\leq M\delta,\end{aligned}$$

where the Lipschitz continuity of $\widehat{q}_1^E$ and the equivalence of norms have been used. Note that M is finite and can be chosen so that the inequality holds for any choice of B, since there are a finite number of dual feasible bases. Thus, the error in approximating $\widehat{q}_1^E$ using a δ-optimal basis must go to zero as δ goes to zero.

The failure to find a δ-optimal basis at a particular value of $\mathbf{b}(t, \mathbf{x}(t))$ simply implies that $(t, \mathbf{x}(t)) \notin K$. If a δ-optimal basis does not exist, then certainly an optimal basis does not exist, which means that $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ implying $(t, \mathbf{x}(t)) \notin K$, and so by Corollary 4.4.2, the calculated solution is no longer a solution of IVP (4.4). However, since the test of whether $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ is only performed as part of the determination of a new optimal basis, after the old one has stopped being δ-optimal, this is a much more reliable indication that the solution cannot be continued.

To generalize this method to the case $n_q > 1$ the overall structure remains unchanged. This is because it is possible to find a basis B which is optimal for the first-level LP and which describes a point which is in the optimal solution set of the i^{th}-level LP, for *all* i. Then $\widehat{\mathbf{q}}(\mathbf{d}) = (\mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{d}, \ldots, \mathbf{c}_{n_q,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{d})$ for all $\mathbf{d}$ such that B is optimal for the first-level LP. This idea is proved in Theorem 4.5.1 and the method for determining the appropriate basis is summarized in Algorithm 2, both presented in §4.5.3.

The numerical method in the general case is summarized in Algorithm 1. An empty basis set returned by Algorithm 2 serves as a flag that $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ and that the solution cannot be continued. The convergence of Algorithm 1 (as δ and step size tend to zero) is guaranteed if the numerical method used to integrate the DAE system (4.17) is convergent for the extended IVP (4.10), which, as mentioned earlier, includes broad classes under the assumptions of Proposition 4.4.2. This follows from simple arguments for the convergence of methods for semi-explicit index-one DAEs; see for instance §3.2.1 of [34]. Overall, Algorithm 1 produces an approximation of the solution of the extended IVP, and gives a reliable and accurate indication of when this solution is no longer a solution of the original IVP (4.4).

Algorithm 1 Overall solution method for the IVP (4.4)

```
Require: $\delta > \epsilon > 0$, $t_f > t_0$
  $\widetilde{t} \leftarrow t_0$, $\widetilde{\mathbf{x}} \leftarrow \mathbf{x}_0$
  loop
    $B \leftarrow B^*(\mathbf{b}(\widetilde{t}, \widetilde{\mathbf{x}}), \epsilon)$ (See Algorithm 2)
    if $B = \varnothing$ then
      Terminate.
    end if
    Solve $\mathbf{A}_B\widetilde{\mathbf{u}}_B = \mathbf{b}(\widetilde{t}, \widetilde{\mathbf{x}})$ for $\widetilde{\mathbf{u}}_B$.
    Set $\mathbf{q}^B : \mathbf{u} \mapsto (\mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{u}, \ldots, \mathbf{c}_{n_q,B}^{\mathrm{T}}\mathbf{u})$.
    while $\widetilde{\mathbf{u}}_B > -\delta\mathbf{1}$ do
      Update $(\widetilde{t}, \widetilde{\mathbf{x}}, \widetilde{\mathbf{u}}_B)$ by integrating the following DAE system with an appropriate
      method:
          $$\begin{aligned} \dot{\mathbf{x}}(t) - \mathbf{f}(t, \mathbf{x}(t), \mathbf{q}^B(\mathbf{u}_B(t))) &= \mathbf{0}, \\ \mathbf{A}_B\mathbf{u}_B(t) - \mathbf{b}(t, \mathbf{x}(t)) &= \mathbf{0}. \end{aligned} \tag{4.17}$$
      if $\widetilde{t} \geq t_f$ then
        Terminate.
      end if
    end while
  end loop
```

An implementation of Algorithm 1 has been coded incorporating DAEPACK [200] component DSL48E for the numerical integration of the DAE and event detection. DSL48E uses a BDF method and the sparse unstructured linear algebra code MA48 [47], and so is appropriate for the numerical integration of stiff systems; these features will be indispensable in the solution of DFBA models in §4.6. Meanwhile, the event detection algorithm is an accurate and efficient method developed in [145]. A code employing CPLEX version 12.4 [85] implements Algorithm 2. This implementation of the algorithms has been named DSL48LPR.

4.5.3 Lexicographic optimization

An inefficient way to try to generalize the basic idea behind Algorithm 1 to $n_q > 1$ would be to calculate an optimal basis for each level LP, disregarding the connections between the levels.

However, by exploiting the relationship between the individual levels in the lexicographic LP, it in fact suffices to determine a *single* optimal basis for the first-level LP (4.1) to calculate some element of the solution set of the i^{th}-level LP for each i. Theorem 4.5.1 formalizes this and its proof provides a constructive method of finding the appropriate basis. The construction is summarized in Algorithm 2.

The benefit of Algorithm 2 is that it allows us to use standard primal simplex. That is, any pivot selection rules can be used, and so we can rely on a commercial implementation of primal simplex to implement Algorithm 2, and then degeneracy and cycling are not a concern. Modifications of the simplex algorithm ("lexicographic simplex") have been presented in [86, Ch. 3], [186, §10.5], and [87, 101, 148] to solve lexicographic LPs. These methods are similar in effect to Algorithm 2. In contrast, these methods either do not consider the parametric results needed here, require specific pivot selection rules, or do not consider degeneracy or cycling.

Theorem 4.5.1. *Assume that* $\mathbf{d} \in F$. *Then there exists a basis* B_1^* *that is optimal for the first-level LP* (4.1) *and*

$$\widehat{\mathbf{q}}(\mathbf{d}) = \left(\mathbf{c}_{1,B_1^*}^{\mathrm{T}} \mathbf{A}_{B_1^*}^{-1} \mathbf{d}, \dots, \mathbf{c}_{n_q,B_1^*}^{\mathrm{T}} \mathbf{A}_{B_1^*}^{-1} \mathbf{d}\right). \tag{4.18}$$

Further, this relation holds for all $\mathbf{d}$ *such that* B_1^* *is optimal for the first-level LP.*

Algorithm 2 Method for determining optimal basis for lexicographic LP (4.1)-(4.2)

```
Require: $\mathbf{d} \in \mathbb{R}^m$, $\epsilon > 0$
  $P_1 \leftarrow \{1, \ldots, n_v\}$
  $n_1 \leftarrow n_v$, $N_1 \leftarrow \varnothing$
  $\widehat{\mathbf{A}}_1 \leftarrow \mathbf{A}$, $\mathbf{d}_1 \leftarrow \mathbf{d}$
  Solve first-level LP with absolute feasibility tolerance $\epsilon$:
      $$q_1^* = \inf\{\mathbf{c}_1^{\mathrm{T}}\mathbf{v} : \mathbf{A}\mathbf{v} = \mathbf{d}, \mathbf{v} \geq \mathbf{0}, \mathbf{v} \in \mathbb{R}^{n_v}\}.$$
  if $-\infty < q_1^* < +\infty$ then
    Determine optimal basis $B_1$ for first-level LP.
  else
    return $B^*(\mathbf{d}, \epsilon) \leftarrow \varnothing$
  end if
  $i \leftarrow 1$
  while $i < n_q$ do
    if $c_{i,j} - \mathbf{c}_{i,B_i}^{\mathrm{T}}\widehat{\mathbf{A}}_{i,B_i}^{-1}\widehat{\mathbf{A}}_{i,j} > 0$, $\forall j \in P_i \backslash B_i$ then
      return $B^*(\mathbf{d}, \epsilon) \leftarrow B_1$
    end if
    if $\mathbf{c}_{i,P_i}^{\mathrm{T}} - \mathbf{c}_{i,B_i}^{\mathrm{T}}\widehat{\mathbf{A}}_{i,B_i}^{-1}\widehat{\mathbf{A}}_i = \mathbf{0}^{\mathrm{T}}$ then
      $P_{i+1} \leftarrow P_i$
      $n_{i+1} \leftarrow n_i$, $N_{i+1} \leftarrow N_i$
      $\widehat{\mathbf{A}}_{i+1} \leftarrow \widehat{\mathbf{A}}_i$, $\mathbf{d}_{i+1} \leftarrow \mathbf{d}_i$
    else
      Choose $j \in P_i$ such that $c_{i,j} - \mathbf{c}_{i,B_i}^{\mathrm{T}}\widehat{\mathbf{A}}_{i,B_i}^{-1}\widehat{\mathbf{A}}_{i,j} > 0$.
      $P_{i+1} = \left\{k \in P_i : c_{i,k} - \mathbf{c}_{i,B_i}^{\mathrm{T}}\widehat{\mathbf{A}}_{i,B_i}^{-1}\widehat{\mathbf{A}}_{i,k} = 0\right\} \cup \{j\}$
      $n_{i+1} \leftarrow \mathrm{card}(P_{i+1})$, $N_{i+1} \leftarrow N_i \cup \{j\}$
      $\widehat{\mathbf{A}}_{i+1} \leftarrow \begin{bmatrix} \mathbf{c}_{i,P_{i+1}}^{\mathrm{T}} \\ \widehat{\mathbf{A}}_{i,P_{i+1}} \end{bmatrix}$, $\mathbf{d}_{i+1} \leftarrow \begin{bmatrix} q_i^* \\ \mathbf{d}_i \end{bmatrix}$
    end if
    Solve $(i+1)^{th}$ projected LP with primal simplex using initial basis $B_1 \cup N_{i+1}$ and absolute feasibility tolerance $\epsilon$:
        $$q_{i+1}^* = \inf_{\mathbf{v} \in \mathbb{R}^{n_{i+1}}} \mathbf{c}_{i+1,P_{i+1}}^{\mathrm{T}}\mathbf{v}$$
        $$\text{s.t. } \widehat{\mathbf{A}}_{i+1}\mathbf{v} = \mathbf{d}_{i+1},$$
        $$\mathbf{v} \geq \mathbf{0}.$$
    if $-\infty < q_{i+1}^* < +\infty$ then
      For $(i+1)^{th}$ projected LP, optimal basis is $B_{i+1} = \widetilde{B}_1 \cup N_{i+1}$.
      $B_1 \leftarrow \widetilde{B}_1$
    else
      return $B^*(\mathbf{d}, \epsilon) \leftarrow \varnothing$
    end if
    $i \leftarrow i + 1$
    $B^*(\mathbf{d}, \epsilon) \leftarrow B_1$
  end while
  return $B^*(\mathbf{d}, \epsilon)$
```

Supporting results

This section presents some definitions, results, and discussion to support the proof of Theorem 4.5.1 in the following section, which deals with finding a specific optimal basis for the lexicographic LP.

Definition 4.5.1. Equivalence.

1. Let $(n_1, n_2) \in \mathbb{N}^2$ with $n_1 \leq n_2$. Two sets $S_1 \subset \mathbb{R}^{n_1}$ and $S_2 \subset \mathbb{R}^{n_2}$ are equivalent if $n_1 < n_2$ and $S_2 = S_1 \times \{\mathbf{0}\}$, or $n_1 = n_2$ and $S_2 = S_1$.
2. Two linear programs are equivalent if their solution sets are equivalent.

Intuitive results regarding equivalence follow.

Lemma 4.5.1. *Let $(n_1, n_2, n_3) \in \mathbb{N}^3$ with $n_1 \leq n_2 \leq n_3$.*

1. Let $F_i \subset \mathbb{R}^{n_i}$ for $i \in \{1, 2, 3\}$. If sets F_1 and F_2 are equivalent and F_2 and F_3 are equivalent, then F_1 and F_3 are equivalent.

2. If two sets $F_1 \in \mathbb{R}^{n_1}$ and $F_2 \in \mathbb{R}^{n_2}$ are equivalent, then for any $\mathbf{c} \in \mathbb{R}^{n_1}$ and $\widetilde{\mathbf{c}} \in \mathbb{R}^{n_2 - n_1}$ the linear programs

$$\min\{\mathbf{c}^{\mathrm{T}}\mathbf{v} : \mathbf{v} \in F_1\} \quad \textit{and} \quad \min\{\widehat{\mathbf{c}}^{\mathrm{T}}\mathbf{v} : \mathbf{v} \in F_2\}$$

are equivalent, where $\widehat{\mathbf{c}} = (\mathbf{c}, \widetilde{\mathbf{c}})$ (with the claim being trivial if $n_1 = n_2$).

For the next two results refer to the lexicographic LP

$$q(\mathbf{d}) = \inf\left\{\mathbf{c}^{\mathrm{T}}\mathbf{v} : \mathbf{M}\mathbf{v} = \mathbf{d}, \mathbf{v} \geq \mathbf{0}\right\}, \tag{4.19}$$

$$\widehat{q}(\mathbf{d}) = \inf\left\{\widehat{\mathbf{c}}^{\mathrm{T}}\mathbf{v} : \mathbf{M}\mathbf{v} = \mathbf{d}, \mathbf{c}^{\mathrm{T}}\mathbf{v} = q(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\right\}. \tag{4.20}$$

The next result establishes the form of the simplex tableau for the two-level lexicographic LP. Strictly speaking, tableau (4.21) below is missing the "zeroth" row of reduced costs for the second-level LP (4.20); for simplicity it is omitted.

Lemma 4.5.2. *Consider the lexicographic LP* (4.19)-(4.20). *Let B be a dual feasible basis for the first-level LP* (4.19), *and assume that the j^{th} reduced cost is positive ($c_j - \mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{M}_j > 0$). For all $\mathbf{d}$ such that B is optimal for the first-level LP, the simplex tableau for the* *second-level LP* (4.20) *resulting from the basis* $\widehat{B} = \{j\} \cup B$ *is*

$$\begin{bmatrix} c_j & \mathbf{c}_B^{\mathrm{T}} \\ \mathbf{M}_j & \mathbf{M}_B \end{bmatrix}^{-1} \begin{bmatrix} q(\mathbf{d}) & \mathbf{c}^{\mathrm{T}} \\ \mathbf{d} & \mathbf{M} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\mathbf{c}^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}} \mathbf{M}_B^{-1} \mathbf{M}}{c_j - \mathbf{c}_B^{\mathrm{T}} \mathbf{M}_B^{-1} \mathbf{M}_j} \\ \mathbf{M}_B^{-1} \mathbf{d} & \mathbf{M}_B^{-1} \left(\mathbf{M} - \mathbf{M}_j \frac{\mathbf{c}^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}} \mathbf{M}_B^{-1} \mathbf{M}}{c_j - \mathbf{c}_B^{\mathrm{T}} \mathbf{M}_B^{-1} \mathbf{M}_j} \right) \end{bmatrix}. \tag{4.21}$$

Proof. The proof proceeds by using Schur complements to form the inverse of $\begin{bmatrix} c_j & \mathbf{c}_B^{\mathrm{T}} \\ \mathbf{M}_j & \mathbf{M}_B \end{bmatrix}$, performing the matrix multiplication, and simplifying, noting that $\mathbf{c}_B^{\mathrm{T}} \mathbf{M}_B^{-1} \mathbf{d} = q(\mathbf{d})$ for all $\mathbf{d}$ such that $\mathbf{M}_B^{-1} \mathbf{d} \geq \mathbf{0}$. □

The concept of a “null variable” is important to the proof of Theorem 4.5.1, which is defined as a variable which is zero everywhere in the feasible set of an LP. It is clear that removing a null variable and the corresponding parameters (components of the cost vector and columns of the constraint matrix) yields an equivalent LP. The next result states a way to identify null variables in the second-level LP (4.20).

Lemma 4.5.3. *Consider the lexicographic LP* (4.19)-(4.20). *Let* B *be a dual feasible basis for the first-level LP* (4.19), *and assume that the* j^{th} *reduced cost is positive* ($c_j - \mathbf{c}_B^{\mathrm{T}} \mathbf{M}_B^{-1} \mathbf{M}_j > 0$). *For all* $\mathbf{d}$ *such that* B *is optimal for the first-level LP and for all* $\mathbf{v}$ *feasible in the second-level LP* (4.20), $v_j = 0$.

Proof. The result follows from the “null variable theorem” in §4.7 of [111]; this states that v_j is a null variable for a general standard-form LP (4.19) if and only if there exists a nonzero $\mathbf{p}$ such that $\mathbf{p}^{\mathrm{T}} \mathbf{d} = 0$, $\mathbf{p}^{\mathrm{T}} \mathbf{M} \geq \mathbf{0}^{\mathrm{T}}$, and the j^{th} component of $\mathbf{p}^{\mathrm{T}} \mathbf{M}$ is strictly greater than zero. Applying this result to the second-level LP (4.20), the result follows from inspection of the tableau (4.21); the first row of $\begin{bmatrix} c_j & \mathbf{c}_B^{\mathrm{T}} \\ \mathbf{M}_j & \mathbf{M}_B \end{bmatrix}^{-1}$ serves as the appropriate $\mathbf{p}$. □

Finally, some aspects of the primal simplex algorithm are noted. If we have an optimal basis already, but a different optimal basis is sought, a pivot could be forced in the sense that, while the i^{th} reduced cost is zero, the i^{th} column is chosen as the pivot column. If the i^{th} reduced cost is zero, but the pivot operation is carried out in the standard way, a new primal feasible basis is obtained for which the reduced costs are the same as the old basis, and so the new basis is also optimal. This is because the reduced costs are updated in a pivot operation by adding a multiple of the pivot row to the reduced costs (the zeroth row of the tableau) so that the i^{th} entry of the zeroth row is zero. But, if the i^{th} reduced cost is already zero, no changes to the zeroth row are made, and so the reduced costs retain the same values.

Proof of Theorem 4.5.1

Proof. Existence and construction of the appropriate basis proceed by induction; again, the construction is summarized in Algorithm 2. At each induction step a special "projected" LP is constructed and optimized. The reason behind considering this projected LP is that we can draw conclusions about the pivots taken when optimizing it with primal simplex. This allows us to argue about the form of the optimal basis.

First introduce some specific notation. For some index set J and a matrix $\mathbf{M}$, the matrix $\mathbf{M}^J$ is the matrix equaling $\mathbf{M}$ with those columns corresponding to J set to $\mathbf{0}$.

Fix $\mathbf{d} \in F$ to the value of interest. For an induction basis, let B_1 be any optimal basis for the first-level LP (4.1) (which must exist by Assumption 4.5.1 and since $\widehat{q}_1(\mathbf{d})$ is finite, see [25, §3.4]), $n_1 = n_v$, $m_1 = m$, $P_1 = \{1, \dots, n_v\}$, $N_1 = \varnothing$, $\widehat{\mathbf{A}}_1 = \mathbf{A}$ and $\mathbf{d}_1 : \mathbf{d}' \mapsto \mathbf{d}'$. An optimal tableau for the first-level LP is

$$\widehat{\mathbf{A}}_{1,B_1}^{-1}\left[\mathbf{d}_1(\mathbf{d}) \quad \widehat{\mathbf{A}}_1\right] = \left[\mathbf{A}_{B_1}^{-1}\mathbf{d} \quad \mathbf{A}_{B_1}^{-1}\left(\mathbf{A}_{P_1}^{N_1}\right)\right].$$

For the i^{th} induction step assume the following:

1. Assume for $k \in \{2, \dots, i\}$, $n_{k-1} \geq n_k$, $m_{k-1} \leq m_k$, $N_{k-1} \subset N_k$, and for $k \in \{1, \dots, i\}$, $P_k = \{1, \dots, n_k\}$, $N_k \subset P_k$, $\widehat{\mathbf{A}}_k \in \mathbb{R}^{m_k \times n_k}$ and $\mathbf{d}_k : F \to \mathbb{R}^{m_k}$. Consider the k^{th} "projected" LP, for $k \in \{1, \dots, i\}$

$$\begin{aligned} q_k^P(\mathbf{d}) = \min_{\mathbf{v} \in \mathbb{R}^{n_k}} \ & \mathbf{c}_{k,P_k}^{\mathrm{T}}\mathbf{v} \\ \text{s.t. } & \widehat{\mathbf{A}}_k\mathbf{v} = \mathbf{d}_k(\mathbf{d}), \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{4.22}$$

2. Assume that the i^{th}-level LP (4.2) is equivalent to the i^{th} projected LP in the sense of Definition 4.5.1.

3. Assume the bases B_1, and for $k \in \{2, \dots, i\}$, $B_k = N_k \cup B_1$ are optimal for the first-level and k^{th} projected LPs, respectively. Also assume that for $k \in \{1, \dots, i-1\}$, $c_{k,j} - \mathbf{c}_{k,B_k}^{\mathrm{T}}\widehat{\mathbf{A}}_{k,B_k}^{-1}\widehat{\mathbf{A}}_{k,j} > 0$ for each $j \in (P_k \backslash P_{k+1}) \cup (N_{k+1} \backslash N_k)$, and $c_{k,j} - \mathbf{c}_{k,B_k}^{\mathrm{T}}\widehat{\mathbf{A}}_{k,B_k}^{-1}\widehat{\mathbf{A}}_{k,j} = 0$

for each $j \in P_i \backslash B_i$.

4. Assume that the tableau for the i^{th} projected LP resulting from the basis B_i is

$$\widehat{\mathbf{A}}_{i,B_i}^{-1} \left[\mathbf{d}_i(\mathbf{d}) \quad \widehat{\mathbf{A}}_i\right] = \begin{bmatrix} \mathbf{0} & \mathbf{E}_i \\ \mathbf{A}_{B_1}^{-1}\mathbf{d} & \mathbf{A}_{B_1}^{-1}\left(\mathbf{A}_{P_i}^{N_i}\right) \end{bmatrix},$$

where $\mathbf{E}_i$ is a $(m_i - m_1) \times n_i$ matrix constructed from the rows of the $n_i \times n_i$ identity matrix that correspond to elements of N_i. Recall that the left-most column of the above tableau is typically called the "zeroth" column.

There are three cases when constructing the next LP. In the first case, consider the reduced costs for the i^{th} projected LP determined from the basis B_i from induction assumption 3. If each reduced cost corresponding to a nonbasic variable is positive (i.e. for all $j \in P_i \backslash B_i$, $c_{i,j} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \widehat{\mathbf{A}}_{i,B_i}^{-1} \widehat{\mathbf{A}}_{i,j} > 0$), then the point described by the basis B_i is the unique optimal solution point for the i^{th} projected LP [25, §3.9]. By induction assumption 2 (equivalence), the solution set of the i^{th}-level LP is also a singleton; let this point be $\mathbf{v}^* \in \mathbb{R}^{n_v}$. Combined with induction assumption 4, the only nonzero components of $\mathbf{v}^*$ are those corresponding to B_1, so we have $\mathbf{c}_1^{\mathrm{T}}\mathbf{v}^* = \mathbf{c}_{1,B_1}^{\mathrm{T}} \mathbf{A}_{B_1}^{-1}\mathbf{d}$. Of course, by the nature of the lexicographic LP, $\mathbf{v}^*$ must be an optimal solution point of the k^{th}-level LP, for all $k \in \{1, \dots, n_q\}$, and so letting $B_1^* = B_1$ we have that Eqn. (4.18) holds.

For the other two cases, a higher-level LP must be considered. Our aim is to construct the $(i+1)^{th}$ projected LP

$$\begin{aligned} q_{i+1}^P(\mathbf{d}) = \min_{\mathbf{v} \in \mathbb{R}^{n_{i+1}}} \; & \mathbf{c}_{i+1,P_{i+1}}^{\mathrm{T}} \mathbf{v} \\ \text{s.t. } & \widehat{\mathbf{A}}_{i+1}\mathbf{v} = \mathbf{d}_{i+1}(\mathbf{d}), \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{4.23}$$

In the second case, if $\mathbf{c}_{i,P_i}^{\mathrm{T}} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \widehat{\mathbf{A}}_{i,B_i}^{-1} \widehat{\mathbf{A}}_i = \mathbf{0}^{\mathrm{T}}$, then $\mathbf{c}_{i,P_i}^{\mathrm{T}}$ and the rows of $\widehat{\mathbf{A}}_i$ are linearly dependent, and so the constraint $\mathbf{c}_{i,P_i}^{\mathrm{T}}\mathbf{v} = q_i^P(\mathbf{d})$ is redundant (it is satisfied everywhere in the feasible set of the i^{th} projected LP). Let $n_{i+1} = n_i$, $m_{i+1} = m_i$, $P_{i+1} = P_i$, $N_{i+1} = N_i$, $\widehat{\mathbf{A}}_{i+1} = \widehat{\mathbf{A}}_i$ and $\mathbf{d}_{i+1} = \mathbf{d}_i$. The basis $B_{i+1} = B_i$ is primal feasible for the $(i+1)^{th}$ projected LP. To help establish that induction assumption 3 will hold for the $(i+1)^{th}$ step, note that we trivially have $c_{i,j} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \widehat{\mathbf{A}}_{i,B_i}^{-1} \widehat{\mathbf{A}}_{i,j} > 0$ for each $j \in (P_i \backslash P_{i+1}) \cup (N_{i+1} \backslash N_i)$, and

$c_{i,j} - \mathbf{c}^{\mathrm{T}}_{i,B_i} \widehat{\mathbf{A}}^{-1}_{i,B_i} \widehat{\mathbf{A}}_{i,j} = 0$ for each $j \in P_{i+1} \backslash B_{i+1}$. The resulting tableau is the same form as in induction assumption 4. Further, the feasible set of the $(i+1)^{th}$ projected LP is the solution set of the i^{th} projected LP; by induction assumption 2 (equivalence) and Lemma 4.5.1, we have that the $(i+1)^{th}$ projected LP is equivalent to the $(i+1)^{th}$-level LP.

In the third case, if there is a $j \in P_i$ such that $c_{i,j} - \mathbf{c}^{\mathrm{T}}_{i,B_i} \widehat{\mathbf{A}}^{-1}_{i,B_i} \widehat{\mathbf{A}}_{i,j} > 0$, then let

$$P_{i+1} = \left\{ k \in P_i : c_{i,k} - \mathbf{c}^{\mathrm{T}}_{i,B_i} \widehat{\mathbf{A}}^{-1}_{i,B_i} \widehat{\mathbf{A}}_{i,k} = 0 \right\} \cup \{j\}.$$

Let n_{i+1} be the number of elements in P_{i+1} and assume without loss of generality that $P_{i+1} = \{1, \dots, n_{i+1}\}$ (the variables could be re-ordered as necessary). Let $m_{i+1} = m_i + 1$, $N_{i+1} = \{j\} \cup N_i$ and $B_{i+1} = N_{i+1} \cup B_1$. Note that $B_{i+1} = \{j\} \cup N_i \cup B_1$, and since $B_i = N_i \cup B_1$, we have $B_{i+1} = \{j\} \cup B_i$. From Lemma 4.5.2, we have that B_{i+1} is primal feasible for the $(i+1)^{th}$ projected LP. Since the basic variables of the i^{th} projected LP have corresponding reduced costs that are zero, from the definition of P_{i+1} we have $B_{i+1} \subset P_{i+1}$ so this is a well-defined basis. To help establish that induction assumption 3 will hold for the $(i+1)^{th}$ step, note that by construction of P_{i+1}, B_{i+1}, and N_{i+1}, the k^{th} reduced cost of the i^{th} projected LP is positive for all $k \in (P_i \backslash P_{i+1}) \cup (N_{i+1} \backslash N_i)$, and the k^{th} reduced cost is zero for all $k \in P_{i+1} \backslash B_{i+1}$. Let

$$\widehat{\mathbf{A}}_{i+1} = \begin{bmatrix} \mathbf{c}^{\mathrm{T}}_{i,P_{i+1}} \\ \widehat{\mathbf{A}}_{i,P_{i+1}} \end{bmatrix} \quad \text{and} \quad \mathbf{d}_{i+1} : \mathbf{d}' \mapsto \begin{bmatrix} q^P_i(\mathbf{d}') \\ \mathbf{d}_i(\mathbf{d}') \end{bmatrix}.$$

By the construction of the index set P_{i+1}, we have that

$$\frac{\mathbf{c}^{\mathrm{T}}_{i,P_{i+1}} - \mathbf{c}^{\mathrm{T}}_{i,B_i} \widehat{\mathbf{A}}^{-1}_{i,B_i} \widehat{\mathbf{A}}_{i,P_{i+1}}}{c_{i,j} - \mathbf{c}^{\mathrm{T}}_{i,B_i} \widehat{\mathbf{A}}^{-1}_{i,B_i} \widehat{\mathbf{A}}_{i,j}}$$

is the j^{th} unit vector in $\mathbb{R}^{n_{i+1}}$ (denoted $\mathbf{e}^{\mathrm{T}}_j$), and so by Lemma 4.5.2, the resulting tableau for the $(i+1)^{th}$ projected LP is

$$\begin{bmatrix} 0 & \mathbf{e}^{\mathrm{T}}_j \\ \widehat{\mathbf{A}}^{-1}_{i,B_i} \mathbf{d}_i(\mathbf{d}) & \widehat{\mathbf{A}}^{-1}_{i,B_i} \left(\widehat{\mathbf{A}}_{i,P_{i+1}} - \widehat{\mathbf{A}}_{i,j} \mathbf{e}^{\mathrm{T}}_j \right) \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{e}^{\mathrm{T}}_j \\ \widehat{\mathbf{A}}^{-1}_{i,B_i} \mathbf{d}_i(\mathbf{d}) & \widehat{\mathbf{A}}^{-1}_{i,B_i} \left(\widehat{\mathbf{A}}^{\{j\}}_{i,P_{i+1}} \right) \end{bmatrix}. \tag{4.24}$$

What is important to note is that the last m_i rows of (4.24) form the first $n_{i+1} + 1$ columns of the tableau in assumption 4, except with the j^{th} column equal to $\mathbf{0}$. Thus, tableau (4.24) is equal to

$$\begin{bmatrix} \mathbf{0} & \mathbf{E}_{i+1} \\ \mathbf{A}_{B_1}^{-1}\mathbf{d} & \mathbf{A}_{B_1}^{-1}\left(\mathbf{A}_{P_{i+1}}^{N_{i+1}}\right) \end{bmatrix}. \tag{4.25}$$

Similarly to the previous case, in this case the $(i+1)^{th}$-level and projected LPs are equivalent. To see this, note that the feasible set of the $(i+1)^{th}$ projected LP (4.23) is equivalent to the solution set of the i^{th} projected LP by Lemma 4.5.3 (only null variables have been removed by the definition of P_{i+1}). By assumption 2 (equivalence) and Lemma 4.5.1, the equivalence of the $(i+1)^{th}$-level and projected LPs follows.

We now optimize the $(i+1)^{th}$ projected LP (however it was constructed). The reason behind considering the projected LPs is that we can assert that after a primal simplex pivot, the new basis is $B'_{i+1} = N_{i+1} \cup B'_1$, where B'_1 is an optimal basis for the first-level LP. We also assert that $B'_k = N_k \cup B'_1$ is optimal for the k^{th} projected LP for all $k \leq i$. Further, the tableau retains the same form, and the reduced costs of the first-level and the k^{th} projected LPs do not change:

$$\mathbf{c}_{k,P_k}^{\mathrm{T}} - \mathbf{c}_{k,B_k}^{\mathrm{T}}\widehat{\mathbf{A}}_{k,B_k}^{-1}\widehat{\mathbf{A}}_k = \mathbf{c}_{k,P_k}^{\mathrm{T}} - \mathbf{c}_{k,B'_k}^{\mathrm{T}}\widehat{\mathbf{A}}_{k,B'_k}^{-1}\widehat{\mathbf{A}}_k \tag{4.26}$$

for all $k \leq i$. Since $\mathbf{d} \in F$ and the $(i+1)^{th}$-level and projected LPs are equivalent, the primal simplex algorithm must terminate. At this point we will have optimal bases $B_k^* = N_k \cup B_1^*$, for the k^{th} projected LP, for all $k \leq i+1$, where B_1^* is optimal for the first-level LP.

To see this, first note that we have for $k \in \{1, \ldots, i\}$,

$$c_{k,j} - \mathbf{c}_{k,B_k}^{\mathrm{T}}\widehat{\mathbf{A}}_{k,B_k}^{-1}\widehat{\mathbf{A}}_{k,j} = 0, \quad \forall j \in P_{i+1} \backslash B_{i+1}. \tag{4.27}$$

This follows from the construction of the index sets P_{i+1} and B_{i+1} (in either case), induction assumption 3, and the inclusion $P_{i+1} \backslash B_{i+1} \subset P_i \backslash B_i$ (which follows from $P_{i+1} \subset P_i$ and $B_{i+1} \supset B_i$). Now consider the specifics of a primal simplex pivot. Under any pivoting rule, let the index of the pivot column chosen be $p_c \in P_{i+1} \backslash B_{i+1}$. Note that the first $m_{i+1} - m_1$ elements of the p_c^{th} column of the tableau (4.25) are zero (the only columns of $\mathbf{E}_{i+1}$ that have nonzero elements correspond to $N_{i+1} \subset B_{i+1}$). So to determine the pivot row we only need to consider the p_c^{th} column of $\mathbf{A}_{B_1}^{-1}\left(\mathbf{A}_{P_{i+1}}^{N_{i+1}}\right)$, but this in fact equals $\mathbf{A}_{B_1}^{-1}\mathbf{A}_{p_c}$. This means that whatever basis element is chosen to exit the basis B_{i+1} is the same element that would exit the basis B_1 if we applied the primal simplex algorithm to the first-level LP and had chosen the p_c^{th} column as the pivot column. By (4.27), the p_c^{th} reduced cost of the first-level LP (given by B_1) is zero, and so this leads us to the conclusion that by following the pivot rules of the primal simplex algorithm applied to the $(i+1)^{th}$ projected LP, we are in fact executing acceptable pivots of the primal simplex algorithm applied to the first-level LP. Further, the discussion following Lemma 4.5.3 establishes that the reduced costs of the first-level LP will remain the same after the pivot (i.e. Eqn. (4.26) holds for $k = 1$). Consequently, we obtain the new primal feasible basis $B'_{i+1} = N_{i+1} \cup B'_1$ for the $(i+1)^{th}$ projected LP, where B'_1 is still optimal for the first-level LP.

Similar reasoning establishes that these pivots are also acceptable primal simplex pivots applied to the k^{th} projected LP, for *all* $k \leq i$. The p_c^{th} reduced cost of the k^{th} projected LP is zero, and so again all the reduced costs retain the same value after the pivot and Eqn. (4.26) holds for $k \in \{2, \dots, i\}$. Again, this means $B'_k = N_k \cup B'_1$ is optimal for the k^{th} projected LP. Further, whatever index p_{out} exits the basis (i.e. $B'_1 = \{p_c\} \cup B_1 \backslash \{p_{out}\}$) will have zero reduced cost in the k^{th} projected LP (since it was basic in the old basis and the reduced costs have the same values after the pivot). Then by (4.27) and since $P_{i+1} \backslash B'_{i+1} \subset \{p_{out}\} \cup (P_{i+1} \backslash B_{i+1})$, we can claim that for each $k \leq i$ and $j \in P_{i+1} \backslash B'_{i+1}$, that the j^{th} reduced cost of the k^{th} projected LP is still zero with the new basis B'_k (that is, $c_{k,j} - \mathbf{c}^{\mathrm{T}}_{k,B'_k} \widehat{\mathbf{A}}^{-1}_{k,B'_k} \widehat{\mathbf{A}}_{k,j} = 0$).

Further, the tableau for the $(i+1)^{th}$ projected LP after this pivot operation has the same form as tableau (4.25) (just with B'_1 replacing B_1). This is because the pivot operation is executed by multiplying the tableau (from the left) by a $m_{i+1} \times m_{i+1}$ matrix of the form

$$\begin{bmatrix} \mathbf{I}_i & \mathbf{0}_i^{\mathrm{T}} \\ \mathbf{0}_i & \mathbf{Q}_1 \end{bmatrix},$$

where $\mathbf{Q}_1$ is an invertible $m_1 \times m_1$ matrix, $\mathbf{I}_i$ is the $(m_{i+1} - m_1) \times (m_{i+1} - m_1)$ identity matrix, and $\mathbf{0}_i$ is a $m_1 \times (m_{i+1} - m_1)$ matrix of zeros. If the index of the pivot row is p_r, then $\mathbf{Q}_1 \mathbf{A}_{B_1}^{-1} \mathbf{A}_{p_c}$ equals the the $(p_r - (m_{i+1} - m_1))^{th}$ unit vector in $\mathbb{R}^{m_1}$. This achieves the overall effect of the pivot operation, which is to change the p_c^{th} column (of tableau (4.25)) into the p_r^{th} unit vector in $\mathbb{R}^{m_{i+1}}$.

Therefore, when the simplex method terminates for the $(i+1)^{th}$ projected LP, we will have an optimal basis $B^*_k = N_k \cup B^*_1$ for the k^{th} projected LP, for all $k \in \{1, \dots, i+1\}$, where B_1^* is optimal for the first-level LP. All the induction assumptions hold for the $(i+1)^{th}$ step; equivalence (induction assumption 2) has already been established, while the statement about the bases and reduced costs (induction assumption 3) and the form of the tableau (induction assumption 4) hold by the discussion above and a mini-induction argument for the sequence of simplex pivots applied to the $(i+1)^{th}$ projected LP.

Proceeding by induction, it follows that we can obtain an optimal basis for the n_q^{th} projected LP, $B_{n_q}^* = N_{n_q} \cup B_1^*$, where B_1^* is an optimal basis for the first-level LP (4.1). The basis $B_{n_q}^*$ describes the point $\mathbf{v}^*$; by equivalence and the nature of the lexicographic LP, this point is in the solution set of the i^{th}-level LP (4.2) for all i. Again by assumption 4, the only nonzero components of $\mathbf{v}^*$ are those corresponding to B_1^*, so we have $\mathbf{c}_i^{\mathrm{T}}\mathbf{v}^* = \mathbf{c}_{i,B_1^*}^{\mathrm{T}}\mathbf{A}_{B_1^*}^{-1}\mathbf{d}$ for all i. So we have that Eqn. (4.18) holds.

We now establish the final claim that Eqn. (4.18) holds for all $\mathbf{d}$ such that B_1^* is optimal. The reasoning follows from the previous argument, although formally a separate induction argument is needed. The essence of the argument is that the basis $B_i^* = N_i \cup B_1^*$ is optimal for the corresponding projected LP as defined earlier for all $\mathbf{d}$ such that B_1^* is optimal for the first-level LP. This is because dual feasibility for each basis does not change, while the form of the tableau from induction assumption 4 indicates that primal feasibility of B_1^* implies primal feasibility of B_i^*. Further, if the i^{th}-level and projected LPs are equivalent for all $\mathbf{d}$ such that B_1^* is optimal, then the $(i+1)^{th}$-level and projected LPs are equivalent for all $\mathbf{d}$ such that B_1^* is optimal. This follows from application of Lemma 4.5.1 and, if necessary, Lemma 4.5.3, which indicates that null variables remain null variables for all $\mathbf{d}$ such that B_i^* is optimal. Combined with the previous observation that optimality of B_1^* implies optimality of B_i^*, this means that the $(i+1)^{th}$-level and projected LPs are equivalent for all $\mathbf{d}$ such that B_1^* is optimal. If the construction terminated early after determining that the i^{th} projected LP has a unique solution, then this projected LP has a unique solution for as long as the basis B_i^* is optimal, which again holds for all $\mathbf{d}$ such that B_1^* is optimal. The conclusion of the induction argument is that for all $\mathbf{d}$ such that B_1^* is optimal, B_1^* describes a point in the solution set of each projected LP, and by equivalence, a point in the solution set of each level of the lexicographic LP (4.1)-(4.2). □

4.6 Examples

The simple example from §4.3 is reconsidered to clarify the qualitative difference between Algorithm 1 and the previously mentioned direct and time-stepping methods. Then, two examples based on dynamic flux balance analysis are presented. In §4.6.2, a model of batch fermentation displaying domain issues is presented. This example also demonstrates a significant numerical difference between the performance of Algorithm 1 and the direct method. In §4.6.3, a model of batch fermentation is presented in which a non-unique solution set of the embedded LP is encountered. The LP is reformulated as a lexicographic LP to resolve the non-uniqueness to obtain a better-defined and more numerically tractable problem. Numerical examples are performed on a 32-bit Linux virtual machine allocated a single core of a 3.07 GHz Intel Xeon CPU and 1.2 GB RAM. In the DSL48LPR implementation of Algorithms 1 and 2, relative and absolute integration tolerances are 10^{-6}, and in Algorithm 1 we set $\epsilon = 10^{-6}$ and $\delta = 2\epsilon$.

4.6.1 Robustness for simple example

Consider once more the simple example from §4.3.1. The solution estimate after an explicit Euler step (of stepsize h) is still $\widetilde{\mathbf{x}}(h) = (h, 0)$. As in §4.3.1, $\widetilde{\mathbf{x}}(h) \notin K = \{\mathbf{z} : z_2 \geq z_1^2\}$. However, in contrast with the direct method, this is not a complication; at any time t, the system of equations to be solved for the DAE reformulation from Algorithm 1 is

$$\begin{aligned} \widetilde{\mathbf{x}}(t+h) - \widetilde{\mathbf{x}}(t) - h\mathbf{f}(\widetilde{\mathbf{x}}(t), q^B(\widetilde{\mathbf{u}}_B(t))) &= \mathbf{0}, \\ \mathbf{A}_B\widetilde{\mathbf{u}}_B(t) - \mathbf{b}(\widetilde{\mathbf{x}}(t)) &= \mathbf{0}, \end{aligned}$$

where q^B is defined as in Algorithm 1. Whatever the choice of the basis B is, $\mathbf{u}_B(t)$ and q^B are well defined and the system of equations has a solution. This is a significant qualitative difference between Algorithm 1 and the direct or time-stepping methods.

Of course, this qualitative difference translates to a noticeable difference in numerical performance. When the solution is at the boundary of K, only Algorithm 1 can guarantee that an approximate solution can be continued. As demonstrated by the next example, this can lead to an unmistakable difference in the quality of the numerical solution. Specifically, the direct method fails or gives an incorrect indication of when the solution of the extended

IVP is no longer a solution of the original IVP (4.4).

4.6.2 *E. coli* fermentation

Batch and fed-batch fermentation reactions are important industrial processes for the production of valuable chemicals such as ethanol. This example considers a model of a fermentation reactor consisting of the dynamic mass balances of the reactor coupled to a genome-scale network reconstruction of the *E. coli* metabolism presented in [70]. Using information gleaned from genomic analysis, *E. coli*'s metabolism can be modeled as a network of reactions that must satisfy simple stoichiometric constraints. Analysis and construction of such a network is called flux balance analysis (FBA) [140]. However, this network is often under-determined; the fluxes of the different substrates and metabolites can vary and still produce a system that satisfies the stoichiometric constraints. Thus, one assumes that fluxes will be such that some cellular objective is maximized. Most often, the production of biomass is chosen as the cellular objective to maximize, and in general it is a reasonable choice [143]. The result, then, is in fact a system that has the same form as (4.4). The simulation represents the initial phase of batch operation of the fermentation reactor under aerobic growth on glucose and xylose media. No ethanol production during aerobic conditions is observed; this phase is used to increase the biomass. Thus, the concentration of ethanol is omitted from the dynamics.

Model

The dynamic mass balance equations of the extracellular environment of the batch reactor are

$$\begin{aligned} \dot{x}(t) &= \mu(t)x(t), \\ \dot{g}(t) &= -m_g u_g(t)x(t), \\ \dot{z}(t) &= -m_z u_z(t)x(t), \end{aligned} \tag{4.28}$$

where $\mathbf{x}(t) = (x(t), g(t), z(t))$ is the vector of biomass, glucose and xylose concentrations, respectively, at time t. The uptake kinetics for glucose, xylose and oxygen are given by the

Michaelis-Menten kinetics

$$u_g(t) = u_{g,max}\frac{g(t)}{K_g + g(t)}, \tag{4.29}$$

$$u_z(t) = u_{z,max}\frac{z(t)}{K_z + z(t)}\frac{1}{1 + \frac{g(t)}{K_{ig}}}, \tag{4.30}$$

$$u_o(t) = u_{o,max}\frac{o(t)}{K_o + o(t)}. \tag{4.31}$$

It is assumed that the oxygen concentration in the reactor, $o(t)$, is controlled and therefore a known value; see Table 4.1 for parameter values. Meanwhile, the growth rate $\mu(t)$ is determined from the metabolic network model of the *E. coli* bacterium iJR904 [153], which is available online [164]. The model consists of 625 unique metabolites, 931 intracellular fluxes, 144 exchange fluxes and an additional flux representing the biomass generation as growth rate $\mu(t)$. The flux balance model is an LP of the form

$$\begin{aligned} \mu(t) = \min_{\mathbf{v}\in\mathbb{R}^{n_v}} \ & \mathbf{c}^{\mathrm{T}}\mathbf{v} \\ \text{s.t. } & \mathbf{S}\mathbf{v} = \mathbf{0}, \\ & v_{g_{ext}} = u_g(t), \\ & v_{z_{ext}} = u_z(t), \\ & v_{o_{ext}} = u_o(t), \\ & \mathbf{v}^{LB} \leq \mathbf{v} \leq \mathbf{v}^{UB}, \end{aligned} \tag{4.32}$$

where n_v is the number of fluxes, n_m is the number of metabolites, $\mathbf{S} \in \mathbb{R}^{n_m \times n_v}$ is the stoichiometry matrix of the metabolic network, and $\mathbf{v}^{LB}$ and $\mathbf{v}^{UB}$ are the lower and upper bounds on the fluxes. The metabolic network is connected to the extracellular environment through the exchange fluxes for glucose, xylose and oxygen $v_{g_{ext}}$, $v_{z_{ext}}$, and $v_{o_{ext}}$, respectively, which are given by Equations (4.29)-(4.31). After putting the LP (4.32) in standard form and assuring that it satisfies Assumption 4.5.1, the LP has 749 constraints and 2150 primal variables. Numerical parameter values are according to [70], repeated in Table 4.1.

Table 4.1: Parameter values for *E. coli* model, Equations (4.28)-(4.32).

Parameter/Symbol	Value/Expression
$[t_0, t_f]$	[0, 10] (h)
$u_{g,max}$	10.5 (mmol/g/h)
$u_{z,max}$	6 (mmol/g/h)
$u_{o,max}$	15 (mmol/g/h)
K_g	0.0027 (g/L)
K_z	0.0165 (g/L)
K_o	0.024 (mmol/L)
K_{ig}	0.005 (g/L)
m_g	0.18 (g/mmol)
m_z	0.15 (g/mmol)
o	$t \mapsto 0.24$ (mmol/L)
$x(t_0)$	0.03 (g/L)
$g(t_0)$	15.5 (g/L)
$z(t_0)$	8 (g/L)

Simulation results

The solution of the system (4.28)-(4.32) was calculated with DSL48LPR and, for comparison, with the direct method, which was implemented with DSL48E (without any events) with the function evaluator calling CPLEX.

The time evolution of the dynamic states is shown in Fig. 4-1. First glucose, as the preferred carbon source, is consumed. After glucose has been depleted, at around 7h, the optimal basis changes and xylose becomes the main carbon source. The final batch time is determined by the glucose and xylose concentrations. The simulation stops when glucose and xylose concentration are equal to zero (around 8.2h); at this point, the LP is infeasible and so by Corollary 4.4.2 the solution ceases to exist. This makes sense physically, since with no carbon source the *E. coli* stop growing and begin to die; cell death is not a phase that this particular flux balance model can really predict and so the simulation must stop.

When simulating the system with DSL48E and CPLEX, the simulation fails at the point when the *E. coli* switches from glucose to xylose metabolism. This is clear when examining the primal variables (the fluxes) in Fig. 4-2. The values of the primal variables change quite rapidly (however they are still continuous). This indicates that the system (4.28)-(4.32) is stiff. Stiff dynamics combined with the numerical manifestation of domain issues as discussed in §4.3 cause the direct method to fail. In contrast, DSL48LPR manages to integrate past the change in metabolism and more accurately indicate when the solution fails to exist.

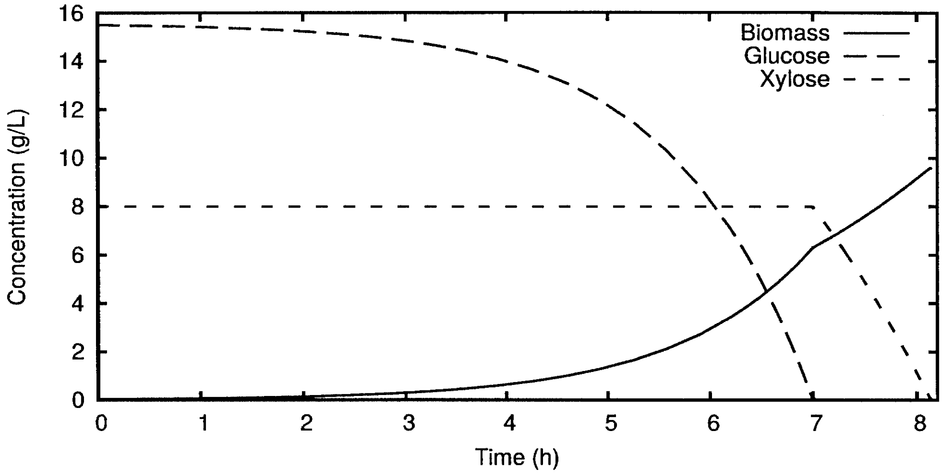


Figure 4-1: Species concentrations from Equations (4.28)-(4.32) in bioreactor as calculated by DSL48LPR.

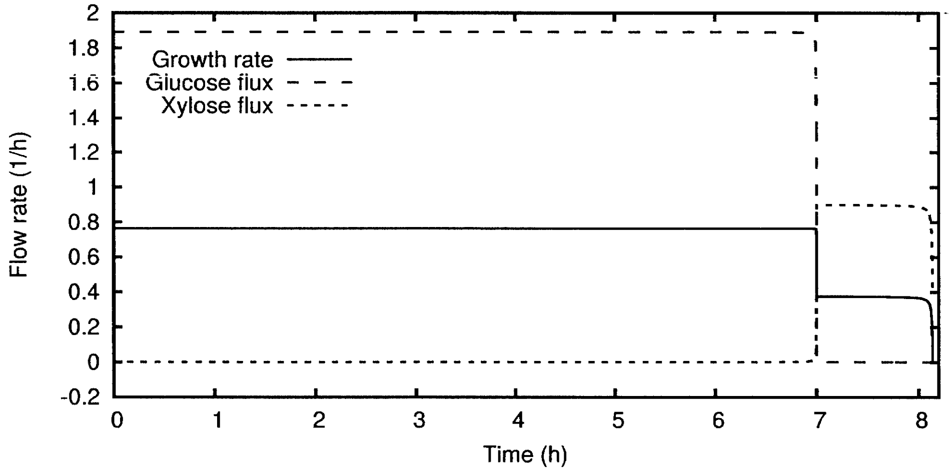


Figure 4-2: A representative selection of exchange fluxes (solution of LP (4.32)) as calculated by DSL48LPR. Note the extremely steep, but still continuous, change at around 7h, when the metabolism changes.

Computational times

This example also provides a good chance to compare the computational times for various solution methods. The time required by DSL48LPR and by various forms of the direct method to complete the simulation are compared in Table 4.2. The direct method was implemented using various different LP algorithms, and this can impact the solution time quite strongly. DSL48LPR is fast, both on the interval $[0, 7]$h and on the whole simulation interval. Meanwhile, DSL48E embedding CPLEX fails to complete the entire simulation, but the computational time to run the simulation to the point of failure can vary quite a lot. Using dual simplex with an advanced basis is the fastest, and competitive with DSL48LPR. This follows from the fact that using a dual feasible basis to warm start dual simplex is very similar to the basic algorithm of DSL48LPR. While this basis is also optimal, CPLEX only needs to solve a linear system to determine the values of the primal variables given the new value of the right-hand side vector. It should be noted that, to our knowledge, this use of dual simplex has not been proposed before for the solution of ODEs with LPs embedded.

The other LP algorithms, however, increase the simulation time. Neglecting that a dual feasible basis is available and using full (Phase I and Phase II) simplex is slower, followed by a barrier method (most likely the primal-dual path following algorithm, see §9.5 of [25]). Although interior point methods for LPs are praised for their polynomial solution time, it is an unwise choice in this context. Comparable to a nonlinear solve in at least 2000 variables, it incurs much more overhead, likely because it is factoring the necessary matrices more often than DSL48E is factoring the Jacobian within DSL48LPR. Further, it is possible that there are issues initializing the algorithm, since the previous solution point may be infeasible after a perturbation of the value of $\mathbf{b}$; consequently, the algorithm again lacks advanced starting point information which slows it down considerably.

4.6.3 Yeast fermentation

Normally, the solution sets of flux-balance models are not singletons [112]. Consider a second dynamic flux balance simulation of fed-batch fermentation using *Saccharomyces cerevisiae.* Besides ethanol, as the main metabolic product of interest, other by-products, such as glycerol, can be analyzed. A non-unique glycerol flux is predicted by the metabolic network reconstruction iND750 [46] of *S. cerevisiae* under anaerobic growth conditions [79]. In order

Table 4.2: Computational times (averaged over 50 runs) and integration statistics for solving Equations (4.28)-(4.32) with various methods. The "$*$" symbol indicates that the method failed before finishing the simulation.

		DSL48E embedding CPLEX		
Method	DSL48LPR	Dual Simplex	Full Simplex	Barrier Method
CPU time (s) (full simulation)	1.196	$*$	$*$	$*$
CPU time (s) (on $[0,7]$h)	1.004	0.436	2.799	4.772
Integration steps (on $[0,7]$h)	408	125	125	125
Jacobian evaluations (on $[0,7]$h)	169	53	53	53
Error test failures (on $[0,7]$h)	30	22	22	22
Convergence test failures (on $[0,7]$h)	0	11	11	11

to determine the range of the glycerol flux during batch fermentation, this example utilizes a lexicographic LP to determine a maximum and then minimum glycerol flux at the optimal growth rate.

This model has been considered in [80] for the production of ethanol by fed-batch fermentation of *S. cerevisiae.* The dynamics are

$$\begin{aligned}
\dot{v}(t) &= d(t), \qquad (4.33)\\
\dot{g}(t) &= -m_g u_g(t)x(t) + d(t)(g_{in} - g(t))/v(t),\\
\dot{x}(t) &= u_b(t)x(t) - d(t)x(t)/v(t),\\
\dot{e}(t) &= m_e u_e(t)x(t) - d(t)e(t)/v(t),\\
\dot{h}(t) &= m_h u_h(t)x(t) - d(t)h(t)/v(t),
\end{aligned}$$

where $v(t)$ is the total volume in the reactor, $d(t)$ is the dilution rate, and $g(t)$, $x(t)$, $e(t)$ and $h(t)$ are the concentrations of glucose, biomass, ethanol and glycerol respectively, in the reactor. Uptake kinetics for glucose and oxygen are again given by Michaelis-Menten kinetics with ethanol inhibition

$$u_g(t) = u_{g,max}\frac{g(t)}{K_g + g(t)}\frac{1}{1+\frac{e(t)}{K_{ie}}}, \tag{4.34}$$

$$u_o(t) = u_{o,max}\frac{o(t)}{K_o + o(t)}. \tag{4.35}$$

Meanwhile, g_{in} is the constant glucose inlet concentration, and $u_b(t)$, $u_e(t)$, $u_h(t)$ are given by

$$\begin{aligned} u_b(t) = \max_{\mathbf{v}}\ & v_b \\ \text{s.t. } & \mathbf{Av} = \mathbf{b}(g(t), e(t), o(t)), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{4.36}$$

$$\begin{aligned} u_e(t) = \max_{\mathbf{v}}\ & v_e \\ \text{s.t. } & \mathbf{Av} = \mathbf{b}(g(t), e(t), o(t)), \\ & v_b = u_b(t), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{4.37}$$

$$\begin{aligned} \text{and } u_h(t) = \max_{\mathbf{v}}\ & v_h \\ \text{s.t. } & \mathbf{Av} = \mathbf{b}(g(t), e(t), o(t)), \\ & v_e = u_e(t), \\ & v_b = u_b(t), \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{4.38}$$

The LP (4.36) is obtained by transforming a flux balance model for yeast in a similar manner to what was done in the previous example; (4.36) is connected to the extracellular environment via the Michaelis-Menten equations (4.34) and (4.35), and then put into standard form.

Note that $(u_b(t), u_e(t), u_h(t))$ is the solution to a lexicographic LP. After maximizing the growth rate, the optimal growth rate is added as a constraint and the resulting program is optimized with respect to ethanol flux. This optimal ethanol flux is again added as a

Table 4.3: Parameter values for Yeast model, Equations (4.33)-(4.38).

Parameter/Symbol	Value/Expression
$[t_0, t_f]$	$[0, 16]$ (h)
$u_{g,max}$	20 (mmol/g/h)
$u_{o,max}$	8 (mmol/g/h)
K_g	0.5 (g/L)
K_{ie}	10 (g/L)
K_o	0.003 (mmol/L)
m_g	0.18015 (g/mmol)
m_e	0.046 (g/mmol)
m_h	0.092 (g/mmol)
o	$t \mapsto \begin{cases} 0.15 \text{ (mmol/L)}, & t < 7.7; \\ 0 \text{ (mmol/L)}, & t \geq 7.7 \end{cases}$
d	$t \mapsto 0.044$ (L/h)
g_{in}	100 (g/L)
$v(t_0)$	0.5 (L)
$g(t_0)$	10 (g/L)
$x(t_0)$	0.05 (g/L)
$e(t_0)$	0 (g/L)
$h(t_0)$	0 (g/L)

constraint and then glycerol flux is maximized. The result is that these three fluxes are now uniquely defined and the dynamic problem (4.33) is well-defined. It is more difficult to address the non-uniqueness of the glycerol flux when solving (4.33) with the direct method; even if it is considered it requires the solution of extra LPs which can be costly. Meanwhile, a lexicographic LP provides a more straightforward way to enforce uniqueness, which reduces the ambiguity of the simulation results.

The parameter values for the simulation are from [80], repeated in Table 4.3. The simulation presents an aerobic-anaerobic operation. The aerobic to anaerobic switch occurs at 7.7h, after which a range of glycerol flux rates are possible. This leads to a maximum and minimum possible glycerol concentration; the discrepancy is called the production envelope [112]. To determine this envelope, a second simulation in which glycerol flux is instead minimized in (4.38) is performed. This simulation shows no glycerol production throughout the batch reaction. At the end of the simulations, the difference between the maximum and minimum glycerol concentrations is 3.71 g/L, where the concentrations of nutrients and metabolites are on the order of 10 g/L throughout the simulation. Clearly, a non-unique solution of the LP can have a significant impact on the overall solution of the dynamic system. The results are seen in Fig. 4-3.

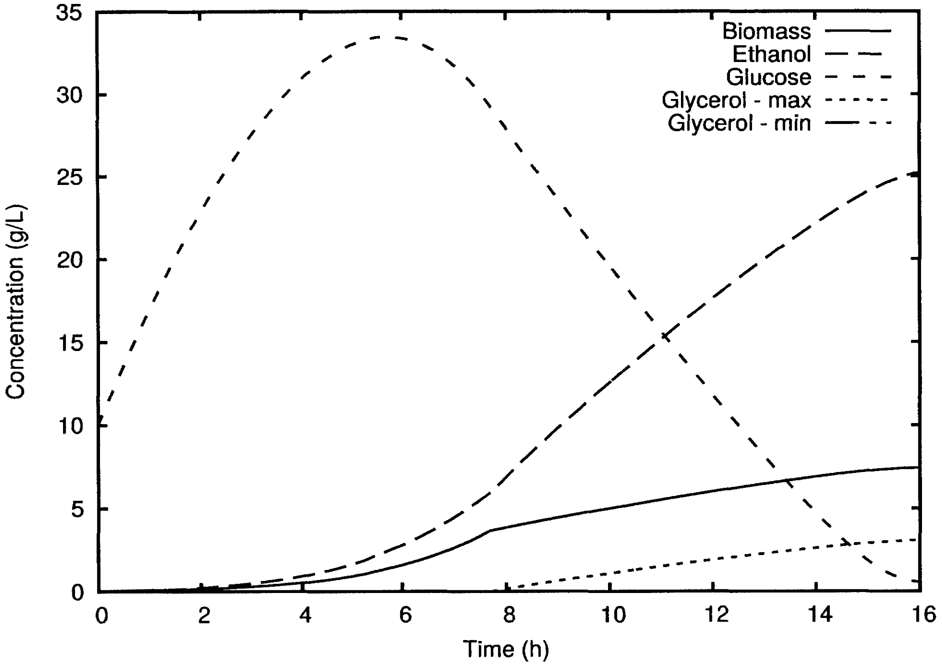


Figure 4-3: Species concentrations from Equations (4.33)-(4.38) in bioreactor as calculated by DSL48LPR. Note that the glycerol concentration potentially can take a range of values, if the glycerol flux is not explicitly fixed to a maximal or minimal value (minimal value is zero).

4.7 Conclusions

This chapter has analyzed the initial value problem in ordinary differential equations with a parametric lexicographic linear program embedded. This problem finds application in dynamic flux balance analysis, which is used in the modeling of industrial fermentation reactions. This work has proposed a numerical method which has distinct advantages over other applicable methods. These advantages allow the method to be applied successfully to examples of DFBA, and achieve unambiguously improved approximate solutions to these examples. The current implementation of the proposed method proves very successful in the motivating application of DFBA. Furthermore, the method is flexible and allows various numerical integration routines to be applied.

Chapter 5

Bounds on reachable sets using ordinary differential equations with linear programs embedded

5.1 Introduction

The problem of interest is the computation of time-varying enclosures of the reachable sets of the initial value problem (IVP)

$$\begin{aligned} \dot{\mathbf{x}}(t, \mathbf{u}, \mathbf{x}_0) &= \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t, \mathbf{u}, \mathbf{x}_0)), \qquad (5.1) \\ \mathbf{x}(t_0, \mathbf{u}, \mathbf{x}_0) &= \mathbf{x}_0, \end{aligned}$$

where $\mathbf{u}$ and $\mathbf{x}_0$ take values in some set of permissible controls and initial conditions, respectively. Using the bounding theory developed in [168], this chapter demonstrates that tight component-wise upper and lower bounds, called state bounds, can be computed by solving numerically a related IVP depending on parametric linear programs. Numerical considerations relate to the work in Ch. 4, but it useful to analyze the problem of ordinary differential equations (ODEs) with linear programs (LPs) embedded in the specific setting of this chapter.

Reachability analysis refers to estimating the set of possible states that a dynamic system may achieve for a range of parameter values or controls. This is an important task in state and parameter estimation [88, 98, 99, 150, 178], uncertainty propagation [72], safety verification and quality assurance [108, 82], and as well global dynamic optimization [177]. This problem traces back as far as the work in [24], however some of the more recent applicable references are [7, 107, 121, 176]. Meanwhile, the goal of this chapter is to introduce a new implementation of the theory developed in [168]. This theory provides a way to incorporate an "*a priori* enclosure" of the reachable sets to improve the estimates computed. As in §5.3.1, this is an enclosure of the reachable set based on mathematical manipulations of Eqn. (5.1). This type of information is conceptually distinct from continuous-time measurements of a physical system that (5.1) models. This type of information serves as a basis to improve the bounds obtained in [118, 128], for instance, which constructs bounds based on observers. Further, the bounding methods in [118, 128], still depend on an application of the classic "Müller theorem," of which the result in [168] is an extension.

The theory in [168] relies on differential inequalities, which in essence yields an IVP derived from (5.1) but involving parametric optimization problems. The implementation in [168] uses interval analysis to estimate the solutions of these optimization problems. This chapter will construct linear programs to estimate the solutions of the necessary optimization problems. An added benefit of this is that the implementation developed in this chapter can handle, in a meaningful way, a polyhedral set of admissible control values, which contrasts with the previous implementation in [168], and related work such as [176], for example, which employed interval arithmetic and so could only meaningfully handle an interval set of admissible control values.

The rest of the chapter is as follows. Section 5.2 introduces notation and establishes the formal problem statement concerning the reachable set estimation. Section 5.3 considers the state bounding problem and demonstrates that estimates of the reachable set can be obtained from the solution of an IVP in ODEs with LPs embedded. Section 5.4 considers numerical aspects of the solution of ODEs with LPs embedded. Section 5.5 applies this formulation to calculate state bounds for reacting chemical systems. Section 5.6 concludes with some final remarks.

5.2 Problem statement

The formal problem statement is as follows. Let $(n_x, n_u) \in \mathbb{N}^2$, nonempty $T = [t_0, t_f] \subset \mathbb{R}$, open $D_u \subset \mathbb{R}^{n_u}$, open $D \subset \mathbb{R}^{n_x}$, nonempty compact $U \subset D_u$, nonempty compact $X_0 \subset D$, and $\mathbf{f} : T \times D_u \times D \to \mathbb{R}^{n_x}$ be given. The goal is to compute functions $(\mathbf{x}^L, \mathbf{x}^U) : T \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ such that $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in [\mathbf{x}^L(t), \mathbf{x}^U(t)]$, for all $(t, \mathbf{u}, \mathbf{x}_0) \in T \times \mathcal{U} \times X_0$, where $\mathcal{U} = \{\mathbf{u} \in L^1(T, \mathbb{R}^{n_u}) : \mathbf{u}(t) \in U,\ a.e.\ t \in T\}$ and $\mathbf{x}$ is a solution of

$$\begin{aligned} \dot{\mathbf{x}}(t, \mathbf{u}, \mathbf{x}_0) &= \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t, \mathbf{u}, \mathbf{x}_0)), \quad a.e.\ t \in T, \\ \mathbf{x}(t_0, \mathbf{u}, \mathbf{x}_0) &= \mathbf{x}_0. \end{aligned} \tag{5.2}$$

Such $\mathbf{x}^L$ and $\mathbf{x}^U$ are called state bounds, as in [168]; the intervals $[\mathbf{x}^L(t), \mathbf{x}^U(t)]$ can also be thought of as enclosures of the reachable sets of the ODE system (5.2).

As mentioned, the approach to constructing bounds in this chapter involves parametric LPs, and so the results and discussion in §2.4 will be useful.

5.3 State bounding

5.3.1 An auxiliary IVP

Sufficient conditions for two functions to constitute state bounds of (5.2) are established in [168]. That paper also addresses how one can leverage an *a priori* enclosure to reduce the state bound overestimation. An *a priori* enclosure $G \subset \mathbb{R}^{n_x}$ is a rough enclosure of the solutions of (5.2): $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in G$, $\forall (t, \mathbf{u}, \mathbf{x}_0) \in T \times \mathcal{U} \times X_0$. Depending on the dynamics, physical arguments, such as conservation of mass, may inspire this. When the ODEs (5.2) are the dynamics of a chemical kinetics model, one can often determine a polyhedral G [167].

For the rest of this section assume there is a polyhedron G that is a rough enclosure for the solutions of (5.2), and that U is a nonempty compact polyhedron. Let $\mathbb{KR}_P^{n_x}$ denote the set of nonempty compact polyhedra in $\mathbb{R}^{n_x}$. Let $P_i^L, P_i^U : \mathbb{KR}_P^{n_x} \to \mathbb{KR}_P^{n_x}$ be given by

$$\begin{aligned} P_i^L &: \widehat{P} \mapsto \left\{ \mathbf{z} \in \widehat{P} : z_i = \min\{\zeta_i : \boldsymbol{\zeta} \in \widehat{P}\} \right\}, \\ P_i^U &: \widehat{P} \mapsto \left\{ \mathbf{z} \in \widehat{P} : z_i = \max\{\zeta_i : \boldsymbol{\zeta} \in \widehat{P}\} \right\}. \end{aligned}$$

Consider the system of ODEs

$$\begin{aligned}\dot{x}_i^L(t) &= q_i^L(t, \mathbf{x}^L(t), \mathbf{x}^U(t)) \\ &= \min\left\{f_i^{cv}(t, \mathbf{p}, \mathbf{z}, \mathbf{x}^L(t), \mathbf{x}^U(t)) : \mathbf{p} \in U, \mathbf{z} \in P_i^L\left([\mathbf{x}^L(t), \mathbf{x}^U(t)] \cap G\right)\right\}, \\ \dot{x}_i^U(t) &= q_i^U(t, \mathbf{x}^L(t), \mathbf{x}^U(t)) \\ &= \max\left\{f_i^{cc}(t, \mathbf{p}, \mathbf{z}, \mathbf{x}^L(t), \mathbf{x}^U(t)) : \mathbf{p} \in U, \mathbf{z} \in P_i^U\left([\mathbf{x}^L(t), \mathbf{x}^U(t)] \cap G\right)\right\},\end{aligned} \tag{5.3}$$

for $i \in \{1, \dots, n_x\}$, with initial conditions that satisfy $X_0 \subset [\mathbf{x}^L(t_0), \mathbf{x}^U(t_0)]$, where for each i, $f_i^{cv}(t, \cdot, \cdot, \mathbf{v}, \mathbf{w})$ is a convex piecewise affine under-estimator of $f_i(t, \cdot, \cdot)$ on $U \times P_i^L([\mathbf{v}, \mathbf{w}] \cap G)$ and $f_i^{cc}(t, \cdot, \cdot, \mathbf{v}, \mathbf{w})$ is a concave piecewise affine over-estimator of $f_i(t, \cdot, \cdot)$ on $U \times P_i^U([\mathbf{v}, \mathbf{w}] \cap G)$. Specifically, there exists a positive integer n_i^L, and for $k \in \{1, \dots, n_i^L\}$, there exist $\mathbf{c}_k^{i,L}(t, \mathbf{v}, \mathbf{w}) \in \mathbb{R}^{n_u + n_x}$ and $h_k^{i,L}(t, \mathbf{v}, \mathbf{w}) \in \mathbb{R}$ such that

$$f_i^{cv}(t, \mathbf{p}, \mathbf{z}, \mathbf{v}, \mathbf{w}) = \max\left\{(\mathbf{c}_k^{i,L}(t, \mathbf{v}, \mathbf{w}))^{\mathrm{T}}\mathbf{y} + h_k^{i,L}(t, \mathbf{v}, \mathbf{w}) : k \in \{1, \dots, n_i^L\}\right\} \leq f_i(t, \mathbf{p}, \mathbf{z}),$$

for each $\mathbf{y} = (\mathbf{p}, \mathbf{z}) \in U \times P_i^L([\mathbf{v}, \mathbf{w}] \cap G)$ (and similarly for f_i^{cc}, except it is taken as the pointwise minimum of a set of affine functions). It will now be shown that the solutions (if any) of (5.3) are state bounds for the system (5.2).

The goal is to apply Theorem 2 of [168]. Its statement and required assumptions are repeated below

Assumption 5.3.1. *For any $\mathbf{z} \in D$, there exists a neighborhood $N(\mathbf{z})$ of $\mathbf{z}$ and $\alpha \in L^1(T)$ such that for almost every $t \in T$ and every $p \in U$,*

$$\|\mathbf{f}(t, \mathbf{p}, \mathbf{z}_1) - \mathbf{f}(t, \mathbf{p}, \mathbf{z}_2)\| \leq \alpha(t)\,\|\mathbf{z}_1 - \mathbf{z}_2\|$$

for every $\mathbf{z}_1$ and $\mathbf{z}_2$ in $N(\mathbf{z}) \cap D$.

Assumption 5.3.2. *Assume $D_\Omega \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ and for $i \in \{1, \dots, n_x\}$, $(\Omega_i^L, \Omega_i^U) : D_\Omega \to \mathbb{KR}^{n_x} \times \mathbb{KR}^{n_x}$ satisfy the following.*

1. *For any $(\mathbf{v}, \mathbf{w}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$, if there exists $(t, \mathbf{u}, \mathbf{x}_0) \in T \times \mathcal{U} \times X_0$ satisfying $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in [\mathbf{v}, \mathbf{w}]$ and $x_i(t, \mathbf{u}, \mathbf{x}_0) = v_i$ (respectively, $x_i(t, \mathbf{u}, \mathbf{x}_0) = w_i$), then $(\mathbf{v}, \mathbf{w}) \in D_\Omega$ and $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in \Omega_i^L(\mathbf{v}, \mathbf{w})$ (respectively, $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in \Omega_i^U(\mathbf{v}, \mathbf{w})$).*

2. *For any* $(\mathbf{v}, \mathbf{w}) \in D_\Omega$*, there exists an open neighborhood* $N(\mathbf{v}, \mathbf{w})$ *of* $(\mathbf{v}, \mathbf{w})$ *and* $L > 0$ *such that*

$$d_H(\Omega_i^L(\mathbf{v}_1, \mathbf{w}_1), \Omega_i^L(\mathbf{v}_2, \mathbf{w}_2)) \leq L(\|\mathbf{v}_1 - \mathbf{v}_2\|_\infty + \|\mathbf{w}_1 - \mathbf{w}_2\|_\infty)$$

for all $(\mathbf{v}_1, \mathbf{w}_1)$ *and* $(\mathbf{v}_2, \mathbf{w}_2)$ *in* $N(\mathbf{v}, \mathbf{w}) \cap D_\Omega$*, and a similar statement for* Ω_i^U *also holds.*

Theorem 5.3.1 (Thm. 2 in [168]). *Let Assumptions 5.3.1 and 5.3.2 hold. Let* $(\mathbf{v}, \mathbf{w}) : T \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ *be absolutely continuous functions satisfying*

1. *For every* $t \in T$ *and every index* i*,*

   (a) $(\mathbf{v}(t), \mathbf{w}(t)) \in D_\Omega$*,*

   (b) $\Omega_i^L(\mathbf{v}(t), \mathbf{w}(t)) \subset D$ *and* $\Omega_i^U(\mathbf{v}(t), \mathbf{w}(t)) \subset D$*,*

2. $X_0 \subset [\mathbf{v}(t_0), \mathbf{w}(t_0)]$*,*
3. *For a.e.* $t \in T$ *and each index* i*,*

   (a) $\dot{v}_i(t) \leq f_i(t, \mathbf{p}, \mathbf{z})$*, for all* $\mathbf{z} \in \Omega_i^L(\mathbf{v}(t), \mathbf{w}(t))$ *and* $\mathbf{p} \in U$*,*

   (b) $\dot{w}_i(t) \geq f_i(t, \mathbf{p}, \mathbf{z})$*, for all* $\mathbf{z} \in \Omega_i^U(\mathbf{v}(t), \mathbf{w}(t))$ *and* $\mathbf{p} \in U$*,*

then $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in [\mathbf{v}(t), \mathbf{w}(t)]$*, for all* $(t, \mathbf{u}, \mathbf{x}_0) \in T \times \mathcal{U} \times X_0$*.*

The main challenge is defining D_Ω, Ω_i^L, Ω_i^U such that Assumption 5.3.2 holds. It is shown that this is the case if we let

$$\begin{aligned}
D_\Omega &= \{(\mathbf{v}, \mathbf{w}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} : [\mathbf{v}, \mathbf{w}] \cap G \neq \varnothing\}, \\
\Omega_i^L &: (\mathbf{v}, \mathbf{w}) \mapsto P_i^L\left([\mathbf{v}, \mathbf{w}] \cap G\right), \\
\Omega_i^U &: (\mathbf{v}, \mathbf{w}) \mapsto P_i^U\left([\mathbf{v}, \mathbf{w}] \cap G\right).
\end{aligned}$$

To see this, choose any $(\mathbf{v}, \mathbf{w}) \in D_\Omega$ and let

$$\begin{aligned}
z_i^m(\mathbf{v}, \mathbf{w}) &= \min\{\zeta_i : \boldsymbol{\zeta} \in [\mathbf{v}, \mathbf{w}] \cap G)\}, \qquad (5.4)\\
z_i^M(\mathbf{v}, \mathbf{w}) &= \max\{\zeta_i : \boldsymbol{\zeta} \in [\mathbf{v}, \mathbf{w}] \cap G)\}
\end{aligned}$$

(note that $v_i \leq z_i^m(\mathbf{v}, \mathbf{w}) \leq z_i^M(\mathbf{v}, \mathbf{w}) \leq w_i$). If there exists $(t, \mathbf{u}, \mathbf{x}_0) \in T \times \mathcal{U} \times X_0$ such that $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in [\mathbf{v}, \mathbf{w}]$, then $\mathbf{x}(t, \mathbf{u}, \mathbf{x}_0) \in [\mathbf{v}, \mathbf{w}] \cap G$ by definition of G, so $(\mathbf{v}, \mathbf{w}) \in D_\Omega$. Further, if $\mathbf{x}(t,\mathbf{u},\mathbf{x}_0) \in [\mathbf{v},\mathbf{w}] \cap G$ and $x_i(t,\mathbf{u},\mathbf{x}_0) = v_i$, then $z_i^m(\mathbf{v},\mathbf{w}) \le x_i(t,\mathbf{u},\mathbf{x}_0) = v_i \le z_i^m(\mathbf{v},\mathbf{w})$, so it is clear that $\mathbf{x}(t,\mathbf{u},\mathbf{x}_0) \in P_i^L([\mathbf{v},\mathbf{w}] \cap G)$. An analogous argument gives the condition for P_i^U.

To see that the second condition holds consider the nature of the sets $P_i^L([\mathbf{v},\mathbf{w}] \cap G)$. Since G is a polyhedron it can be expressed as $G = \{\mathbf{z} \in \mathbb{R}^{n_x} : \mathbf{A}_G\mathbf{z} \le \mathbf{b}_G\}$ for some $\mathbf{A}_G \in \mathbb{R}^{m_g \times n_x}$ and $\mathbf{b}_G \in \mathbb{R}^{m_g}$. Thus $[\mathbf{v},\mathbf{w}] \cap G = \{\mathbf{z} : \mathbf{A}\mathbf{z} \le \mathbf{b}(\mathbf{v},\mathbf{w})\}$ where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_G \\ -\mathbf{I} \\ \mathbf{I} \end{bmatrix}, \quad \mathbf{b}(\mathbf{v},\mathbf{w}) = \begin{bmatrix} \mathbf{b}_G \\ -\mathbf{v} \\ \mathbf{w} \end{bmatrix}. \tag{5.5}$$

By Lemma 2.4.2, z_i^m is a Lipschitz continuous function on D_Ω with Lipschitz constant L_1. Finally, noting that

$$\Omega_i^L(\mathbf{v},\mathbf{w}) = P_i^L([\mathbf{v},\mathbf{w}] \cap G) = \{\mathbf{z} : \mathbf{A}\mathbf{z} \le \mathbf{b}(\mathbf{v},\mathbf{w}), z_i \le z_i^m(\mathbf{v},\mathbf{w}), z_i \ge z_i^m(\mathbf{v},\mathbf{w})\},$$

by Lemma 2.4.2 there exists $L_2 > 0$ such that

$$\begin{aligned} d_H(\Omega_i^L(\mathbf{v}_1,\mathbf{w}_1),&\Omega_i^L(\mathbf{v}_2,\mathbf{w}_2)) \\ &\le L_2\left(\|\mathbf{b}(\mathbf{v}_1,\mathbf{w}_1) - \mathbf{b}(\mathbf{v}_2,\mathbf{w}_2)\|_\infty + 2\left|z_i^m(\mathbf{v}_1,\mathbf{w}_1) - z_i^m(\mathbf{v}_2,\mathbf{w}_2)\right|\right) \\ &\le L_2(\|\mathbf{v}_1 - \mathbf{v}_2\|_\infty + \|\mathbf{w}_1 - \mathbf{w}_2\|_\infty + 2L_1(\|\mathbf{v}_1 - \mathbf{v}_2\|_\infty + \|\mathbf{w}_1 - \mathbf{w}_2\|_\infty)) \\ &\le L_2(1 + 2L_1)(\|\mathbf{v}_1 - \mathbf{v}_2\|_\infty + \|\mathbf{w}_1 - \mathbf{w}_2\|_\infty) \end{aligned}$$

for all $(\mathbf{v}_1,\mathbf{w}_1)$ and $(\mathbf{v}_2,\mathbf{w}_2)$ in D_Ω. Similar reasoning shows that the required Lipschitz condition holds for each Ω_i^U as well.

The rest of the hypotheses of Theorem 5.3.1 are easy to verify. Solutions of (5.3) are understood in the Carathéodory sense, and thus $\mathbf{x}^L$ and $\mathbf{x}^U$ are absolutely continuous. Meanwhile, any solutions that exist must satisfy $[\mathbf{x}^L(t),\mathbf{x}^U(t)] \cap G \ne \varnothing$ (otherwise the optimization problems are not defined), which implies that $(\mathbf{x}^L(t),\mathbf{x}^U(t)) \in D_\Omega$. Further, the objective functions of the optimization problems are assumed to be affine underestimators of $f_i(t,\cdot,\cdot)$ on $U \times P_i^L([\mathbf{x}^L(t),\mathbf{x}^U(t)] \cap G)$, implying that $P_i^L([\mathbf{x}^L(t),\mathbf{x}^U(t)] \cap G) = \Omega_i^L(\mathbf{x}^L(t),\mathbf{x}^U(t)) \subset D$ (and similarly for Ω_i^U). It is already assumed that the initial conditions satisfy Hypothesis 2 of Theorem 5.3.1, and clearly Hypothesis 3 is satisfied. Thus, any solutions of (5.3) are state bounds.

5.3.2 Convergence

This section considers the convergence of the state bounds constructed in §5.3.1 as the "sizes" of the sets of admissible control values and initial conditions decrease. To begin, it is necessary to abuse notation in this section and allow the state bounds $\mathbf{x}^L$, $\mathbf{x}^U$ to have dependence on the sets of control values and initial conditions. That is to say, assume $[\mathbf{x}^L(t, U', X_0'), \mathbf{x}^U(t, U', X_0')] \ni \mathbf{x}(t, \mathbf{u}, \mathbf{x}_0)$, for any $(t, \mathbf{u}, \mathbf{x}_0) \in T \times \mathcal{U}' \times X_0'$, any solution $\mathbf{x}$ of IVP (5.2) (where $\mathcal{U}' = \{\mathbf{u} \in L^1(T, \mathbb{R}^{n_u}) : \mathbf{u}(t) \in U', \ a.e.\ t \in T\}$), and any nonempty compact polyhedron $U' \subset U$ and nonempty compact $X_0' \subset X_0$. For simplicity, write $X^B(t, U', X_0') = [\mathbf{x}^L(t, U', X_0'), \mathbf{x}^U(t, U', X_0')]$. Similarly, let the functions $\mathbf{q}^L$, $\mathbf{q}^U$ defining the dynamics in the auxiliary ODE system (5.3) have dependence on the polyhedral set of control values U'.

Next, make the following definitions (taken from Ch. 3 of [163]).

Definition 5.3.1. For $Y \subset \mathbb{R}^n$, denote the set of nonempty interval subsets of Y by $\mathbb{I}Y \equiv \{[\mathbf{v}, \mathbf{w}] : \mathbf{v} \leq \mathbf{w}, [\mathbf{v}, \mathbf{w}] \subset Y\}$. Denote the interval hull (the intersection of all interval supersets) of a set Y by hull(Y). If there exist τ, $\beta > 0$ such that

$$d_H\left(\text{hull}(\mathbf{x}(t, \mathcal{U}', X_0')), X^B(t, U', X_0')\right) \leq \tau w(U' \times X_0')^{\beta}, \quad \forall (t, U', X_0') \in T \times \mathbb{I}U \times \mathbb{I}X_0,$$

where $\mathcal{U}' = \{\mathbf{u} \in L^1(T, \mathbb{R}^{n_u}) : \mathbf{u}(t) \in U', \ a.e.\ t \in T\}$, then X^B is said to have Hausdorff convergence in $U \times X_0$ of order β with prefactor τ uniformly on T.

The next result establishes that the state bounds constructed in §5.3.1 have Hausdorff convergence in $U \times X_0$ of order at least 1 uniformly on T (with some prefactor). More colloquially, the state bounds are said to converge at least linearly with respect to the uncertain initial conditions and admissible control values. To simplify the discussion, the following result depends on the assumption that there exist state bounds that are known *a priori* to converge at least linearly. An example of such bounds are the "naïve" state bounds in Definition 3.4.3 in [163], where the functions $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{o}}$ defining the dynamics of the IVP (5.6) below are the lower and upper bounds, respectively, of the natural interval extension of $\mathbf{f}$. (Although the naïve state bounds in [163] are for parametric ODEs instead of control systems, a modification of the proof of Theorem 3.4.9 in [163] shows that these state bounds are also first-order convergent for control systems.) Meanwhile, Condition (5.7) below is satisfied as long as interval bounds on f_i are included in the definition of the piecewise affine estimators f_i^{cv}, f_i^{cc}. Since the affine relaxation method described in Ch. 3 requires simultaneous evaluation of interval bounds, this is easily satisfied.

Proposition 5.3.1. *Let $(\widetilde{\mathbf{u}}, \widetilde{\mathbf{o}}) : T \times \mathbb{I}D \times \mathbb{I}U \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ be locally Lipschitz continuous and monotonic in the sense that*

$$\begin{aligned}&\widetilde{\mathbf{u}}(t, [\mathbf{v}, \mathbf{w}], U') \leq \widetilde{\mathbf{u}}(t, [\mathbf{v}', \mathbf{w}'], U'), \ \text{and} \ \widetilde{\mathbf{o}}(t, [\mathbf{v}', \mathbf{w}'], U') \leq \widetilde{\mathbf{o}}(t, [\mathbf{v}, \mathbf{w}], U'),\\ &a.e.\ t \in T, \quad \forall([\mathbf{v}, \mathbf{w}], [\mathbf{v}', \mathbf{w}'], U') \in \mathbb{I}D \times \mathbb{I}D \times \mathbb{I}U : [\mathbf{v}', \mathbf{w}'] \subset [\mathbf{v}, \mathbf{w}].\end{aligned}$$

Suppose that for all $(U', X_0') \in \mathbb{I}U \times \mathbb{I}X_0$, $(\widetilde{\mathbf{v}}, \widetilde{\mathbf{w}}) : T \times \mathbb{I}U \times \mathbb{I}X_0 \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ are solutions of

$$\begin{aligned}\dot{\widetilde{\mathbf{v}}}(t, U', X_0') &= \widetilde{\mathbf{u}}(t, [\widetilde{\mathbf{v}}(t, U', X_0'), \widetilde{\mathbf{w}}(t, U', X_0')], U'),\\ \dot{\widetilde{\mathbf{w}}}(t, U', X_0') &= \widetilde{\mathbf{o}}(t, [\widetilde{\mathbf{v}}(t, U', X_0'), \widetilde{\mathbf{w}}(t, U', X_0')], U'),\end{aligned} \tag{5.6}$$

with initial conditions that satisfy

$$[\widetilde{\mathbf{v}}(t_0, U', X_0'), \widetilde{\mathbf{w}}(t_0, U', X_0')] \supset [\mathbf{x}^L(t_0, U', X_0'), \mathbf{x}^U(t_0, U', X_0')].$$

Assume that $\widetilde{\mathbf{v}}(\cdot, U', X_0')$ and $\widetilde{\mathbf{w}}(\cdot, U', X_0')$ are state bounds for the solutions of (5.2) *and that $\widetilde{\mathbf{v}}$ and $\widetilde{\mathbf{w}}$ have Hausdorff convergence in $U \times X_0$ of order* 1 *uniformly in T:*

$$\begin{aligned}&d_H\left(\text{hull}(\mathbf{x}(t, \mathcal{U}', X_0')), [\widetilde{\mathbf{v}}(t, U', X_0'), \widetilde{\mathbf{w}}(t, U', X_0')]\right) \leq \tau w(U' \times X_0'),\\ &\hspace{20em}\forall(t, U', X_0') \in T \times \mathbb{I}U \times \mathbb{I}X_0.\end{aligned}$$

Assume that

$$\begin{aligned}&[\mathbf{q}^L(t, \mathbf{v}, \mathbf{w}, U'), \mathbf{q}^U(t, \mathbf{v}, \mathbf{w}, U')] \subset [\widetilde{\mathbf{u}}(t, [\mathbf{v}, \mathbf{w}], U'), \widetilde{\mathbf{o}}(t, [\mathbf{v}, \mathbf{w}], U')],\\ &a.e.\ t \in T, \quad \forall([\mathbf{v}, \mathbf{w}], U') \in \mathbb{I}D \times \mathbb{I}U.\end{aligned} \tag{5.7}$$

Then the state bounds $(\mathbf{x}^L, \mathbf{x}^U)$ constructed in §5.3.1 (the solutions, if any, of the IVP (5.3)*), have Hausdorff convergence in $U \times X_0$ of order* 1 *uniformly in T.*

Proof. By Theorem 3.4.6 of [163], we have

$$[\widetilde{\mathbf{v}}(t,U',X_0'),\widetilde{\mathbf{w}}(t,U',X_0')] \supset [\mathbf{x}^L(t,U',X_0'),\mathbf{x}^U(t,U',X_0')]$$

for all $(t,U',X_0') \in T \times \mathbb{I}U \times \mathbb{I}X_0$. Since both are state bounds,

$$d_H\left(\text{hull}(\mathbf{x}(t,\mathcal{U}',X_0')),[\mathbf{x}^L(t,U',X_0'),\mathbf{x}^U(t,U',X_0')]\right) \leq \\ d_H\left(\text{hull}(\mathbf{x}(t,\mathcal{U}',X_0')),[\widetilde{\mathbf{v}}(t,U',X_0'),\widetilde{\mathbf{w}}(t,U',X_0')]\right),$$

and so by the assumption on the convergence of $\widetilde{\mathbf{v}}$ and $\widetilde{\mathbf{w}}$, we also have that $\mathbf{x}^L$ and $\mathbf{x}^U$ have Hausdorff convergence in $U \times X_0$ of order 1 uniformly in T. □

5.4 ODEs with LPs embedded

This section will analyze the IVP (5.3) that must be solved to obtain state bounds. Specifically, it will consider how to construct the convex and concave piecewise affine under and over-estimators of the dynamics as well as the numerical solution.

The system (5.3) is indeed an initial value problem in ODEs, where the dynamics are given by parametric optimization problems. Consider the equations defining $\dot{x}_i^L(t)$ for example. As discussed as the end of §2.4, since the objective function $f_i^{cv}(t,\cdot,\cdot,\mathbf{x}^L(t),\mathbf{x}^U(t))$ is a convex piecewise affine function, and the feasible set $U \times P_i^L([\mathbf{x}^L(t),\mathbf{x}^U(t)] \cap G)$ is a polyhedron, this is equivalent to a parametric linear optimization problem. Further, because the parameterization of this LP depends on the dynamic states $(\mathbf{x}^L(t),\mathbf{x}^U(t))$, these LPs must be solved along with the dynamic states during the integration routine. This is the essence of "ODEs with LPs embedded."

This formulation is similar to others that have appeared in the literature; see for instance the work on complementarity systems [165] and differential variational inequalities [144]. However, much of the work on these problems is slightly more general than necessary to understand and efficiently solve the IVP (5.3). Instead, the approach taken here will be to establish that, under mild assumptions, standard numerical integration routines and LP solvers can be used to solve the IVP of interest.

5.4.1 Lipschitz continuity of dynamics

First, it will be established that the dynamics of the IVP (5.3) satisfy the Lipschitz continuity condition in Definition 2.5.1. As discussed in §2.5, the purpose is to establish that the IVP (5.3) is amenable to solution by many different classes of numerical integration methods.

To achieve this, define the following parametric optimization problems which define the dynamics in Eqn. (5.3):

$$q_i^L(t, \mathbf{v}, \mathbf{w}) = \min \left\{ f_i^{cv}(t, \mathbf{p}, \mathbf{z}, \mathbf{v}, \mathbf{w}) : \mathbf{p} \in U, \mathbf{z} \in P_i^L\left([\mathbf{v}, \mathbf{w}] \cap G\right) \right\}, \tag{5.8}$$

$$q_i^U(t, \mathbf{v}, \mathbf{w}) = \max \left\{ f_i^{cc}(t, \mathbf{p}, \mathbf{z}, \mathbf{v}, \mathbf{w}) : \mathbf{p} \in U, \mathbf{z} \in P_i^U\left([\mathbf{v}, \mathbf{w}] \cap G\right) \right\}. \tag{5.9}$$

Analyze the LP (5.8) (the analysis for LP (5.9) is similar). Since U is a polyhedron, there exist a matrix $\mathbf{A}_U \in \mathbb{R}^{m_u \times n_u}$ and vector $\mathbf{b}_U \in \mathbb{R}^{m_u}$ such that the feasible set of (5.8) can be rewritten as

$$\begin{aligned}
&U \times P_i^L\left([\mathbf{v}, \mathbf{w}] \cap G\right) = \\
&\quad \left\{ (\mathbf{p}, \mathbf{z}) \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_x} : \mathbf{A}_U \mathbf{p} \leq \mathbf{b}_U, \mathbf{A}\mathbf{z} \leq \mathbf{b}(\mathbf{v}, \mathbf{w}), z_i \leq z_i^m(\mathbf{v}, \mathbf{w}), z_i \geq z_i^m(\mathbf{v}, \mathbf{w}) \right\},
\end{aligned} \tag{5.10}$$

where $\mathbf{A}$ and $\mathbf{b}$ are given by Eqn. (5.5) and z_i^m is given by Eqn. (5.4). For brevity, write this as

$$U \times P_i^L\left([\mathbf{v}, \mathbf{w}] \cap G\right) = \left\{ \mathbf{y} \in \mathbb{R}^{n_u + n_x} : \mathbf{A}_i^L \mathbf{y} \leq \mathbf{b}_i^L(\mathbf{v}, \mathbf{w}) \right\}.$$

By the discussion in §5.3, z_i^m is a Lipschitz continuous function on D_Ω. Thus, it is clear that $\mathbf{b}_i^L$ is as well.

Next, by definition, for some integer n_i^L, there are functions $\mathbf{h}^{i,L} : T \times D_\Omega \to \mathbb{R}^{n_i^L}$ and for $k \in \{1, \ldots, n_i^L\}$, $\mathbf{c}_k^{i,L} : T \times D_\Omega \to \mathbb{R}^{n_u + n_x}$ such that

$$f_i^{cv}(t, \mathbf{p}, \mathbf{z}, \mathbf{v}, \mathbf{w}) = \max \left\{ (\mathbf{c}_k^{i,L}(t, \mathbf{v}, \mathbf{w}))^{\mathrm{T}} \mathbf{y} + h_k^{i,L}(t, \mathbf{v}, \mathbf{w}) : k \in \{1, \ldots, n_i^L\} \right\},$$

where $\mathbf{y} = (\mathbf{p}, \mathbf{z})$. For more compact notation, let $\mathbf{c}^{i,L} = (\mathbf{c}_1^{i,L}, \ldots, \mathbf{c}_{n_i^L}^{i,L})$. Assume that $\mathbf{c}^{i,L}$ and $\mathbf{h}^{i,L}$ are locally Lipschitz continuous on D_Ω, uniformly on T (see Definition 2.5.1), and further are continuous on $T \times D_\Omega$. The following lemma establishes that for all $(\mathbf{v}, \mathbf{w}) \in D_\Omega$, there exists a neighborhood $N(\mathbf{v}, \mathbf{w})$ such that the images of $T \times N(\mathbf{v}, \mathbf{w}) \cap D_\Omega$ under $\mathbf{c}^{i,L}$ and $\mathbf{h}^{i,L}$ are bounded.

Lemma 5.4.1. *Let (T, d_T) be a compact metric space, (X, d_X) and (Y, d_Y) be metric spaces, and let $f : T \times X \to Y$ be continuous on $T \times X$ and locally Lipschitz continuous on X, uniformly on T. Then for all $x \in X$, there exists a neighborhood $N(x)$ of x such that f is bounded on $T \times N(x)$.*

Proof. Choose $x^* \in X$. By Definition 2.5.1, there exists a neighborhood $N(x^*)$ and $L(x^*) > 0$ such that

$$d_Y(f(t, x^*), f(t, x)) \leq L(x^*) d_X(x^*, x)$$

for all $(t, x) \in T \times N(x^*)$. By definition, there exists $\delta > 0$ such that $d_X(x^*, x) \leq \delta$ for all $x \in N(x^*)$, so $d_Y(f(t, x^*), f(t, x)) \leq L(x^*)\delta$ for all $(t, x) \in T \times N(x^*)$. Since $f(\cdot, x^*)$ is continuous on compact T, the image of T under $f(\cdot, x^*)$ is compact and so bounded. Thus there exist $b > 0$ and $y \in Y$ such that $d_Y(f(t, x^*), y) \leq b$ for all $t \in T$. Then for all $(t, x) \in T \times N(x^*)$,

$$d_Y(f(t, x), y) \leq d_Y(f(t, x), f(t, x^*)) + d_Y(f(t, x^*), y) \leq L(x^*)\delta + b$$

which establishes that f is bounded on $T \times N(x^*)$. □

If

$$\widetilde{q}_i^L(\widetilde{\mathbf{b}}, \widetilde{\mathbf{c}}_1, \ldots, \widetilde{\mathbf{c}}_{n_i^L}, \widetilde{\mathbf{h}}) = \min \left\{ \max_k \{(\widetilde{\mathbf{c}}_k^{\mathrm{T}} \mathbf{y} + \widetilde{h}_k\} : \mathbf{y} \in \mathbb{R}^{n_u + n_x}, \mathbf{A}_i^L \mathbf{y} \leq \widetilde{\mathbf{b}} \right\}, \tag{5.11}$$

then $q_i^L(t, \mathbf{v}, \mathbf{w}) = \widetilde{q}_i^L(\mathbf{b}_i^L(\mathbf{v}, \mathbf{w}), \mathbf{c}^{i,L}(t, \mathbf{v}, \mathbf{w}), \mathbf{h}^{i,L}(t, \mathbf{v}, \mathbf{w}))$. Let F_i^L be the set of $\widetilde{\mathbf{b}}$ such that the feasible set of optimization problem (5.11) is nonempty. It is easy to see that F_i^L is a closed set; see §4.7 of [25]. Then for all $\widetilde{\mathbf{b}} \in F_i^L$, (5.11) is an optimization problem with a convex piecewise affine objective over a nonempty, bounded polyhedral set. Consequently, we can apply Lemma 2.4.3 to see that $\widetilde{q}_i^L$ is locally Lipschitz continuous on $F_i^L \times \mathbb{R}^{n_i^L(n_u + n_x)} \times \mathbb{R}^{n_i^L}$. This implies that $\widetilde{q}_i^L$ is Lipschitz continuous on any compact subset of $F_i^L \times \mathbb{R}^{n_i^L(n_u + n_x)} \times \mathbb{R}^{n_i^L}$. Then, choose $(\mathbf{v}, \mathbf{w}) \in D_\Omega$. Without loss of generality, let $N(\mathbf{v}, \mathbf{w})$ be a neighborhood satisfying Definition 2.5.1 for both $\mathbf{c}^{i,L}$ and $\mathbf{h}^{i,L}$. Let K_i^L be the closure of $\mathbf{b}_i^L(N(\mathbf{v}, \mathbf{w}) \cap D_\Omega) \times \mathbf{c}^{i,L}(T \times N(\mathbf{v}, \mathbf{w}) \cap D_\Omega) \times \mathbf{h}^{i,L}(T \times N(\mathbf{v}, \mathbf{w}) \cap D_\Omega)$. Note that K_i^L is a bounded subset of $F_i^L \times \mathbb{R}^{n_i^L(n_u + n_x)} \times \mathbb{R}^{n_i^L}$, using the Lipschitz continuity of $\mathbf{b}_i^L$ and the boundedness property of $\mathbf{c}^{i,L}$ and $\mathbf{h}^{i,L}$. Thus, K_i^L is a compact subset of

$F_i^L \times \mathbb{R}^{n_i^L(n_u+n_x)} \times \mathbb{R}^{n_i^L}$, and thus there exists a $L_q > 0$ such that

$$\begin{aligned}\left|q_i^L(t,\mathbf{v}_1,\mathbf{w}_1) - q_i^L(t,\mathbf{v}_2,\mathbf{w}_2)\right| \le L_q\big(&\left\|\mathbf{b}_i^L(\mathbf{v}_1,\mathbf{w}_1) - \mathbf{b}_i^L(\mathbf{v}_2,\mathbf{w}_2)\right\|_\infty + \\ &\left\|\mathbf{c}^{i,L}(t,\mathbf{v}_1,\mathbf{w}_1) - \mathbf{c}^{i,L}(t,\mathbf{v}_2,\mathbf{w}_2)\right\|_\infty + \\ &\left\|\mathbf{h}^{i,L}(t,\mathbf{v}_1,\mathbf{w}_1) - \mathbf{h}^{i,L}(t,\mathbf{v}_2,\mathbf{w}_2)\right\|_\infty\big),\end{aligned}$$

for all $(\mathbf{v}_1,\mathbf{w}_1)$ and $(\mathbf{v}_2,\mathbf{w}_2)$ in $N(\mathbf{v},\mathbf{w}) \cap D_\Omega$ and all $t \in T$. Using the Lipschitz continuity of $\mathbf{b}_i^L$ and uniform local Lipschitz continuity of $\mathbf{c}^{i,L}$ and $\mathbf{h}^{i,L}$, we see that q_i^L is locally Lipschitz continuous on D_Ω, uniformly on T. Similar analysis establishes that q_i^U has the same property.

Consequently, we can apply many different classes of numerical integration methods to solve the IVP (5.3); as mentioned earlier this includes implicit and explicit Runge-Kutta and linear multistep methods. The benefit of this is that we can rely on the sophisticated automatic error control of implementations of these methods by adaptive time stepping. The result is that a highly accurate numerical solution of the IVP (5.3) can be obtained, although without outward rounding techniques, these solutions will not be "validated" in the sense of [107].

5.4.2 Parametric affine relaxations

It was assumed in §5.4.1 that we had parameterized convex and concave piecewise affine relaxations of the original dynamics $\mathbf{f}$. Fortunately, the methods discussed in Ch. 3 provide us with this information.

First, let $D^{\mathbb{I}} = \{(\mathbf{v}',\mathbf{w}') \in D \times D : [\mathbf{v}',\mathbf{w}'] \subset D\}$, and let $[\mathbf{v}_u,\mathbf{w}_u] \subset D_u$ be an interval enclosure of U (since U is polyhedral, this could be obtained, for instance, by the procedure in Definition 4 in [168]; see also Algorithm 3 in §6.4.1). In the context of Proposition 3.2.2, let $X = T \times D^{\mathbb{I}}$, $Z = T \times D_u \times D$, and $Z_D : (t,\mathbf{v}',\mathbf{w}') \mapsto [t,t] \times [\mathbf{v}_u,\mathbf{w}_u] \times [\mathbf{v}',\mathbf{w}']$. Then we can construct affine relaxations on $[t,t] \times [\mathbf{v}_u,\mathbf{w}_u] \times [\mathbf{v}',\mathbf{w}']$ of each component f_i of $\mathbf{f}$ in a manner similar to what was demonstrated in Example 3.4.1. The result is that we obtain affine under and overestimators which satisfy

$$\mathbf{f}_i^{al}(t,\mathbf{v}',\mathbf{w}')^{\mathrm{T}}\mathbf{y}_t + f_i^{bl}(t,\mathbf{v}',\mathbf{w}') \le f_i(\mathbf{y}_t) \le \mathbf{f}_i^{au}(t,\mathbf{v}',\mathbf{w}')^{\mathrm{T}}\mathbf{y}_t + f_i^{bu}(t,\mathbf{v}',\mathbf{w}'),$$

for all $\mathbf{y}_t = (t, \mathbf{p}, \mathbf{z}) \in [t, t] \times [\mathbf{v}_u, \mathbf{w}_u] \times [\mathbf{v}', \mathbf{w}']$. Since we only care about relaxations of $f_i(t, \cdot, \cdot)$, we can rearrange the above expressions and define

$$\begin{aligned}
\widetilde{\mathbf{c}}^{i,L} &: (t, \mathbf{v}', \mathbf{w}') \mapsto \left(f^{al}_{i,2}(t, \mathbf{v}', \mathbf{w}'), f^{al}_{i,3}(t, \mathbf{v}', \mathbf{w}'), \dots, f^{al}_{i,1+n_u+n_x}(t, \mathbf{v}', \mathbf{w}')\right),\\
\widetilde{h}^{i,L} &: (t, \mathbf{v}', \mathbf{w}') \mapsto f^{bl}_i(t, \mathbf{v}', \mathbf{w}') + (t) f^{al}_{i,1}(t, \mathbf{v}', \mathbf{w}'),\\
\widetilde{\mathbf{c}}^{i,U} &: (t, \mathbf{v}', \mathbf{w}') \mapsto \left(f^{au}_{i,2}(t, \mathbf{v}', \mathbf{w}'), f^{au}_{i,3}(t, \mathbf{v}', \mathbf{w}'), \dots, f^{au}_{i,1+n_u+n_x}(t, \mathbf{v}', \mathbf{w}')\right),\\
\widetilde{h}^{i,U} &: (t, \mathbf{v}', \mathbf{w}') \mapsto f^{bu}_i(t, \mathbf{v}', \mathbf{w}') + (t) f^{au}_{i,1}(t, \mathbf{v}', \mathbf{w}').
\end{aligned}$$

The result is that $\widetilde{\mathbf{c}}^{i,L}$, $\widetilde{h}^{i,L}$, $\widetilde{\mathbf{c}}^{i,U}$, and $\widetilde{h}^{i,U}$ are locally Lipschitz continuous mappings on X which satisfy for all $(t, \mathbf{v}', \mathbf{w}') \in X$

$$\widetilde{\mathbf{c}}^{i,L}(t, \mathbf{v}', \mathbf{w}')^{\mathrm{T}}\mathbf{y} + \widetilde{h}^{i,L}(t, \mathbf{v}', \mathbf{w}') \le f_i(t, \mathbf{y}) \le \widetilde{\mathbf{c}}^{i,U}(t, \mathbf{v}', \mathbf{w}')^{\mathrm{T}}\mathbf{y} + \widetilde{h}^{i,U}(t, \mathbf{v}', \mathbf{w}'),$$

for all $\mathbf{y} = (\mathbf{p}, \mathbf{z}) \in [\mathbf{v}', \mathbf{w}'] \times [\mathbf{v}_u, \mathbf{w}_u]$.

Finally, the ultimate goal is to obtain affine underestimators of $f_i(t, \cdot, \cdot)$ on $U \times P^L_i([\mathbf{v}, \mathbf{w}] \cap G)$ (and affine overestimators on $U \times P^U_i([\mathbf{v}, \mathbf{w}] \cap G)$). To this end, since $P^L_i([\mathbf{v}, \mathbf{w}] \cap G)$ is a polyhedron (see Eqn. (5.10)), we can apply the interval-tightening operation in Algorithm 3 of §6.4.1 to the interval $[\mathbf{v}, \mathbf{w}]$ in order to obtain a "tighter" interval enclosure of $P^L_i([\mathbf{v}, \mathbf{w}] \cap G)$. Let the endpoints of the interval defined this way be $\mathbf{v}^L_i : D_\Omega \to \mathbb{R}^{n_x}$ and $\mathbf{w}^L_i : D_\Omega \to \mathbb{R}^{n_x}$ which, by the properties of the interval tightening operator (see Proposition 6.4.1), are Lipschitz continuous and satisfy $[\mathbf{v}^L_i(\mathbf{v}, \mathbf{w}), \mathbf{w}^L_i(\mathbf{v}, \mathbf{w})] \supset P^L_i([\mathbf{v}, \mathbf{w}] \cap G)$, for all $(\mathbf{v}, \mathbf{w}) \in D_\Omega$. Then, assuming that $(\mathbf{v}^L_i, \mathbf{w}^L_i)$ is a mapping into $D^{\mathbb{I}}$, the composite maps on $T \times D_\Omega$

$$\begin{aligned}
\mathbf{c}^{i,L} &: (t, \mathbf{v}, \mathbf{w}) \mapsto \widetilde{\mathbf{c}}^{i,L}(t, \mathbf{v}^L_i(\mathbf{v}, \mathbf{w}), \mathbf{w}^L_i(\mathbf{v}, \mathbf{w})),\\
h^{i,L} &: (t, \mathbf{v}, \mathbf{w}) \mapsto \widetilde{h}^{i,L}(t, \mathbf{v}^L_i(\mathbf{v}, \mathbf{w}), \mathbf{w}^L_i(\mathbf{v}, \mathbf{w})),\\
\mathbf{c}^{i,U} &: (t, \mathbf{v}, \mathbf{w}) \mapsto \widetilde{\mathbf{c}}^{i,U}(t, \mathbf{v}^L_i(\mathbf{v}, \mathbf{w}), \mathbf{w}^L_i(\mathbf{v}, \mathbf{w})),\\
h^{i,U} &: (t, \mathbf{v}, \mathbf{w}) \mapsto \widetilde{h}^{i,U}(t, \mathbf{v}^L_i(\mathbf{v}, \mathbf{w}), \mathbf{w}^L_i(\mathbf{v}, \mathbf{w})),
\end{aligned}$$

are locally Lipschitz continuous. Thus, applying Lemma 2.5.1, we see that $\mathbf{c}^{i,L}$, $h^{i,L}$, $\mathbf{c}^{i,U}$, and $h^{i,U}$ are locally Lipschitz continuous on D_Ω, uniformly on T. In addition, they are continuous and T is compact, and so by Lemma 5.4.1, the extra local boundedness property holds.

By the assumption on the form of the relaxations f_i^{cv} and f_i^{cc}, the extension of the parametric affine relaxations theory discussed in §3.7.1 can be used, where multiple affine relaxations can be calculated giving piecewise affine convex and concave relaxations. However, in the examples in §5.5, the basic affine relaxation method is used; in general the improvement (if any) in the state bounds resulting from using multiple affine relaxations does not justify the extra computational cost. Thus, piecewise affine under and overestimators consisting of only two affine relaxations each (i.e. $n_i^L = n_i^U = 2$ for all i) are used, where one of the affine relaxations comes from the interval bounds. One modification of the theory from Ch. 3 is the use of different parameterized affine relaxations for bivariate multiplication (resulting from an older implementation of the ideas in Ch. 3). That is, in Table 3.3, the rules for the library function $f : (y_1, y_2) \mapsto y_1 y_2$ are changed to

$$\mathbf{f}^{al} : (\mathbf{y}^L, \mathbf{y}^U) \mapsto (y_2^L, y_1^L), \qquad f^{bl} : (\mathbf{y}^L, \mathbf{y}^U) \mapsto -y_1^L y_2^L,$$
$$\mathbf{f}^{au} : (\mathbf{y}^L, \mathbf{y}^U) \mapsto (y_2^L, y_1^U), \qquad f^{bu} : (\mathbf{y}^L, \mathbf{y}^U) \mapsto -y_1^U y_2^L.$$

5.4.3 Numerical solution

Chapter 4 discusses the problem of trying to calculate a solution of an initial value problem defined by a parametric linear program. One issue that can occur is that the effective domain may not be an open set, and most numerical methods for the integration of a system of ODEs assume that the domain of the dynamics is an open set. In the present setting, a similar complication is possible; any solutions of the IVP (5.3) must satisfy $\{(\mathbf{x}^L(t), \mathbf{x}^U(t)) : t \in T\} \subset D_\Omega$, and D_Ω may not be an open set. However, in our experience, there is little complication when using the "direct" method of solving IVP (5.3) numerically, where we use an LP solver to evaluate the functions $(\mathbf{q}^L, \mathbf{q}^U)$ in the derivative evaluator of a numerical integration routine (such as one mentioned in §2.5). The main concern is how to most efficiently evaluate $(\mathbf{q}^L, \mathbf{q}^U)$ defined in Eqns. (5.8) and (5.9).

To address this concern, consider the reformulation of the LPs (5.8) and (5.9) as standard form LPs. Let $\mathbf{z} = (t, \mathbf{v}, \mathbf{w}) \in T \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ denote the vector parameterizing these problems. These programs share the same following formulation when put in standard form via elementary techniques (see, for instance, Ch. 1 of [25]):

$$q(\mathbf{z}) = \min\{\mathbf{c}^{\mathrm{T}}\mathbf{y} : \mathbf{M}(\mathbf{z})\mathbf{y} = \mathbf{d}(\mathbf{z}), \mathbf{y} \geq \mathbf{0}\}, \tag{5.12}$$

where, for some $(m,n) \in \mathbb{N}^2$, $\mathbf{d} : T \times D_\Omega \to \mathbb{R}^m$ and $\mathbf{M} : T \times D_\Omega \to \mathbb{R}^{m \times n}$ are continuous mappings, under the assumption that f_i^{cv} and f_i^{cc} are continuous. Note that the parameterization occurs in both the right-hand side of the constraints and the constraint matrix, and thus the so-called technology matrix case of parametric linear programming results. In general, this kind of parametric dependence can lead to a discontinuous objective value (see [208]). However, as already shown in §2.4, this cannot happen for the problems of interest, and considering the origin of the reformulation, we can show that these problems can be handled in a fairly efficient way.

First, a basis $B \subset \{1, \dots, n\}$ is an index set which describes a vertex of the feasible set of an LP. For the LP (5.12), a basis will have m elements, and so let $\mathbf{M}_B(\mathbf{z})$ be the square submatrix formed by taking the columns of $\mathbf{M}(\mathbf{z})$ which correspond to elements of B, called a basis matrix. Similarly, given a vector $\mathbf{v} \in \mathbb{R}^n$, let $\mathbf{v}_B \in \mathbb{R}^m$ be defined by taking the components of $\mathbf{v}$ which correspond to elements of B.

Next, note that for all $\mathbf{z} \in T \times D_\Omega$, the linear programs (5.8) and (5.9) have solutions because, in this case, the feasible sets are nonempty and bounded. Thus the equivalent problems in standard form must also have solutions. This implies that there exist bases which each describe a vertex which is optimal for the reformulated problems. These bases are called optimal bases. In terms of the standard form LP (5.12), a basis B is optimal if the corresponding basis matrix $\mathbf{M}_B(\mathbf{z})$ satisfies the algebraic conditions

$$(\mathbf{M}_B(\mathbf{z}))^{-1}\mathbf{d}(\mathbf{z}) \geq \mathbf{0}, \tag{5.13}$$

$$\mathbf{c}^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}(\mathbf{M}_B(\mathbf{z}))^{-1}\mathbf{M}(\mathbf{z}) \geq \mathbf{0}^{\mathrm{T}}, \tag{5.14}$$

referred to as primal and dual feasibility, respectively. When solving an LP with the simplex algorithm, the algorithm can be "warm-started" by providing a basis which is either primal or dual feasible; the algorithm terminates much more quickly than if it was cold-started (if it had to go through Phase I first).

Thus, it is desirable to know, given an optimal basis B, whether the left-hand side of the inequality in either of (5.13) or (5.14) may be continuous on an open subset of $T \times D_\Omega$ containing $\mathbf{z}$. If this is the case, then a given optimal basis B that satisfies either (5.13) and/or (5.14) with strict inequality will remain primal and/or dual feasible for some finite amount of time. Consequently, in the course of numerical integration of (5.3), for many steps we can warm-start the simplex algorithm to solve the LPs. This will speed up the solution time immensely. The number of steps where a basis is unavailable to warm-start simplex may be small relative to the overall number of steps taken.

This is indeed the case: given a basis B such that $\mathbf{M}_B(\mathbf{z})$ is optimal, there is an open subset of $T \times D_\Omega$ containing $\mathbf{z}$ on which the left-hand side of the inequality in either of (5.13) or (5.14) is continuous. To see this, we can use Cramer's rule; see §4.4 of [191]. For $\mathbf{S} \in \mathbb{R}^{m \times m}$, $\mathbf{e} \in \mathbb{R}^m$, the vector $\mathbf{y} = \mathbf{S}^{-1}\mathbf{e}$ is given componentwise by

$$y_j = \frac{\det(\mathbf{T}_j)}{\det(\mathbf{S})},$$

where $\mathbf{T}_j$ is the matrix formed by replacing the j^{th} column of $\mathbf{S}$ with $\mathbf{e}$. Since the determinant of a matrix is continuous with respect to the entries of the matrix, it is a simple application of Cramer's rule to see that the left-hand side of the inequalities (5.13) and (5.14) are continuous on the set of those $\mathbf{z}'$ such that $\mathbf{M}_B(\mathbf{z}')$ is invertible. Further, the set of those $\mathbf{z}'$ such that $\mathbf{M}_B(\mathbf{z}')$ is invertible is an open set containing $\mathbf{z}$ noting that it is the preimage of $(-\infty, 0) \cup (0, +\infty)$, an open set, under the continuous mapping $\det(\mathbf{M}_B(\cdot))$. Consequently, the direct method of solving the IVP (5.3) numerically can be fairly efficient.

5.4.4 Complexity

This section considers the computational complexity of the proposed bounding method. As the general numerical implementation of the method involves the solution of an initial value problem in ODEs in which the dynamics are defined by the solution of linear programs, the complexity of the method will depend on the choice of numerical integrator and linear program solver. Some observations follow. For this discussion, "cost" is roughly measured in terms of floating-point operations or just "operations."

First, the complexity of computing a solution of an IVP in Lipschitz ODEs with a general numerical integration method is somewhat out of the scope of this chapter; it is a fairly open problem and has interesting ties to deeper questions in computational complexity theory. For a more theoretical discussion see [93, 94].

A more practical observation is that, beside evaluating the dynamics, the dominant cost at each step in most implicit numerical integration methods for stiff systems is the matrix factorization required for Newton iteration (in the context of the backward differentiation formulae (BDF), see §5.2.2 of [34]). Consequently, there is, in general, an order $(2n_x)^3$ cost associated with the numerical integration of IVP (5.3). However, most implementations of the BDF, for instance, will deploy a deferred Jacobian, and avoid matrix factorization at each step; see §5.2.2 of [34] and §6.5 of [105].

The focus of the rest of this section is to analyze the complexity of evaluating the right-hand sides defining the IVP (5.3). A more complete answer to this question can be found. This depends heavily on the standard-form LPs (5.12) which define the dynamics (5.8) and (5.9). First, a bound on the size of these standard-form LPs is needed. This depends on n_x, n_u, m_u (the number of halfspaces required to represent the polyhedral set of control values U), m_g (the number of halfspaces required to represent the polyhedral *a priori* enclosure G), and n_i^L or n_i^U (the number of affine functions that make up the piecewise affine estimators f_i^{cv} or f_i^{cc}). To define q_i^L, one can check that the number of constraints in the corresponding standard-form LP (5.12) is bounded above by $2n_x + m_g + m_u + 1 + n_i^L$, and that the number of variables is bounded above by $4n_x + 2n_u + m_g + m_u + 2 + n_i^L$ (this is obtained by a rather slavish addition of slack variables, dummy variables, and splitting "free" variables into nonnegative and nonpositive parts). In addition, evaluation of $\mathbf{d}$, the right-hand side of the constraints of LP (5.12), involves evaluating z_i^m (or z_i^M), which itself requires the solution of a linear program.

Citing the celebrated result that there exist polynomial time algorithms for linear programming (see Ch. 8 of [25]), we can establish that one component $q_i^{L/U}$ of the dynamics defining IVP (5.3) can be evaluated with polynomial cost (with respect to n_x, n_u, m_u, m_g, n_i^L, and n_i^U). However, as indicated in §5.4.3, in practice one might obtain better performance using the simplex algorithm and attempting to warm-start the method using a basis recorded from the previous function evaluation. In summary, however the linear programs are solved, evaluating the dynamics requires the solution of $4n_x$ LPs since the IVP is a system in $\mathbb{R}^{2n_x}$. Compared to the interval arithmetic-based method in [168], the solution of these linear programs is the most significant increase in cost.

The last step is to consider the complexity of evaluating the piecewise affine under and overestimators. The analysis in §3.5 establishes that this can be achieved with a cost that is a scalar multiple of the cost of evaluating the original right-hand side function $\mathbf{f}$. This evaluation is repeated $\sum_{i=1}^{n_x} n_i^L + n_i^U$ times, corresponding to each piece of the piecewise affine under and overestimators f_i^{cv}, f_i^{cc}, $i \in \{1, \ldots, n_x\}$.

5.5 Examples

This section assesses the performance of a MATLAB implementation of the proposed bounding method, using an implicit linear multistep integration method (`ode113`, see [172]) and CPLEX [85] to solve the necessary linear programs. MATLAB release r2011b is used on a workstation with a 3.07 GHz Intel Xeon processor.

5.5.1 Polyhedral control values versus interval hull

The simple reaction network

$$\begin{aligned} &\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C}, \\ &\mathrm{A} + \mathrm{C} \rightarrow \mathrm{D}, \end{aligned}$$

is considered to demonstrate the ability of the system (5.3) to utilize a polyhedral set U. The dynamic equations governing the evolution of the species concentrations $\mathbf{x} = (x_A, x_B, x_C, x_D)$ in a closed system are

$$\begin{aligned} \dot{x}_A &= -u_1 x_A x_B - u_2 x_A x_C, \\ \dot{x}_B &= -u_1 x_A x_B, \\ \dot{x}_C &= u_1 x_A x_B - u_2 x_A x_C, \\ \dot{x}_D &= u_2 x_A x_C. \end{aligned} \tag{5.15}$$

The goal to estimate the reachable set on $T = [0, 0.1]$ (s), with $X_0 = \{\mathbf{x}_0 = (1, 1, 0, 0)\}$ (M) and rate parameters $\mathbf{u} = (u_1, u_2) \in \mathcal{U}$, with

$$U = \left\{ \mathbf{k} \in \mathbb{R}^2 : \widehat{\mathbf{k}} \le \mathbf{k} \le 10\widehat{\mathbf{k}},\ k_1 + 2.5k_2 = 550 \right\},$$

where $\widehat{\mathbf{k}} = (50, 20)$. A polyhedral enclosure G can be determined from considering the stoichiometry of the system and other physical arguments such as mass balance; see [167]

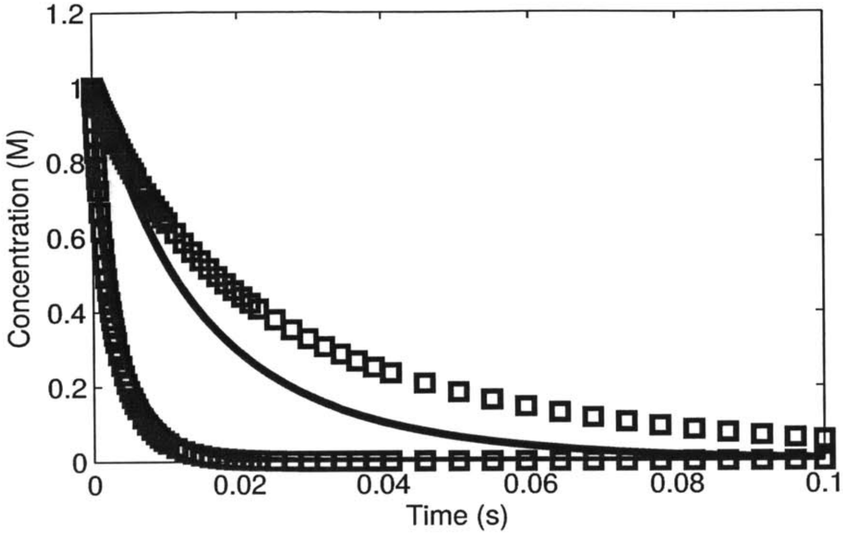


Figure 5-1: Interval bounds for x_A (from system (5.15)) computed from the system of ODEs with LPs embedded (5.3). Bounds for polyhedral U are plotted with solid lines, while bounds for the interval hull of U are plotted with boxes.

for more details. For this system, we have

$$\begin{aligned} G &= \{\mathbf{z} \in \mathbb{R}^4 : \mathbf{0} \leq \mathbf{z} \leq \bar{\mathbf{x}}, \mathbf{M}\mathbf{z} = \mathbf{M}\mathbf{x}_0\}, \quad \text{with} \\ \mathbf{M} &= \begin{bmatrix} 1 & -1 & 0 & 1 \\ -1 & 2 & 1 & 0 \end{bmatrix}, \\ \bar{\mathbf{x}} &= (1, 1, 1, 0.5). \end{aligned}$$

Bounds on the states $\mathbf{x}$ are calculated using the system of ODEs with LPs embedded (5.3), first using U as is, and again using the interval hull of U. The difference can be quite apparent; see Fig. 5-1. The ability to use a polyhedral set of admissible control values distinguishes this method from previous work in [168, 176], for example. In each case, the method takes approximately 1.3 seconds.

5.5.2 Comparison with previous implementation

The enzyme reaction network considered in Example 2 of [168] is used here to demonstrate the effectiveness of the system (5.3) in producing tight state bounds for an uncertain dynamic system. The reaction network is

$$\mathrm{A}+\mathrm{F} \rightleftharpoons \mathrm{F}:\mathrm{A} \rightarrow \mathrm{F}+\mathrm{A}',$$
$$\mathrm{A}'+\mathrm{R} \rightleftharpoons \mathrm{R}:\mathrm{A}' \rightarrow \mathrm{R}+\mathrm{A}.$$

The dynamic equations governing the evolution of the species concentrations

$$\mathbf{x} = (x_{\mathrm{F}}, x_{\mathrm{A}}, x_{\mathrm{F:A}}, x_{\mathrm{A'}}, x_{\mathrm{R}}, x_{\mathrm{R:A'}})$$

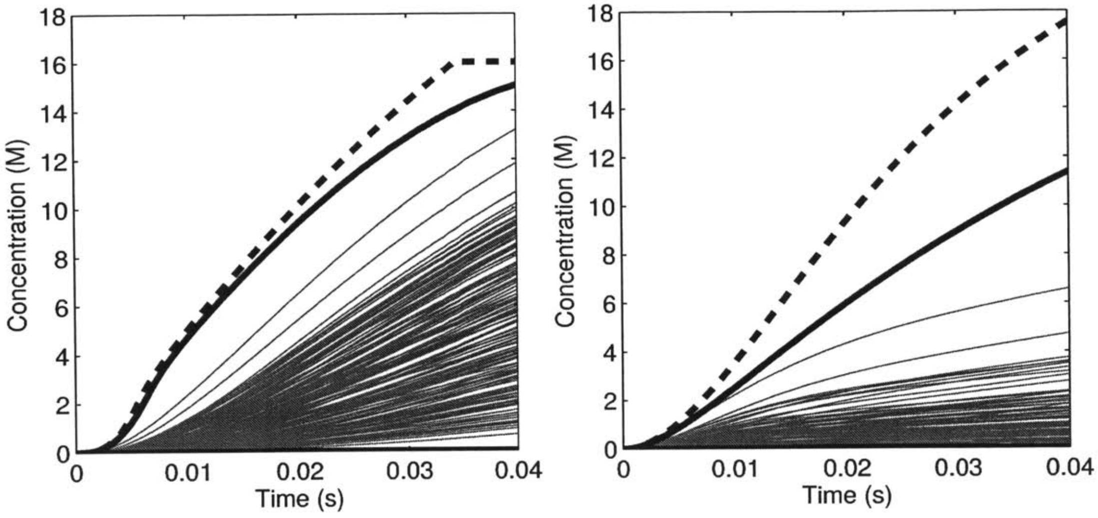
in a closed system are

$$\begin{aligned}
\dot{x}_{\mathrm{F}} &= -u_1 x_{\mathrm{F}} x_{\mathrm{A}} + u_2 x_{\mathrm{F:A}} + u_3 x_{\mathrm{F:A}}, \qquad (5.16)\\
\dot{x}_{\mathrm{A}} &= -u_1 x_{\mathrm{F}} x_{\mathrm{A}} + u_2 x_{\mathrm{F:A}} + u_6 x_{\mathrm{R:A'}},\\
\dot{x}_{\mathrm{F:A}} &= u_1 x_{\mathrm{F}} x_{\mathrm{A}} - u_2 x_{\mathrm{F:A}} - u_3 x_{\mathrm{F:A}},\\
\dot{x}_{\mathrm{A'}} &= u_3 x_{\mathrm{F:A}} - u_4 x_{\mathrm{A'}} x_{\mathrm{R}} + u_5 x_{\mathrm{R:A'}},\\
\dot{x}_{\mathrm{R}} &= -u_4 x_{\mathrm{A'}} x_{\mathrm{R}} + u_5 x_{\mathrm{R:A'}} + u_6 x_{\mathrm{R:A'}},\\
\dot{x}_{\mathrm{R:A'}} &= u_4 x_{\mathrm{A'}} x_{\mathrm{R}} - u_5 x_{\mathrm{R:A'}} - u_6 x_{\mathrm{R:A'}}.
\end{aligned}$$

The goal to estimate the reachable sets on $T = [0, 0.04]$ (s), with $\mathbf{x}_0 = (20, 34, 0, 0, 16, 0)$ (M), $X_0 = \{\mathbf{x}_0\}$, and rate parameters $\mathbf{u} = (u_1, \dots, u_6) \in \mathcal{U}$, with $U = \left\{\mathbf{k} \in \mathbb{R}^6 : \widehat{\mathbf{k}} \leq \mathbf{k} \leq 10\widehat{\mathbf{k}}\right\}$ and $\widehat{\mathbf{k}} = (0.1, 0.033, 16, 5, 0.5, 0.3)$. For this system, a polyhedral enclosure G is

$$\begin{aligned}
G &= \{\mathbf{z} \in \mathbb{R}^6 : \mathbf{0} \leq \mathbf{z} \leq \bar{\mathbf{x}}, \mathbf{M}\mathbf{z} = \mathbf{M}\mathbf{x}_0\}, \quad \text{with}\\
\mathbf{M} &= \begin{bmatrix} -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & 1 & 0 & 1 & -1 & 0 \end{bmatrix},\\
\bar{\mathbf{x}} &= (20, 34, 20, 34, 16, 16).
\end{aligned}$$

The state bounds resulting from the solution of (5.3) and the interval arithmetic-based implementation used in [168] are similar; the bounds resulting from the solution of (5.3) are at least as tight as those in [168]. See Fig. 5-2. As demonstrated by Fig. 5-2b, using affine relaxations can lead to a significant improvement in the bounds. The proposed method takes



(a) Species R : A′

(b) Species A′

Figure 5-2: Interval bounds computed from the system of ODEs with LPs embedded (5.3) (solid black lines) and from the implementation in [168] (dashed black lines). Solutions of (5.16) for constant $\mathbf{u} \in \mathcal{U}$ are plotted with thin solid lines.

approximately 2.2 seconds, whereas a comparable MATLAB implementation of the method in [168] takes 0.65 seconds.

5.6 Conclusions

This chapter has considered the problem of bounding the reachable set of a nonlinear dynamic system pointwise in time. The approach taken is an implementation of the theory in [168], which in turn is based on the theory of differential inequalities. The implementation leads to a system of ordinary differential equations depending on parametric linear programs. Thus, this chapter also analyzes how numerically tractable such a system is. The new implementation yields tighter bounds than the previous one, especially when the admissible controls take values in a compact polyhedron.

Chapter 6

Efficient polyhedral enclosures for the reachable sets of nonlinear control systems

6.1 Introduction

This chapter considers estimating the reachable set of the initial value problem in ordinary differential equations

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

where the initial condition is in some known set and the inputs $\mathbf{u}$ vary in some known set (see §6.2 for a formal problem statement). The time-varying inputs $\mathbf{u}$ can model noise, disturbances, or control inputs; estimating the reachable set in these cases is critical to robust MPC [16], fault detection [108] and global optimization of dynamic systems [177]. This chapter will focus on the construction of a time-varying polyhedral enclosure of the reachable set. Specifically, given a matrix $\mathbf{A}$, an auxiliary initial value problem in ordinary differential equations is constructed whose solution yields the function $\mathbf{b}$, such that the set $\{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}(t)\}$ contains the values of the solutions of the dynamic system at all times t.

The major contribution of this chapter is as follows. First, the construction of the auxiliary system of bounding equations closely resembles the bounding method in [168] which produces interval bounds (i.e., component-wise upper and lower bounds) very quickly. That work is itself an extension of differential inequality-based comparison theorems in, for instance, [72]. The extension of these comparison theorems to the construction of polyhedral bounds is a significant improvement, allowing bounds with the increased flexibility of polyhedra (as opposed to intervals) to be calculated with powerful and efficient methods for numerical integration. The general concept of previous implementations of polyhedron-based bounding methods, such as those in [7, 39, 40, 64], is to "move" the faces of the approximating polyhedra in accordance with the maximum value of the dynamics on that face, which is similar to the theory in this chapter. However, these previous methods manually implement the time-stepping, which contrasts with the proposed method, which takes advantage of established codes for numerical integration, and thus benefits from their handling of step size to control errors to a desired tolerance. The methods in this chapter could be used to implement the step forward in time in these previous methods; however, the work presented here stands on its own as an effective method on the overall time scale of interest, as demonstrated by the numerical experiments in §6.5. Further, these examples provide new insight into "intelligent" choices for the matrix $\mathbf{A}$ which defines the polyhedral approximation.

Another significant contribution of this work is its ability to distinguish meaningfully between time-varying inputs and constant, but unknown, parameters. This contrasts with the previous work involving comparison theorems, such as in [74, 97, 152, 167, 168]. As noted in [168], comparison theorems in general take into account time-varying uncertainty, which can be an advantage or disadvantage depending on one's perspective or the model of interest. In contrast, methods based on parametric Taylor-models, such as in [37, 107], intrinsically handle constant, but uncertain parameter inputs. Since the present work is inspired by comparison theorems, it is natural that it should be able to handle time-varying uncertainty. But the use of polyhedra allows one to propagate affine relaxations of the states with respect to the parameters by treating the unknown parameters as extra states, but with time derivatives equal to zero. This explicitly enforces these uncertain inputs to be constant, and as demonstrated by an example in §6.5.3, this leads to an improvement in the bounds.

Other work that should be mentioned involves the computation of the reachable set or an approximation of it through a level set that evolves in time, such as in [121]. However, this involves the solution of a Hamilton-Jacobi-type partial differential equation, and consequently is more computationally demanding than the comparison theorem based methods, including this work. This is noted in [84], in which the solution of matrix exponentials is favored over the solution of partial differential equations to obtain a polyhedral approximation of the reachable set of a dynamic system. However, that work only considers linear dependence on the inputs, and weakly nonlinear dynamics.

The rest of this chapter is organized as follows. Section 6.2 gives a rigorous problem statement. Section 6.3 is split into two subsections. Section 6.3.1 states and proves one of the main results of this chapter, Theorem 6.3.1. This theorem is a general result stating conditions under which a polyhedral-valued mapping is an enclosure of the reachable set. Section 6.3.2 provides a specific instance of this theory in Corollary 6.3.3, which states that the solution of an auxiliary initial value problem yields polyhedral bounds. This leads to a numerically implementable method for constructing polyhedral bounds discussed in §6.4. One of the main goals of §6.4 is to establish that the auxiliary problem satisfies basic assumptions to be amenable to solution with general classes of numerical integration methods. Section 6.5 demonstrates a numerical implementation of the proposed bounding method on a few examples. Section 6.6 concludes with some final remarks.

6.2 Problem statement

Let $(n_x, n_u) \in \mathbb{N}^2$, nonempty interval $T = [t_0, t_f] \subset \mathbb{R}$, $D_x \subset \mathbb{R}^{n_x}$, and $D_u \subset \mathbb{R}^{n_u}$ be given. For $U \subset D_u$, let the set of time-varying inputs be

$$\mathcal{U} = \left\{ \mathbf{u} \in L^1(T, \mathbb{R}^{n_u}) : \mathbf{u}(t) \in U, a.e.\ t \in T \right\},$$

and let the set of possible initial conditions be $X_0 \subset D_x$. Given $\mathbf{f} : T \times D_u \times D_x \to \mathbb{R}^{n_x}$, the problem of interest is the initial value problem in ODEs

$$\dot{\mathbf{x}}(t, \mathbf{u}) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t, \mathbf{u})), \quad a.e.\ t \in T, \tag{6.1a}$$

$$\mathbf{x}(t_0, \mathbf{u}) \in X_0. \tag{6.1b}$$

For given $\mathbf{u} \in \mathcal{U}$, a solution is an absolutely continuous mapping $\mathbf{x}(\cdot, \mathbf{u}) : T \to D_x$ which satisfies Equations (6.1). The goal of this chapter is to construct a polyhedral-valued mapping $B : T \rightrightarrows \mathbb{R}^{n_x}$ such that for all $\mathbf{u} \in \mathcal{U}$ and any solution $\mathbf{x}(\cdot, \mathbf{u})$ (if one exists for this $\mathbf{u}$), $\mathbf{x}(t, \mathbf{u}) \in B(t)$, for all $t \in T$. Specifically, given a $m \in \mathbb{N}$ and a matrix $\mathbf{A} \in \mathbb{R}^{m \times n_x}$, the goal is to find $\mathbf{b} : T \to \mathbb{R}^m$ such that $B(t) = \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}(t)\}$. This mapping B will be referred to as polyhedral bounds, or just bounds.

6.3 Bounding theory

This section will give a general theorem for polyhedral bounds, which requires a specific assumption, then discuss how to satisfy this assumption.

6.3.1 General theory

Lemma 6.3.1. *Let $T \subset \mathbb{R}$ be an interval, $b : T \to \mathbb{R}$ be absolutely continuous, $\mathbf{x} : T \to \mathbb{R}^n$ be absolutely continuous, and $\mathbf{a} \in \mathbb{R}^n$. Then the real-valued function $g : t \mapsto \max\{0, \mathbf{a}^{\mathrm{T}}\mathbf{x}(t) - b(t)\}$ is absolutely continuous. Further, for almost all t such that $\mathbf{a}^{\mathrm{T}}\mathbf{x}(t) > b(t)$ and for any $\mathbf{v}$ such that $\mathbf{a}^{\mathrm{T}}\mathbf{v} \leq \dot{b}(t)$,*

$$\dot{g}(t) \leq \|\mathbf{a}\|_* \|\mathbf{v} - \dot{\mathbf{x}}(t)\| .$$

Proof. Note that $g_1 : t \mapsto \mathbf{a}^{\mathrm{T}}\mathbf{x}(t) - b(t)$ is absolutely continuous, as the sum of absolutely continuous functions. Obviously, $g_2 : t \mapsto 0$ is absolutely continuous, and so g, as the maximum of the two, can be written as $g(t) = 1/2(g_1(t) + g_2(t) + |g_1(t) - g_2(t)|)$. We note this is absolutely continuous, since the composition of a Lipschitz continuous function with an absolutely continuous function is absolutely continuous, and again the sum of absolutely continuous functions is absolutely continuous.

On the set of t such that $\mathbf{a}^{\mathrm{T}}\mathbf{x}(t) > b(t)$, we have $g(\cdot) = \mathbf{a}^{\mathrm{T}}\mathbf{x}(\cdot) - b(\cdot)$. Since g is absolutely continuous, we have that for almost all t such that $\mathbf{a}^{\mathrm{T}}\mathbf{x}(t) > b(t)$, $\dot{g}(t) = \mathbf{a}^{\mathrm{T}}\dot{\mathbf{x}}(t) - \dot{b}(t)$. Thus, for any $\mathbf{v}$ such that $\mathbf{a}^{\mathrm{T}}\mathbf{v} \leq \dot{b}(t)$, $\dot{g}(t) + \mathbf{a}^{\mathrm{T}}\mathbf{v} \leq \mathbf{a}^{\mathrm{T}}\dot{\mathbf{x}}(t) - \dot{b}(t) + \dot{b}(t)$. It follows that $\dot{g}(t) \leq \mathbf{a}^{\mathrm{T}}\dot{\mathbf{x}}(t) - \mathbf{a}^{\mathrm{T}}\mathbf{v}$ and so $\dot{g}(t) \leq \mathbf{a}^{\mathrm{T}}(\dot{\mathbf{x}}(t) - \mathbf{v})$. Finally, from the generalization of the Cauchy-Schwarz inequality (that is, from the definition of the dual norm), we have $\dot{g}(t) \leq \|\mathbf{a}\|_* \|\dot{\mathbf{x}}(t) - \mathbf{v}\|$. □

The following Assumptions and Theorem provide the heart of the general bounding theory. The parallels between this theory and that in [168] should be fairly clear. However, before continuing, it is useful to note a definition of the matrix $\mathbf{A}$ and mappings M_i required in Assumption 6.3.2 which yield the classic interval-based comparison theorem-type results in [72], for instance. This example provides an intuitive geometric interpretation to keep in mind while understanding the general theory. Interval bounds are obtained by letting

$m = 2n_x$ and $\mathbf{A} = [-\mathbf{I}\ \ \mathbf{I}]^{\mathrm{T}}$. Then for $i \in \{1, \dots, m\}$, let $M_i(t, \mathbf{d})$ be the i^{th} lower face of the interval $\{\mathbf{z} : \mathbf{A}\mathbf{z} \le \mathbf{d}\}$ (assuming it is nonempty). Similarly, for $i \in \{n_x + 1, \dots, 2n_x\}$, let $M_i(t, \mathbf{d})$ be the $(i - n_x)^{th}$ upper face of the interval. One can check that these definitions satisfy Assumption 6.3.2. Section 6.3.2 focuses on a more general construction that satisfies Assumption 6.3.2, which will provide the basis for the following numerical developments.

Assumption 6.3.1 is fairly critical to the general theory. However, it is not restrictive at all. It is related to standard assumptions that the solutions of IVP (6.1) for a given $\mathbf{u} \in \mathcal{U}$ and fixed initial condition are unique and that general classes of numerical integration methods are applicable to the IVP (6.1) (see discussion in §2.5).

Assumption 6.3.1. *For any $\mathbf{z} \in D_x$, there exists a neighborhood $N(\mathbf{z})$ and $\alpha \in L^1(T)$ such that for almost every $t \in T$ and every $\mathbf{p} \in U$*

$$\|\mathbf{f}(t, \mathbf{p}, \mathbf{z}_1) - \mathbf{f}(t, \mathbf{p}, \mathbf{z}_2)\| \le \alpha(t) \|\mathbf{z}_1 - \mathbf{z}_2\|,$$

for every $\mathbf{z}_1, \mathbf{z}_2 \in N(\mathbf{z}) \cap D_x$.

Assumption 6.3.2. *Consider the problem stated in §6.2. Assume that for some $m \in \mathbb{N}$, $\mathbf{A} = [\mathbf{a}_i^{\mathrm{T}}] \in \mathbb{R}^{m \times n_x}$, $D_M \subset T \times \mathbb{R}^m$, and $M_i : D_M \rightrightarrows \mathbb{R}^{n_x}$ satisfy the following conditions for each $i \in \{1, \dots, m\}$:*

1. *For any $\mathbf{d} \in \mathbb{R}^m$, if there exists $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\mathbf{A}\mathbf{x}(t, \mathbf{u}) \le \mathbf{d}$ and $\mathbf{a}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$ for some solution $\mathbf{x}(\cdot, \mathbf{u})$ of IVP (6.1), then $(t, \mathbf{d}) \in D_M$ and $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$.*
2. *For any $(t, \mathbf{d}) \in D_M$, there exists a neighborhood $N(\mathbf{d})$ of $\mathbf{d}$, $t' > t$, and $L_M > 0$ such that for any $(s, \mathbf{d}_1)$ and $(s, \mathbf{d}_2)$ in $((t, t') \times N(\mathbf{d})) \cap D_M$ and $\mathbf{z}_1 \in M_i(s, \mathbf{d}_1)$, there exists a $\mathbf{z}_2 \in M_i(s, \mathbf{d}_2)$ such that*

$$\|\mathbf{z}_1 - \mathbf{z}_2\| \le L_M \|\mathbf{d}_1 - \mathbf{d}_2\|_1.$$

Theorem 6.3.1. *Let Assumptions 6.3.1 and 6.3.2 hold. If*

1. *$\mathbf{b} : T \to \mathbb{R}^m$ is absolutely continuous and $B : T \ni t \mapsto \{\mathbf{z} : \mathbf{A}\mathbf{z} \le \mathbf{b}(t)\}$,*
2. *$X_0 \subset B(t_0)$,*
3. *for almost every $t \in T$ and each $i \in \{1, \dots, m\}$, $(t, \mathbf{b}(t)) \in D_M$ and $M_i(t, \mathbf{b}(t)) \subset D_x$,*

4. for almost every $t \in T$ and each $i \in \{1,\dots,m\}$,

$$\mathbf{a}_i^{\mathrm{T}}\mathbf{f}(t,\mathbf{p},\mathbf{z}) \leq \dot{b}_i(t), \quad \forall(\mathbf{p},\mathbf{z}) \in U \times M_i(t,\mathbf{b}(t)),$$

then for all $\mathbf{u} \in \mathcal{U}$ and any solution $\mathbf{x}(\cdot,\mathbf{u})$ of IVP (6.1), $\mathbf{x}(t,\mathbf{u}) \in B(t)$, for all $t \in T$.

Proof. Fix $\mathbf{u} \in \mathcal{U}$. If no solution of IVP (6.1) exists for this $\mathbf{u}$, then the conclusion of the theorem holds trivially. Otherwise, choose some solution and for convenience use the abbreviation $\mathbf{x}(t) \equiv \mathbf{x}(t,\mathbf{u})$. For each $t \in T$ and $i \in \{1,\dots,m\}$, let $g_i(t) = \max\{0, \mathbf{a}_i^{\mathrm{T}}\mathbf{x}(t) - b_i(t)\}$. By Lemma 6.3.1, each g_i is absolutely continuous. It follows that $\mathbf{A}\mathbf{x}(t) \leq \mathbf{b}(t)+\mathbf{g}(t)$. Consequently, $\mathbf{g}(t) = \mathbf{0}$ implies $\mathbf{x}(t) \in B(t)$, and by the contrapositive $\mathbf{x}(t) \notin B(t)$ implies $\mathbf{g}(t) \neq \mathbf{0}$. Thus, for a contradiction, assume that there exists a $\widetilde{t} \in T$ such that $\mathbf{x}(\widetilde{t}) \notin B(\widetilde{t})$. Then the set $T_v = \{t \in T : \|\mathbf{g}(t)\|_1 > 0\}$ is nonempty.

Let $t_1 = \inf T_v$. By Hypothesis 2, $\mathbf{g}(t_0) = \mathbf{0}$ and so by continuity of $\mathbf{g}$, $\|\mathbf{g}(t_1)\|_1 = 0$. Furthermore, there exists $t_2 > t_1$ and index set I such that $g_i(t) = 0$ for $i \notin I$ and $t \in [t_1,t_2)$, and $\mathbf{a}_i^{\mathrm{T}}\mathbf{x}(t) = b_i(t) + g_i(t)$ for $i \in I$ and $t \in [t_1,t_2)$. Explicitly, for each i define $T_i \equiv \{t : g_i(t) > 0\}$. By continuity of $\mathbf{g}$, each T_i is open. Let $I = \{i : t_1 = \inf T_i\}$ (which must be nonempty) and then choose $t_2 > t_1$ such that $(t_1,t_2) \subset \bigcap_{i\in I} T_i$ and $(t_1,t_2) \cap (\bigcup_{i \notin I} T_i) = \varnothing$.

Then by Assumption 6.3.2, $(t, \mathbf{b}(t)+\mathbf{g}(t)) \in D_M$ and $\mathbf{x}(t) \in M_i(t,\mathbf{b}(t)+\mathbf{g}(t))$ for $i \in I$, $t \in [t_1,t_2)$. Without loss of generality, let $N(\mathbf{b}(t_1))$, $t_3 > t_1$, and $L_M > 0$ satisfy Condition 2 of Assumption 6.3.2 at the point $\mathbf{b}(t_1)$, for each $i \in I$. Since $\mathbf{b}$ and $\mathbf{g}$ are continuous, there exists a $t_4 \in (t_1, \min\{t_2,t_3\})$ such that $\mathbf{b}(t)$ and $(\mathbf{b}(t)+\mathbf{g}(t))$ are in $N(\mathbf{b}(t_1))$ for each $t \in (t_1,t_4)$. Along with Hypothesis 3, it follows that for $i \in I$ and almost every $t \in (t_1,t_4)$, there exists an element $\mathbf{z}_i(t) \in M_i(t,\mathbf{b}(t))$ with

$$\|\mathbf{z}_i(t) - \mathbf{x}(t)\| \leq L_M \|\mathbf{g}(t)\|_1. \tag{6.2}$$

Let $N(\mathbf{x}(t_1))$, and $\alpha \in L^1(T)$ satisfy Assumption 6.3.1 at the point $\mathbf{x}(t_1)$. Since $\mathbf{x}$ and $\|\mathbf{g}\|_1$ are continuous, using Inequality (6.2) and the triangle inequality

$$\|\mathbf{z}_i(t) - \mathbf{x}(t_1)\| \leq \|\mathbf{z}_i(t) - \mathbf{x}(t)\| + \|\mathbf{x}(t) - \mathbf{x}(t_1)\|,$$

there exists a $t_5 \in (t_1,t_4)$ such that $\mathbf{z}_i(t), \mathbf{x}(t) \in N(\mathbf{x}(t_1))$, for all $i \in I$ and almost every

$t \in (t_1, t_5)$. Consequently,

$$\|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t))\| \le \alpha(t) \|\mathbf{z}_i(t) - \mathbf{x}(t)\|, \; a.e.\, t \in (t_1, t_5). \tag{6.3}$$

But by Hypothesis 4, $\mathbf{a}_i^{\mathrm{T}}\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) \le \dot{b}_i(t)$ which by Lemma 6.3.1 means

$$\begin{aligned} \dot{g}_i(t) &\le \|\mathbf{a}_i\|_* \|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \dot{\mathbf{x}}(t)\| \\ &= \|\mathbf{a}_i\|_* \|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t))\|. \end{aligned}$$

Combining this with Inequalities (6.2) and (6.3) we have

$$\dot{g}_i(t) \le L_M \alpha(t) \|\mathbf{a}_i\|_* \|\mathbf{g}(t)\|_1, \; a.e.\, t \in (t_1, t_5).$$

Since this holds for each $i \in I$ and $g_i(t) = 0$ for each $i \notin I$,

$$\sum_{i \in I} \dot{g}_i(t) \le L_M \alpha(t) \sum_{i \in I} \|\mathbf{a}_i\|_* \|\mathbf{g}(t)\|_1 = L_M \alpha(t) \sum_{j \in I} \|\mathbf{a}_j\|_* \sum_{i \in I} g_i(t)$$

to which we can apply Gronwall's inequality (see for instance [209]) to get

$$\sum_{i \in I} g_i(t) \le \sum_{i \in I} g_i(t_1) \exp\left(\int_{[t_1, t]} L_M \sum_{j \in I} \|\mathbf{a}_j\|_* |\alpha| \right), \; \forall t \in [t_1, t_5].$$

But since $\sum_i g_i(t_1) = 0$, this yields $\sum_i g_i(t) \le 0$, and since each g_i is nonnegative always and $g_i(t) = 0$ for each $i \notin I$, we have $g_i(t) = 0$ for all i and all $t \in (t_1, t_5) \subset T_v$, which is a contradiction. Since the choices of $\mathbf{u} \in \mathcal{U}$ and corresponding solution were arbitrary, the result follows. □

6.3.2 Implementation

This section describes how to construct the mappings M_i such that they satisfy Assumption 6.3.2 and lead to a numerically implementable bounding method.

Similar to the work in [166, 168], these mappings allow one to use *a priori* information about the solution set of (6.1) in the form of a polyhedral-valued mapping $G : T \rightrightarrows \mathbb{R}^{n_x}$ for which it is known that $\mathbf{x}(t, \mathbf{u}) \in G(t)$, for all $t \in T$ and $\mathbf{u} \in \mathcal{U}$ for which a solution exists. The specific conditions are formalized in the following assumption and subsequent result.

Assumption 6.3.3. *For $m_g \in \mathbb{N}$, let $\mathbf{A}_G \in \mathbb{R}^{m_g \times n_x}$ and $\mathbf{b}_G : T \to \mathbb{R}^{m_g}$. Assume that for all $\mathbf{u} \in \mathcal{U}$ and any solution $\mathbf{x}(\cdot, \mathbf{u})$ of IVP* (6.1)*, $\mathbf{A}_G\mathbf{x}(t, \mathbf{u}) \leq \mathbf{b}_G(t)$, for all $t \in T$.*

Proposition 6.3.2. *Let Assumption 6.3.3 hold. For $m \in \mathbb{N}$, let $\mathbf{A} = [\mathbf{a}_i^{\mathrm{T}}] \in \mathbb{R}^{m \times n_x}$. Let*

$$P_M : (t, \mathbf{d}) \mapsto \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{d}, \mathbf{A}_G\mathbf{z} \leq \mathbf{b}_G(t)\}. \tag{6.4}$$

Then $\mathbf{A}$*,*

$$D_M = \{(t, \mathbf{d}) \in T \times \mathbb{R}^m : P_M(t, \mathbf{d}) \neq \varnothing\}, \text{ and} \tag{6.5}$$

$$M_i : (t, \mathbf{d}) \mapsto \arg\max\{\mathbf{a}_i^{\mathrm{T}}\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{d}, \mathbf{A}_G\mathbf{z} \leq \mathbf{b}_G(t)\}. \tag{6.6}$$

satisfy Assumption 6.3.2.

Proof. To see that Condition 1 of Assumption 6.3.2 holds, choose any $i \in \{1, \ldots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\mathbf{A}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\mathbf{a}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$. Since $\mathbf{A}_G\mathbf{x}(t, \mathbf{u}) \leq \mathbf{b}_G(t)$, it holds that $\mathbf{x}(t, \mathbf{u}) \in P_M(t, \mathbf{d})$, and thus $(t, \mathbf{d}) \in D_M$. Further, since $\mathbf{a}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$, and any $\mathbf{z}$ such that $\mathbf{a}_i^{\mathrm{T}}\mathbf{z} > d_i$ would be infeasible in LP (6.6), we must have $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$.

Next, note that if $P_M(t, \mathbf{d})$ is nonempty, then $M_i(t, \mathbf{d})$ is nonempty for all i ($M_i(t, \mathbf{d})$ is the solution set of a linear program that must be feasible and bounded). Then to see that Condition 2 of Assumption 6.3.2 holds, choose any $(s, \mathbf{d}_1), (s, \mathbf{d}_2) \in D_M$. By definition of D_M and the previous observation, $M_i(s, \mathbf{d}_j)$ is nonempty for $i \in \{1, \ldots, m\}$ and $j \in \{1, 2\}$. Applying Lemma 2.4.2, we have that there exists a $L > 0$ and for each $\mathbf{z}_1 \in M_i(s, \mathbf{d}_1)$, there exists a $\mathbf{z}_2 \in M_i(s, \mathbf{d}_2)$ such that

$$\|\mathbf{z}_1 - \mathbf{z}_2\| \leq L \left\|(\mathbf{d}_1, \mathbf{b}_G(s)) - (\mathbf{d}_2, \mathbf{b}_G(s))\right\|_1 = L \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1.$$

□

The following corollary establishes a useful topological property of the set D_M as well as the fact that it is non-trivial.

Corollary 6.3.2. *Let Assumption 6.3.3 hold. Assume that $\mathbf{b}_G$ is continuous and that IVP* (6.1) *has a solution for some $\mathbf{u} \in \mathcal{U}$. Then for any choice of matrix $\mathbf{A}$, D_M defined in Eqn.* (6.5) *is nonempty and closed.*

Proof. Let $P : (\mathbf{d}_1, \mathbf{d}_2) \mapsto \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{d}_1, \mathbf{A}_G\mathbf{z} \leq \mathbf{d}_2\}$. Note that $F = \{(\mathbf{d}_1, \mathbf{d}_2) : P(\mathbf{d}_1, \mathbf{d}_2) \neq \varnothing\}$ is closed. This follows from §4.7 of [25]; the argument is that F is the projection of the polyhedron $\{(\mathbf{z}, \mathbf{d}_1, \mathbf{d}_2) : \mathbf{A}\mathbf{z} - \mathbf{d}_1 \leq \mathbf{0}, \mathbf{A}_G\mathbf{z} - \mathbf{d}_2 \leq \mathbf{0}\}$ and thus a polyhedron as well, and so closed.

We note that $P(\mathbf{d}, \mathbf{b}_G(t))$ is exactly $P_M(t, \mathbf{d})$ defined in Eqn. (6.4), and so D_M is the set of $(t, \mathbf{d})$ such that $(\mathbf{d}, \mathbf{b}_G(t)) \in F$. Since IVP (6.1) has a solution for $\mathbf{u} \in \mathcal{U}$, there exists $\mathbf{x}(\cdot, \mathbf{u})$ such that for each $t \in T$, $\mathbf{x}(t, \mathbf{u})$ is in $P(\mathbf{d}, \mathbf{b}_G(t))$, where $\mathbf{d} = \mathbf{A}\mathbf{x}(t, \mathbf{u})$. Therefore F and D_M are nonempty. Finally, note that the mapping $(t, \mathbf{d}) \mapsto (\mathbf{d}, \mathbf{b}_G(t))$ is continuous from $T \times \mathbb{R}^m$ to $\mathbb{R}^{m+m_g}$. Thus, the preimage of F under this mapping must be closed relative to $T \times \mathbb{R}^m$, and it is clear that this preimage must be D_M. And since $T \times \mathbb{R}^m$ is closed, a set which is closed relative to it must be closed relative to $\mathbb{R} \times \mathbb{R}^m$. □

There are other possible choices for the definitions of D_M and the mappings M_i. For instance, the procedure for "tightening" an interval based on a set of linear constraints, given in Definition 4 of [168], provides a potential alternative. However, if this procedure is used, M_i would be interval-valued. In general, this interval would not be degenerate, and this loosely means that the dynamics must be overestimated on a much larger set, leading to more conservative bounds. In contrast, M_i in Proposition 6.3.2 takes the value of the face of a polyhedron (and thus is also polyhedral-valued), and therefore it is always a set in an affine subspace with a lower dimension. Thus, the dynamics are overestimated on a smaller set, leading to tighter bounds.

Further, in applications, the set of possible input values U is typically at least polyhedral (and more often an interval). Consequently, we note that in practice, Hypothesis 4 of Theorem 6.3.1 requires that a linear combination of the dynamics is overestimated on a polyhedron. This observation shapes the following result, which combines Theorem 6.3.1 and Proposition 6.3.2 to obtain a system of differential equations; the solution of this system is $\mathbf{b}$, the right-hand sides of the polyhedral-valued bounds $B : t \mapsto \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}(t)\}$. However, as noted, Hypothesis 4 requires that the dynamics of this system of differential equations overestimate potentially nonlinear optimization problems. Consequently, what is proposed in the following corollary is to solve parametric linear programming relaxations of these problems instead, which overall leads to a much more efficiently solved system. This forms the basis of the numerical method of the next section.

Corollary 6.3.3. *Let Assumptions 6.3.1 and 6.3.3 hold. For $m \in \mathbb{N}$ and $\mathbf{A} \in \mathbb{R}^{m \times n_x}$ define D_M and M_i, $i \in \{1, \dots, m\}$ as in Equations* (6.5) *and* (6.6). *For $i \in \{1, \dots, m\}$, let the mappings $\mathbf{c}_i \equiv (\mathbf{c}_i^u, \mathbf{c}_i^x) : D_M \to \mathbb{R}^{n_u} \times \mathbb{R}^{n_x}$ and $h_i : D_M \to \mathbb{R}$ be given. Assume the following:*

1. *For some $m_u \in \mathbb{N}$, there exist $\mathbf{A}_U \in \mathbb{R}^{m_u \times n_u}$ and $\mathbf{b}_U \in \mathbb{R}^{m_u}$ such that $U = \{\mathbf{p} : \mathbf{A}_U \mathbf{p} \leq \mathbf{b}_U\}$ and is nonempty and compact.*
2. *For $i \in \{1, \dots, m\}$ and all $(t, \mathbf{d}) \in D_M$, $M_i(t, \mathbf{d})$ is compact and a subset of D_x.*
3. *For $i \in \{1, \dots, m\}$, for each $(t, \mathbf{d}) \in D_M$,*

$$\mathbf{a}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z}) \leq (\mathbf{c}_i^u(t, \mathbf{d}))^{\mathrm{T}} \mathbf{p} + (\mathbf{c}_i^x(t, \mathbf{d}))^{\mathrm{T}} \mathbf{z} + h_i(t, \mathbf{d}), \quad \forall (\mathbf{p}, \mathbf{z}) \in U \times M_i(t, \mathbf{d}).$$

4. *The mapping $\mathbf{q} : D_M \to \mathbb{R}^m$ is defined componentwise by*

$$\begin{aligned} q_i(t, \mathbf{d}) = & \max_{(\mathbf{p}, \mathbf{z})} (\mathbf{c}_i^u(t, \mathbf{d}))^{\mathrm{T}} \mathbf{p} + (\mathbf{c}_i^x(t, \mathbf{d}))^{\mathrm{T}} \mathbf{z} + h_i(t, \mathbf{d}) \\ & \text{s.t.} \begin{bmatrix} \mathbf{A}_U & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{A}_G \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{z} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}_U \\ \mathbf{d} \\ \mathbf{b}_G(t) \end{bmatrix}, \\ & \mathbf{a}_i^{\mathrm{T}} \mathbf{z} = \max\{\mathbf{a}_i^{\mathrm{T}} \mathbf{y} : \mathbf{A}\mathbf{y} \leq \mathbf{d}, \mathbf{A}_G \mathbf{y} \leq \mathbf{b}_G(t)\}. \end{aligned} \tag{6.7}$$

5. *The mapping $\mathbf{b} : T \to \mathbb{R}^m$ is any solution of the initial value problem in ordinary differential equations*

$$\dot{\mathbf{b}}(t) = \mathbf{q}(t, \mathbf{b}(t)), \quad a.e.\ t \in T, \tag{6.8}$$

   with initial conditions that satisfy $X_0 \subset \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}(t_0)\}$.

Then for all $\mathbf{u} \in \mathcal{U}$ and any solution $\mathbf{x}(\cdot, \mathbf{u})$ of IVP (6.1), *$\mathbf{A}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{b}(t)$, for all $t \in T$.*

Proof. First, it is clear that the feasible set of the linear program (6.7) which defines $q_i(t, \mathbf{d})$ is the nonempty compact set $U \times M_i(t, \mathbf{d})$. It follows that $q_i(t, \mathbf{d})$ is well defined for each $(t, \mathbf{d}) \in D_M$. Next, by assumption, Assumption 6.3.1 is satisfied. Further, if D_M and M_i are defined as in Equations (6.5) and (6.6), then by Proposition 6.3.2, Assumption 6.3.2 is satisfied. Let $\mathbf{b}$ be any solution of the IVP (6.8). Then it is clear that $\mathbf{b}$ must be absolutely continuous. Let $B : t \mapsto \{\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{b}(t)\}$. Then by the assumption on the initial conditions of $\mathbf{b}$, $X_0 \subset B(t_0)$. Thus, the first two hypotheses of Theorem 6.3.1 are satisfied. Further, if $\mathbf{b}$ is a solution of IVP (6.8), then $(t, \mathbf{b}(t)) \in D_M$ for almost every $t \in T$, since otherwise $\mathbf{q}$

would not be defined and Eqn. (6.8) could not be satisfied almost every $t \in T$. Consequently, by Hypothesis 2, Hypothesis 3 of Theorem 6.3.1 is satisfied. Finally, by assumption on $\mathbf{c}_i$ and h_i and construction of the linear programming relaxation $\mathbf{q}$, for all $(t, \mathbf{d}) \in D_M$

$$q_i(t, \mathbf{d}) \geq \sup\{\mathbf{a}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z}) : (\mathbf{p}, \mathbf{z}) \in U \times M_i(t, \mathbf{d})\}.$$

Therefore, $\mathbf{b}$ must satisfy $\dot{b}_i(t) \geq \mathbf{a}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z})$, for all $(\mathbf{p}, \mathbf{z}) \in U \times M_i(t, \mathbf{b}(t))$, for each $i \in \{1, \ldots, m\}$ and almost every t. Thus, all the assumptions and hypotheses of Theorem 6.3.1 are satisfied and so B must bound all solutions of IVP (6.1). □

6.4 Numerical implementation

The goals of this section are to state an algorithm to compute $\mathbf{q}$ defining IVP (6.8), and in specific, a method to compute the affine relaxations $(\mathbf{c}_i, h_i)$ required in Hypothesis 3 of Corollary 6.3.3. Furthermore, it is established that, with these definitions, $\mathbf{q}$ satisfies an appropriate Lipschitz continuity condition to ensure that IVP (6.8) is amenable to numerical solution.

6.4.1 Computing affine relaxations and the dynamics

To implement a bounding method based on Corollary 6.3.3, affine relaxations are needed. Further, some specific parameterization properties of these relaxations are required. To simplify the discussion, the following assumption is made. With this assumption, we only need to calculate an interval enclosure of $M_i(t, \mathbf{d})$ to establish Hypothesis 3 of Corollary 6.3.3. See Ch. 3 (as well as the discussion in §5.4.2) for a method to construct affine relaxations on intervals which satisfy Assumption 6.4.1.

Assumption 6.4.1. *Let $D_x^{\mathbb{I}} = \{(\mathbf{v}, \mathbf{w}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} : \mathbf{v} \leq \mathbf{w}, [\mathbf{v}, \mathbf{w}] \subset D_x\}$. Assume that for each $i \in \{1, \ldots, m\}$, there exist continuous $\widetilde{\mathbf{c}}_i \equiv (\widetilde{\mathbf{c}}_i^u, \widetilde{\mathbf{c}}_i^x) : T \times D_x^{\mathbb{I}} \to \mathbb{R}^{n_u} \times \mathbb{R}^{n_x}$ and continuous $\widetilde{h}_i : T \times D_x^{\mathbb{I}} \to \mathbb{R}$ such that for each $(\mathbf{v}, \mathbf{w}) \in D_x^{\mathbb{I}}$ and $t \in T$,*

$$\mathbf{a}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z}) \leq (\widetilde{\mathbf{c}}_i^u(t, \mathbf{v}, \mathbf{w}))^{\mathrm{T}} \mathbf{p} + (\widetilde{\mathbf{c}}_i^x(t, \mathbf{v}, \mathbf{w}))^{\mathrm{T}} \mathbf{z} + \widetilde{h}_i(t, \mathbf{v}, \mathbf{w}),$$

for all $(\mathbf{p}, \mathbf{z}) \in U \times [\mathbf{v}, \mathbf{w}]$. Further, for all $(\mathbf{v}, \mathbf{w}) \in D_x^{\mathbb{I}}$, there exists a neighborhood $N^i(\mathbf{v}, \mathbf{w})$

and $\widetilde{L}_i > 0$ such that for all $(t, \mathbf{v}_1, \mathbf{w}_1)$ and $(t, \mathbf{v}_2, \mathbf{w}_2)$ in $T \times N^i(\mathbf{v}, \mathbf{w}) \cap D_x^{\mathbb{I}}$

$$\begin{aligned}\|\widetilde{\mathbf{c}}_i(t, \mathbf{v}_1, \mathbf{w}_1) - \widetilde{\mathbf{c}}_i(t, \mathbf{v}_2, \mathbf{w}_2)\| &\leq \widetilde{L}_i \|(\mathbf{v}_1, \mathbf{w}_1) - (\mathbf{v}_2, \mathbf{w}_2)\|, \\ \left|\widetilde{h}_i(t, \mathbf{v}_1, \mathbf{w}_1) - \widetilde{h}_i(t, \mathbf{v}_2, \mathbf{w}_2)\right| &\leq \widetilde{L}_i \|(\mathbf{v}_1, \mathbf{w}_1) - (\mathbf{v}_2, \mathbf{w}_2)\|.\end{aligned}$$

Next, we introduce a procedure for "tightening" an interval given a set of linear constraints, originally from Definition 4 in [168]. See Algorithm 3; this algorithm defines the operation I_t, which tightens a nonempty interval $[\mathbf{v}, \mathbf{w}]$ by excluding points which cannot satisfy given linear constraints $\mathbf{M}\mathbf{z} \leq \mathbf{d}$. Specifically, the discussion in §5.2 of [168] establishes that the tightened interval $I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M})$ satisfies

$$\{\mathbf{z} \in [\mathbf{v}, \mathbf{w}] : \mathbf{M}\mathbf{z} \leq \mathbf{d}\} \subset I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}) \subset [\mathbf{v}, \mathbf{w}].$$

This property and another regarding parametric regularity are stated formally in Proposition 6.4.1.

Algorithm 3 Definition of the interval-tightening operator I_t

```
Require: $(n_m, n) \in \mathbb{N}^2$, $\mathbf{M} = [m_{i,j}] \in \mathbb{R}^{n_m \times n}$, $\mathbf{d} \in \mathbb{R}^{n_m}$, $(\mathbf{v}, \mathbf{w}) \in \mathbb{R}^n \times \mathbb{R}^n$, $\mathbf{v} \leq \mathbf{w}$
  $(\widehat{\mathbf{v}}, \widehat{\mathbf{w}}) \leftarrow (\mathbf{v}, \mathbf{w})$
  for $i \in \{1, \dots, n_m\}$ do
    for $j \in \{1, \dots, n\}$ do
      if $m_{i,j} \neq 0$ then
        $\gamma \leftarrow \text{median}\left\{\widehat{v}_j, \widehat{w}_j, {}^1\!/\!_{m_{i,j}}\left(d_i + \sum_{k \neq j} \max\{-m_{i,k}\widehat{v}_k, -m_{i,k}\widehat{w}_k\}\right)\right\}$
        if $m_{i,j} > 0$ then
          $\widehat{w}_j \leftarrow \gamma$
        end if
        if $m_{i,j} < 0$ then
          $\widehat{v}_j \leftarrow \gamma$
        end if
      end if
    end for
  end for
  return $I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}) \leftarrow [\widehat{\mathbf{v}}, \widehat{\mathbf{w}}]$
```

Proposition 6.4.1. *For any $(n_m, n) \in \mathbb{N}^2$, let $\mathbf{M} \in \mathbb{R}^{n_m \times n}$. For any $(\mathbf{v}, \mathbf{w}, \mathbf{d}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n_m}$ with $\mathbf{v} \leq \mathbf{w}$, the interval-tightening operator I_t defined in Algorithm 3 satisfies*

$I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}) \neq \varnothing$ and

$$\{\mathbf{z} \in [\mathbf{v}, \mathbf{w}] : \mathbf{M}\mathbf{z} \leq \mathbf{d}\} \subset I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}) \subset [\mathbf{v}, \mathbf{w}].$$

Further, let $\mathbf{v}_{It}$ and $\mathbf{w}_{It}$ be the endpoints of I_t:

$$[\mathbf{v}_{It}(\mathbf{v}, \mathbf{w}, \mathbf{d}; \mathbf{M}), \mathbf{w}_{It}(\mathbf{v}, \mathbf{w}, \mathbf{d}; \mathbf{M})] = I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}).$$

Then there exists a $L_{\mathbf{M}} > 0$ such that for $(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1)$ and $(\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2)$ in $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n_m}$ with $\mathbf{v}_1 \leq \mathbf{w}_1$ and $\mathbf{v}_2 \leq \mathbf{w}_2$,

$$\|\mathbf{v}_{It}(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1; \mathbf{M}) - \mathbf{v}_{It}(\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2; \mathbf{M})\| \leq L_{\mathbf{M}} \|(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1) - (\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2)\|,$$
$$\|\mathbf{w}_{It}(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1; \mathbf{M}) - \mathbf{w}_{It}(\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2; \mathbf{M})\| \leq L_{\mathbf{M}} \|(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1) - (\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2)\|.$$

Proof. First, the fact that I_t is never empty-valued is established. The end value that I_t takes is given by $[\widehat{\mathbf{v}}, \widehat{\mathbf{w}}]$, which is initialized as $[\mathbf{v}, \mathbf{w}]$ (which is nonempty since $\mathbf{v} \leq \mathbf{w}$), and then inductively defined in two nested loops. Consider the i^{th} iteration of the outer loop and the j^{th} iteration of the inner loop. The key thing to note is that the component $\widehat{v}_j$ or $\widehat{w}_j$ is assigned the value of γ, which is defined as the median of $\widehat{v}_j$, $\widehat{w}_j$, and some other value. But since $\widehat{v}_j \leq \widehat{w}_j$, whatever this other value is, we always have $\gamma \in [\widehat{v}_j, \widehat{w}_j]$. Whether $\widehat{v}_j$ or $\widehat{w}_j$ is redefined at this iteration, both $[\widehat{v}_j, \gamma]$ and $[\gamma, \widehat{w}_j]$ are nonempty. Since this holds for each j and i, the end result is a nonempty interval $[\widehat{\mathbf{v}}, \widehat{\mathbf{w}}]$. This also establishes that $I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}) \subset [\mathbf{v}, \mathbf{w}]$. The argument from §5.2 of [168] establishes that $I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}) \supset \{\mathbf{z} \in [\mathbf{v}, \mathbf{w}] : \mathbf{M}\mathbf{z} \leq \mathbf{d}\}$.

To see that the final property holds, note that $(\mathbf{v}_{It}, \mathbf{w}_{It})$ are defined by the finite composition of Lipschitz continuous mappings. These operations are the identity mapping, scalar multiplication, addition, maximum of two numbers, and the median value of three numbers,

$$\text{median}\,\{a, b, c\} = \min\{\max\{a, b\}, \max\{a, c\}, \max\{b, c\}\},$$

and thus it is clear that the operations are Lipschitz continuous. Although there are "if" statements in Algorithm 3, these depend on the matrix $\mathbf{M}$ which is not varying. Consequently, $\mathbf{v}_{It}$ and $\mathbf{w}_{It}$ are Lipschitz continuous, as claimed. □

At this point, a specific algorithm for computing $\mathbf{q}$ defining the dynamics in IVP (6.8) can be stated. See Algorithm 4. In Step 1 of Algorithm 4, an interval enclosure of $M_i(t,\mathbf{d})$ is given by $[\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d})]$, obtained by recursively applying the tightening operation I_t to some interval enclosure of the overall polyhedron $\{\mathbf{z} : \mathbf{A}\mathbf{z} \le \mathbf{d}, \mathbf{A}_G\mathbf{z} \le \mathbf{b}_G(t)\}$. In the algorithm, this initial enclosure is taken as the interval hull. However, this means that none of the constraints defining the overall polyhedron will result in a reduction of the size of the interval when applying the tightening operation. Thus, what should be noted is that the first inequality used in tightening is the one unique to the definition of $M_i(t,\mathbf{d})$: $-\mathbf{a}_i^{\mathrm{T}}\mathbf{z} \le -b_i^*(t,\mathbf{d})$. Intuitively, using this constraint first results in the most significant reduction in the size of the interval.

Algorithm 4 Calculation of dynamics $\mathbf{q}$ of bounding IVP (6.8)

```
Require: $(t,\mathbf{d}) \in D_M$
  Calculate $\mathbf{b}^*(t,\mathbf{d})$ by $b_i^*(t,\mathbf{d}) = \max\{\mathbf{a}_i^{\mathrm{T}}\mathbf{z} : \mathbf{A}\mathbf{z} \le \mathbf{d}, \mathbf{A}_G\mathbf{z} \le \mathbf{b}_G(t)\}$.
  Calculate $[\mathbf{v}^*(t,\mathbf{d}),\mathbf{w}^*(t,\mathbf{d})]$ by

      $v_j^*(t,\mathbf{d}) = \min\{z_j : \mathbf{A}\mathbf{z} \le \mathbf{d}, \mathbf{A}_G\mathbf{z} \le \mathbf{b}_G(t)\}$,
      $w_j^*(t,\mathbf{d}) = \max\{z_j : \mathbf{A}\mathbf{z} \le \mathbf{d}, \mathbf{A}_G\mathbf{z} \le \mathbf{b}_G(t)\}$.

  for $i \in \{1,\dots,m\}$ do
    1. Calculate $[\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d})]$ by the following:
        (a) $[\hat{\mathbf{v}},\hat{\mathbf{w}}] \leftarrow [\mathbf{v}^*(t,\mathbf{d}),\mathbf{w}^*(t,\mathbf{d})]$.
        (b) $[\hat{\mathbf{v}},\hat{\mathbf{w}}] \leftarrow I_t([\hat{\mathbf{v}},\hat{\mathbf{w}}], -b_i^*(t,\mathbf{d}); -\mathbf{a}_i^{\mathrm{T}})$.
        (c) $[\hat{\mathbf{v}},\hat{\mathbf{w}}] \leftarrow I_t([\hat{\mathbf{v}},\hat{\mathbf{w}}], \mathbf{b}^*(t,\mathbf{d}); \mathbf{A})$.
        (d) $[\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d})] \leftarrow I_t([\hat{\mathbf{v}},\hat{\mathbf{w}}], \mathbf{b}_G(t); \mathbf{A}_G)$.
    2. Calculate $\mathbf{c}_i^u(t,\mathbf{d}) = \tilde{\mathbf{c}}_i^u(t,\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d}))$, $\mathbf{c}_i^x(t,\mathbf{d}) = \tilde{\mathbf{c}}_i^x(t,\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d}))$,
       and $h_i(t,\mathbf{d}) = \tilde{h}_i(t,\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d}))$ (See Assumption 6.4.1).
    3. Calculate

       $q_i(t,\mathbf{d}) = \max_{(\mathbf{p},\mathbf{z})} (\mathbf{c}_i^u(t,\mathbf{d}))^{\mathrm{T}}\mathbf{p} + (\mathbf{c}_i^x(t,\mathbf{d}))^{\mathrm{T}}\mathbf{z} + h_i(t,\mathbf{d})$
       $\text{s.t.} \begin{bmatrix} \mathbf{A}_U & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{A}_G \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{z} \end{bmatrix} \le \begin{bmatrix} \mathbf{b}_U \\ \mathbf{d} \\ \mathbf{b}_G(t) \end{bmatrix}$,
       $\mathbf{a}_i^{\mathrm{T}}\mathbf{z} = b_i^*(t,\mathbf{d})$.

  end for
  return $\mathbf{q}(t,\mathbf{d})$
```

6.4.2 Lipschitz continuity of the dynamics

This section establishes that $\mathbf{q}$ defined in Algorithm 4 satisfies a Lipschitz continuity condition akin to that in Definition 2.5.1. As discussed in §2.5, this condition helps establish that an IVP is amenable to solution with most numerical integration methods. It should be noted that the domain of $\mathbf{q}$ defined in Algorithm 4 is D_M. This is not necessarily an open set, which can often cause numerical issues. However, as discussed in §5.4.3, in practice the Lipschitz condition on $\mathbf{q}$ is sufficient for the successful solution of IVP (6.8) with most numerical integration methods.

The following lemma helps establish that the affine relaxations $(\mathbf{c}_i, h_i)$ defined in Algorithm 4 have the appropriate continuity properties required in Theorem 6.4.1 below.

Lemma 6.4.2. *Let Assumptions 6.3.3 and 6.4.1 hold. For $m \in \mathbb{N}$ and $\mathbf{A} \in \mathbb{R}^{m \times n_x}$ define D_M as in Eqn. (6.5). Assume the following:*

1. *$\mathbf{b}_G$ is continuous on T.*
2. *For $i \in \{1, \dots, m\}$, Step 1 in Algorithm 4 defines $(\mathbf{v}^i, \mathbf{w}^i) : D_M \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ which satisfies $[\mathbf{v}^i(t,\mathbf{d}), \mathbf{w}^i(t,\mathbf{d})] \subset D_x$, for all $(t, \mathbf{d}) \in D_M$.*
3. *For $i \in \{1, \dots, m\}$, define $\mathbf{c}_i = (\mathbf{c}_i^u, \mathbf{c}_i^x) : D_M \to \mathbb{R}^{n_u} \times \mathbb{R}^{n_x}$ and $h_i : D_M \to \mathbb{R}$ by Step 2 in Algorithm 4; i.e.*

$$\mathbf{c}_i^u(t,\mathbf{d}) = \widetilde{\mathbf{c}}_i^u(t, \mathbf{v}^i(t,\mathbf{d}), \mathbf{w}^i(t,\mathbf{d})), \quad \mathbf{c}_i^x(t,\mathbf{d}) = \widetilde{\mathbf{c}}_i^x(t, \mathbf{v}^i(t,\mathbf{d}), \mathbf{w}^i(t,\mathbf{d})),$$
$$h_i(t,\mathbf{d}) = \widetilde{h}_i(t, \mathbf{v}^i(t,\mathbf{d}), \mathbf{w}^i(t,\mathbf{d})).$$

Then for $i \in \{1, \dots, m\}$, $\mathbf{c}_i$ and h_i are continuous, and for all $(t, \mathbf{d}) \in D_M$, there exists a neighborhood $N^i(\mathbf{d})$ of $\mathbf{d}$ and $L_i > 0$ such that for all $(t', \mathbf{d}_1)$ and $(t', \mathbf{d}_2)$ in $(T \times N^i(\mathbf{d})) \cap D_M$

$$\left\|\mathbf{c}_i(t', \mathbf{d}_1) - \mathbf{c}_i(t', \mathbf{d}_2)\right\| \le L_i \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|,$$
$$\left|h_i(t', \mathbf{d}_1) - h_i(t', \mathbf{d}_2)\right| \le L_i \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|.$$

Proof. By Lemma 2.4.2 and the fact that $\mathbf{b}_G$ is continuous, $\mathbf{v}^*$, $\mathbf{w}^*$, and $\mathbf{b}^*$ defined in Algorithm 4 are continuous, and there exists a $L > 0$ such that for any $(t, \mathbf{d}_1)$ and $(t, \mathbf{d}_2)$ in D_M

$$\left\|\mathbf{v}^*(t, \mathbf{d}_1) - \mathbf{v}^*(t, \mathbf{d}_2)\right\| \le L \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|,$$

and similarly for $\mathbf{w}^*$ and $\mathbf{b}^*$. Combined with the Lipschitz continuity of the endpoints of I_t from Proposition 6.4.1, $\mathbf{v}^i$ and $\mathbf{w}^i$ are continuous and must satisfy

$$\left\|\mathbf{v}^i(t,\mathbf{d}_1)-\mathbf{v}^i(t,\mathbf{d}_2)\right\|_1 \le L_{v,i}\left\|\mathbf{d}_1-\mathbf{d}_2\right\|, \tag{6.9}$$

$$\left\|\mathbf{w}^i(t,\mathbf{d}_1)-\mathbf{w}^i(t,\mathbf{d}_2)\right\|_1 \le L_{v,i}\left\|\mathbf{d}_1-\mathbf{d}_2\right\|, \tag{6.10}$$

for some $L_{v,i} > 0$ and all $(t,\mathbf{d}_1)$ and $(t,\mathbf{d}_2)$ in D_M.

Choose $(t,\mathbf{d}) \in D_M$. By Hypothesis 2, $(\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d})) \in D_x^{\mathbb{I}}$ (as defined in Assumption 6.4.1). Let $N^i(\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d}))$ be the open neighborhood of $(\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d}))$ assumed to exist by Assumption 6.4.1. Assume without loss of generality that

$$N^i(\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d})) = N_\delta(\mathbf{v}^i(t,\mathbf{d})) \times N_\delta(\mathbf{w}^i(t,\mathbf{d})).$$

By Inequalities (6.9) and (6.10), if $(t,\mathbf{d}') \in D_M$ satisfies that $\|\mathbf{d}-\mathbf{d}'\| < \delta/L_{v,i} = \varepsilon_i$, it follows that $(\mathbf{v}^i(t,\mathbf{d}'),\mathbf{w}^i(t,\mathbf{d}')) \in N^i(\mathbf{v}^i(t,\mathbf{d}),\mathbf{w}^i(t,\mathbf{d}))$. Consequently, for any $(t,\mathbf{d}_1)$ and $(t,\mathbf{d}_2)$ in $(T \times N_{\varepsilon_i}(\mathbf{d})) \cap D_M$,

$$\begin{aligned}\left\|\widetilde{\mathbf{c}}_i(t,\mathbf{v}^i(t,\mathbf{d}_1),\right.&\left.\mathbf{w}^i(t,\mathbf{d}_1)) - \widetilde{\mathbf{c}}_i(t,\mathbf{v}^i(t,\mathbf{d}_2),\mathbf{w}^i(t,\mathbf{d}_2))\right\| \\ &\le \widetilde{L}_i \left\|(\mathbf{v}^i(t,\mathbf{d}_1),\mathbf{w}^i(t,\mathbf{d}_1)) - (\mathbf{v}^i(t,\mathbf{d}_2),\mathbf{w}^i(t,\mathbf{d}_2))\right\|_1 \\ &\le \widetilde{L}_i \left(L_{v,i}\left\|\mathbf{d}_1-\mathbf{d}_2\right\| + L_{v.i}\left\|\mathbf{d}_1-\mathbf{d}_2\right\|\right) \\ &= 2\widetilde{L}_i L_{v,i}\left\|\mathbf{d}_1-\mathbf{d}_2\right\|.\end{aligned}$$

This establishes that $\mathbf{c}_i$ is continuous and satisfies the Lipschitz condition. A similar argument establishes that h_i is continuous and satisfies this condition as well. □

Theorem 6.4.1 below shows that $\mathbf{q}$ defined in Algorithm 4 satisfies the desired Lipschitz continuity assumption of many numerical integration methods, and thus that IVP (6.8) is amenable to numerical solution.

Theorem 6.4.1. *Let Assumptions 6.3.3 and 6.4.1 hold. For $m \in \mathbb{N}$ and $\mathbf{A} \in \mathbb{R}^{m \times n_x}$ define D_M as in Eqn.* (6.5). *In addition, assume the following.*

1. *For some $m_u \in \mathbb{N}$, there exist $\mathbf{A}_U \in \mathbb{R}^{m_u \times n_u}$ and $\mathbf{b}_U \in \mathbb{R}^{m_u}$ such that $U = \{\mathbf{p} : \mathbf{A}_U\mathbf{p} \le \mathbf{b}_U\}$ and is nonempty and compact.*

2. *$\mathbf{b}_G$ is continuous on T.*

3. *IVP* (6.1) *has a solution for some $\mathbf{u} \in \mathcal{U}$.*

4. *For $i \in \{1, \dots, m\}$, Step 1 in Algorithm 4 defines $(\mathbf{v}^i, \mathbf{w}^i) : D_M \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ which satisfies $[\mathbf{v}^i(t, \mathbf{d}), \mathbf{w}^i(t, \mathbf{d})] \subset D_x$, for all $(t, \mathbf{d}) \in D_M$.*

Then the mapping $\mathbf{q}$ defined in Algorithm 4 is continuous, and for all $(t, \mathbf{d}) \in D_M$, there exists a neighborhood $N^q(\mathbf{d})$ of $\mathbf{d}$ and $L_q > 0$ such that for all $(t', \mathbf{d}_1)$ and $(t', \mathbf{d}_2)$ in $(T \times N^q(\mathbf{d})) \cap D_M$

$$\left\|\mathbf{q}(t', \mathbf{d}_1) - \mathbf{q}(t', \mathbf{d}_2)\right\| \leq L_q \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|.$$

Proof. For $i \in \{1, \dots, m\}$, define $b_i^* : D_M \to \mathbb{R}$ by $b_i^*(t, \mathbf{d}) = \max\{\mathbf{a}_i^{\mathrm{T}}\mathbf{y} : \mathbf{A}\mathbf{y} \leq \mathbf{d}, \mathbf{A}_G\mathbf{y} \leq \mathbf{b}_G(t)\}$. It is clear that b_i^* is well defined (i.e. the maximum is indeed attained for any $(t, \mathbf{d}) \in D_M$). Furthermore, since $\mathbf{b}_G$ is continuous and by Lemma 2.4.2 the optimal objective value of an LP is continuous with respect to the right-hand side of its constraints, b_i^* is continuous. Applying Lemma 2.4.2 again, we have that there exists a $L_i^* > 0$ such that

$$\begin{aligned} |b_i^*(t_1, \mathbf{d}_1) - b_i^*(t_2, \mathbf{d}_2)| &\leq L_i^* \left\|(\mathbf{d}_1, \mathbf{b}_G(t_1)) - (\mathbf{d}_2, \mathbf{b}_G(t_2))\right\|_1 \\ &= L_i^* \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1 + L_i^* \left\|\mathbf{b}_G(t_1) - \mathbf{b}_G(t_2)\right\|_1 \end{aligned}$$

for all $(t_1, \mathbf{d}_1)$ and $(t_2, \mathbf{d}_2)$ in D_M. Let $\widehat{m} = m_u + m + m_g + 2$. Let $\widehat{\mathbf{b}}_i : D_M \to \mathbb{R}^{\widehat{m}}$ be given by $\widehat{\mathbf{b}}_i(t, \mathbf{d}) = (\mathbf{b}_U, \mathbf{d}, \mathbf{b}_G(t), b_i^*(t, \mathbf{d}), -b_i^*(t, \mathbf{d}))$. Again, $\widehat{\mathbf{b}}_i$ is the composition of continuous functions and so is continuous. We also have

$$\begin{aligned} \left\|\widehat{\mathbf{b}}_i(t_1, \mathbf{d}_1) - \widehat{\mathbf{b}}_i(t_2, \mathbf{d}_2)\right\|_1 &= \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1 + \left\|\mathbf{b}_G(t_1) - \mathbf{b}_G(t_2)\right\|_1 + 2\left|b_i^*(t_1, \mathbf{d}_1) - b_i^*(t_2, \mathbf{d}_2)\right| \\ &\leq (2L_i^* + 1)(\left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1 + \left\|\mathbf{b}_G(t_1) - \mathbf{b}_G(t_2)\right\|_1), \end{aligned}$$

for all $(t_1, \mathbf{d}_1)$ and $(t_2, \mathbf{d}_2)$ in D_M. From this inequality, there exists a $\widehat{L}_i > 0$ such that

$$\left\|\widehat{\mathbf{b}}_i(t, \mathbf{d}_1) - \widehat{\mathbf{b}}_i(t, \mathbf{d}_2)\right\|_1 \leq \widehat{L}_i \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1$$

for all $(t, \mathbf{d}_1)$ and $(t, \mathbf{d}_2)$ in D_M. Further, since $\mathbf{b}_G$ is continuous on compact T it is also bounded, and for any $(t, \mathbf{d}) \in D_M$ and (bounded) neighborhood $N(\mathbf{d})$ of $\mathbf{d}$, there exists a finite $k \geq 0$ such that

$$\left\| \widehat{\mathbf{b}}_i(t_1, \mathbf{d}_1) - \widehat{\mathbf{b}}_i(t_2, \mathbf{d}_2) \right\| \leq k$$

for all $(t_1, \mathbf{d}_1)$ and $(t_2, \mathbf{d}_2)$ in $(T \times N(\mathbf{d})) \cap D_M$, which is to say that the image of $(T \times N(\mathbf{d})) \cap D_M$ under $\widehat{\mathbf{b}}_i$ is bounded.

For $i \in \{1, \dots, m\}$, define $\mathbf{c}_i = (\mathbf{c}_i^u, \mathbf{c}_i^x) : D_M \to \mathbb{R}^{n_u} \times \mathbb{R}^{n_x}$ and $h_i : D_M \to \mathbb{R}$ as in Step 2 in Algorithm 4 (the same definition in Lemma 6.4.2). Then by Lemma 6.4.2 each $\mathbf{c}_i$ (and h_i) are continuous, and so a similar boundedness condition holds for each $\mathbf{c}_i$: For any $(t, \mathbf{d}) \in D_M$ and bounded neighborhood $N(\mathbf{d})$ of $\mathbf{d}$, $T \times \overline{N(\mathbf{d})}$ is compact. By Corollary 6.3.2, D_M is nonempty and closed, so $(T \times \overline{N(\mathbf{d})}) \cap D_M$ is compact, and so its image under $\mathbf{c}_i$ is compact and thus bounded.

Let

$$\widehat{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_U & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{A}_G \\ \mathbf{0} & \mathbf{a}_i^{\mathrm{T}} \\ \mathbf{0} & -\mathbf{a}_i^{\mathrm{T}} \end{bmatrix}.$$

For $\mathbf{d} \in \mathbb{R}^{\widehat{m}}$, let $P_i(\mathbf{d}) = \{\mathbf{y} : \widehat{\mathbf{A}}_i \mathbf{y} \leq \mathbf{d}\}$. Let $F_i = \{\mathbf{d} : P_i(\mathbf{d}) \neq \varnothing\}$. F_i is a closed set, by a similar argument as in Corollary 6.3.2. Let $\widehat{q}_i : \mathbb{R}^{n_u + n_x} \times F_i \to \mathbb{R}$ be given by $\widehat{q}_i(\mathbf{c}, \mathbf{d}) = \max\{\mathbf{c}^{\mathrm{T}}\mathbf{y} : \widehat{\mathbf{A}}_i \mathbf{y} \leq \mathbf{d}\}$. We note that

$$P_i(\widehat{\mathbf{b}}_i(t, \mathbf{d})) = U \times \arg\max\{\mathbf{a}_i^{\mathrm{T}}\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{d}, \mathbf{A}_G\mathbf{z} \leq \mathbf{b}_G(t)\}.$$

By the definition of D_M and Hypothesis 1, $P_i(\widehat{\mathbf{b}}_i(t, \mathbf{d}))$ is nonempty for each $(t, \mathbf{d}) \in D_M$. By Hypothesis 4 and Proposition 6.4.1,

$$[\mathbf{v}^i(t, \mathbf{d}), \mathbf{w}^i(t, \mathbf{d})] \supset \arg\max\{\mathbf{a}_i^{\mathrm{T}}\mathbf{z} : \mathbf{A}\mathbf{z} \leq \mathbf{d}, \mathbf{A}_G\mathbf{z} \leq \mathbf{b}_G(t)\}$$

and so along with Hypothesis 1, $P_i(\widehat{\mathbf{b}}_i(t, \mathbf{d}))$ is also compact for each $(t, \mathbf{d}) \in D_M$. This also establishes that $\widehat{\mathbf{b}}_i(t, \mathbf{d}) \in F_i$ for each $(t, \mathbf{d}) \in D_M$. Applying Lemma 2.4.2, we note that there exists a $L > 0$ such that for all $(\mathbf{d}_1, \mathbf{d}_2) \in F_i \times F_i$, for each $\mathbf{y}_1 \in P_i(\mathbf{d}_1)$, there exists a

$\mathbf{y}_2 \in P_i(\mathbf{d}_2)$ such that

$$\|\mathbf{y}_1 - \mathbf{y}_2\| \leq L \|\mathbf{d}_1 - \mathbf{d}_2\|.$$

Since $P_i(\widehat{\mathbf{b}}_i(t, \mathbf{d}'))$ is nonempty and compact for all $(t, \mathbf{d}') \in D_M$, there exists a finite $k(t, \mathbf{d}') \geq 0$ such that for all $\mathbf{y}'_1$ and $\mathbf{y}'_2$ in $P_i(\widehat{\mathbf{b}}_i(t, \mathbf{d}'))$,

$$\left\|\mathbf{y}'_1 - \mathbf{y}'_2\right\| \leq k(t, \mathbf{d}').$$

Thus, for any $\mathbf{d} \in F_i$, and for any $\mathbf{y}_1$ and $\mathbf{y}_2$ in $P_i(\mathbf{d})$, we can fix $(t, \mathbf{d}') \in D_M$ and $(\mathbf{y}'_1, \mathbf{y}'_2) \in P_i(\widehat{\mathbf{b}}_i(t, \mathbf{d}')) \times P_i(\widehat{\mathbf{b}}_i(t, \mathbf{d}'))$ such that

$$\begin{aligned}\|\mathbf{y}_1 - \mathbf{y}_2\| &\leq \left\|\mathbf{y}_1 - \mathbf{y}'_1\right\| + \left\|\mathbf{y}'_1 - \mathbf{y}'_2\right\| + \left\|\mathbf{y}'_2 - \mathbf{y}_2\right\| \\ &\leq 2L \left\|\mathbf{d} - \widehat{\mathbf{b}}_i(t, \mathbf{d}')\right\| + k(t, \mathbf{d}') < +\infty.\end{aligned}$$

Consequently, $P_i(\mathbf{d})$ is compact for each $\mathbf{d} \in F_i$. It follows that $\widehat{q}_i$ is finite and well-defined on $\mathbb{R}^{n_u + n_x} \times F_i$, and further by Lemma 2.4.3 it is locally Lipschitz continuous.

Finally, note that $q_i(t, \mathbf{d}) = \widehat{q}_i(\mathbf{c}_i(t, \mathbf{d}), \widehat{\mathbf{b}}_i(t, \mathbf{d})) + h_i(t, \mathbf{d})$. We have that q_i is continuous, since $\widehat{q}_i$, $\mathbf{c}_i$, $\widehat{\mathbf{b}}_i$, and h_i are continuous. Now choose $(t, \mathbf{d}) \in D_M$. By Lemma 6.4.2 there exists a neighborhood $N^i(\mathbf{d})$ of $\mathbf{d}$ and $L_i > 0$ such that for all $(t', \mathbf{d}_1)$ and $(t', \mathbf{d}_2)$ in $(T \times N^i(\mathbf{d})) \cap D_M$

$$\begin{aligned}\left\|\mathbf{c}_i(t', \mathbf{d}_1) - \mathbf{c}_i(t', \mathbf{d}_2)\right\| &\leq L_i \|\mathbf{d}_1 - \mathbf{d}_2\|, \\ \left|h_i(t', \mathbf{d}_1) - h_i(t', \mathbf{d}_2)\right| &\leq L_i \|\mathbf{d}_1 - \mathbf{d}_2\|.\end{aligned}$$

Let K_i be the image of $(T \times N^i(\mathbf{d})) \cap D_M$ under $(\mathbf{c}_i, \widehat{\mathbf{b}}_i)$. As established earlier, $\mathbf{c}_i$ and $\widehat{\mathbf{b}}_i$ are bounded on $(T \times N^i(\mathbf{d})) \cap D_M$, and so it follows that K_i is bounded and so its closure is a compact subset of $\mathbb{R}^{n_u + n_x} \times F_i$. Since $\widehat{q}_i$ is locally Lipschitz continuous (on locally compact $\mathbb{R}^{n_u + n_x} \times F_i$), it is Lipschitz continuous on K_i, and so there exists $\widehat{L}_q > 0$ such that

$$\begin{aligned}\left|q_i(t', \mathbf{d}_1) - q_i(t', \mathbf{d}_2)\right| \leq \widehat{L}_q &\left\|\mathbf{c}_i(t', \mathbf{d}_1) - \mathbf{c}_i(t', \mathbf{d}_2)\right\|_1 + \\ \widehat{L}_q &\left\|\widehat{\mathbf{b}}_i(t', \mathbf{d}_1) - \widehat{\mathbf{b}}_i(t', \mathbf{d}_2)\right\|_1 + \\ &\left|h_i(t', \mathbf{d}_1) - h_i(t', \mathbf{d}_2)\right|\end{aligned}$$

for all $(t', \mathbf{d}_1)$ and $(t', \mathbf{d}_2)$ in $(T \times N^i(\mathbf{d})) \cap D_M$. Then, applying the properties of $\mathbf{c}_i$, $\widehat{\mathbf{b}}_i$ and h_i, we have that

$$\left|q_i(t', \mathbf{d}_1) - q_i(t', \mathbf{d}_2)\right| \leq (\widehat{L}_q L_i + \widehat{L}_q \widehat{L}_i + L_i) \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|,$$

for all $(t', \mathbf{d}_1)$ and $(t', \mathbf{d}_2)$ in $(T \times N^i(\mathbf{d})) \cap D_M$, applying the equivalence of norms on $\mathbb{R}^n$ as necessary. Since this holds for each $i \in \{1, \ldots, m\}$, the desired conclusion holds. □

6.5 Numerical examples

This section considers the performance of a numerical implementation of the bounding method established in Corollary 6.3.3, using the definition of $\mathbf{q}$ in Algorithm 4. This implementation is a C/C++ code which solves the IVP (6.8) with the implementation of the Backwards Differentiation Formulae (BDF) in the CVODE component of the SUNDIALS suite [78] (`http://computation.llnl.gov/casc/sundials/main.html`). Newton's method is used for the corrector iteration. CPLEX version 12.4 [85] is used to solve the linear programs required to define the dynamics in Algorithm 4. Further, all LPs are solved with advanced starting information ("warm-started") with dual simplex. This results in a fairly significant speedup of the code, as Phase I simplex typically can be skipped. The feasibility and optimality tolerances used to solve the LPs and the integration tolerances are given below for each individual example. It should be noted that for these values of the tolerances, an infeasible LP is never encountered in these examples. All numerical studies were performed on a 64-bit Linux virtual machine allocated a single core of a 3.07 GHz Intel Xeon processor and 1.28 GB of RAM.

6.5.1 Lotka-Volterra problem

The Lotka-Volterra problem is a classic problem in the study of nonlinear dynamic systems and often serves as a benchmark for numerical methods. It is thought of as a model for the evolution in time of the populations of a predator and a prey species, and the solution is asymptotically periodic. The equations describing this system are

$$\dot{x}_1(t) = u_1(t)x_1(t)(1 - x_2(t)), \tag{6.11}$$

$$\dot{x}_2(t) = u_2(t)x_2(t)(x_1(t) - 1). \tag{6.12}$$

For this study the initial conditions are $\mathbf{x}(0) = (1.2, 1.1)$. The goal is to compute enclosures of the solutions for any value of the inputs $\mathbf{u} \in \mathcal{U}$, where $U = [2.99, 3.01] \times [0.99, 1.01]$. These are the same input ranges and initial conditions used in [107], which demonstrates the performance of the code VSPODE, an implementation of a Taylor model based bounding procedure.

For this example, one could claim that since x_1 and x_2 represent the populations of species, they should always be nonnegative, and consequently one could set $\mathbf{A}_G = -\mathbf{I}$ and $\mathbf{b}_G : t \mapsto \mathbf{0}$. However, for what will be considered "meaningful" bounds, this kind of *a priori* enclosure does not make a difference. Consequently, for the purpose of applying the theory, the vacuous enclosure given by $\mathbf{A}_G = [0, 0]$ and $\mathbf{b}_G : t \mapsto 0$ is used, although in the implementation this information is unnecessary and is easily omitted.

In [107], VSPODE manages to propagate upper and lower bounds on the solution which remain a subset of the interval in state space $[0.5, 1.5] \times [0.5, 1.5]$ on the time interval $T = [0, 10]$. This is used as a metric to determine whether the calculated bounds are "meaningful." First, interval bounds are calculated, which is to say that the matrix $\mathbf{A} = [-\mathbf{I}\ \ \mathbf{I}]^{\mathrm{T}}$ is used. Unfortunately, despite the use of affine relaxations to improve the estimate of the dynamics, the upper and lower bounds for each species calculated using the above $\mathbf{A}$ matrix cease to be a subset of the interval $[0.5, 1.5] \times [0.5, 1.5]$ before $t = 4$, or before the completion of one full cycle. This, of course, is one of the drawbacks of pure interval enclosures, and what has motivated the development of Taylor model bounding methods. It should be noted that although the bounds are meaninglessly loose in this case, there is no associated numerical "breakdown" before $t = 10$; the solution of the linear programs defining the dynamics and the numerical integration method still proceed without error.

However, using the polyhedral bounding theory discussed in the present work, it is possible to obtain much better upper and lower bounds, which remain a subset of the interval $[0.5, 1.5] \times [0.5, 1.5]$ for all $t \in T$. This can be achieved by letting the i^{th} row $\mathbf{a}_i^{\mathrm{T}}$ of the matrix $\mathbf{A} \in \mathbb{R}^{16\times 2}$ be given by

$$\mathbf{a}_i^{\mathrm{T}} = \left[\cos\left(({}^{i}\!/\!{}_{16})2\pi\right), \quad \sin\left(({}^{i}\!/\!{}_{16})2\pi\right)\right]. \tag{6.13}$$

Each row of $\mathbf{A}$ merely represents the normal of a face of a 16-sided polygon. Lower and upper bounds on each component can be chosen from these bounding hyperplanes. The results are plotted in Fig. 6-1. The upper and lower bounds resulting from $\mathbf{A}$ in Eqn. (6.13) are superior, and indeed are a subset of $[0.5, 1.5] \times [0.5, 1.5]$ for all $t \in [0, 10]$. In effect, the use of the current theory more than doubles the time interval over which meaningful bounds can be calculated.

Using LP feasibility and optimality tolerances of 10^{-5} and 10^{-6}, respectively, and absolute and relative integration tolerances of 10^{-6}, the CPU time required to solve the bounding system is 0.050 seconds (with $\mathbf{A}$ defined in Eqn. (6.13)). For comparison, the purely interval bounds require 0.020 seconds, while the time required by VSPODE (on a processor with a comparable clock speed) reported in [107] is 0.59 seconds. Although the enclosure obtained from VSPODE is tighter, it is solving an intrinsically different problem; namely, one in which the inputs u_1 and u_2 are *constant* functions on T. That is, for all $t \in T$, $\mathbf{u}(t) \equiv \widehat{\mathbf{u}}$ for some $\widehat{\mathbf{u}} \in U$.

6.5.2 Stirred-tank reactor

This next example demonstrates that, in contrast to the previous example, a brute-force approach to constructing polyhedral bounds is not necessary in all cases. For the following class of engineering-relevant problems, an intelligent choice of bounds is available. Furthermore, much of the justification that the choice is intelligent does not have anything to do with the idea that it "wraps" the reachable set in an intelligent manner, which is the typical geometric justification of many bounding methods. Rather, it relates to the idea that the quantities that must be estimated to construct these bounds can be estimated well with tools such as interval arithmetic.

The general form of the material balance equations for a homogeneous, constant-density, stirred-tank reactor with constant material volume is

$$\dot{\mathbf{x}}(t) = \mathbf{S}\mathbf{r}(t, \mathbf{x}(t)) + ({}^{1}\!/\!{}_{V})\left(\mathbf{C}_{in}(t)\mathbf{v}_{in}(t) - v_{out}(t)\mathbf{x}(t)\right), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{6.14}$$

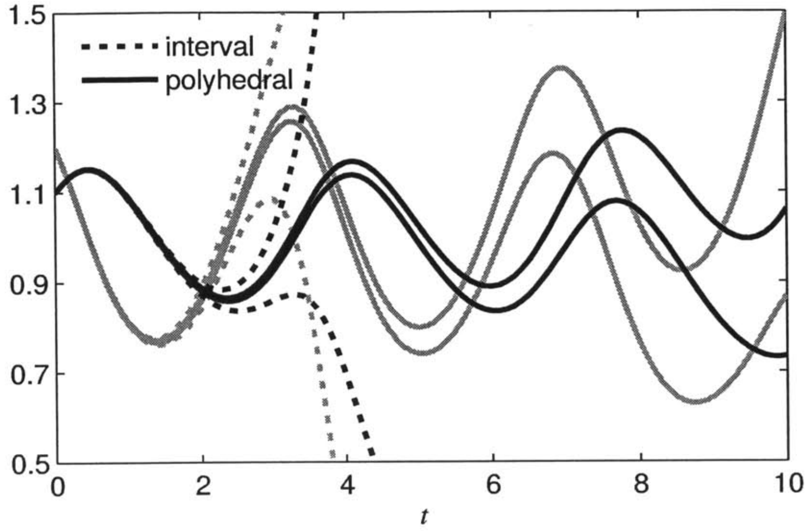


Figure 6-1: Upper and lower bounds on the components of the solution of the Lotka-Volterra problem versus time; x_1 is in gray, while x_2 is in black. Results from polyhedral bounds using $\mathbf{A}$ as in Eqn. (6.13) are solid lines, while results from purely interval bounds are dashed lines.

where $\mathbf{x}(t)$ is the vector of the n_x species concentrations at time t, $\mathbf{S} \in \mathbb{R}^{n_x \times n_r}$ and $\mathbf{r}$ are the stoichiometry matrix and vector of n_r rate functions, respectively, V is the constant reactor volume, $\mathbf{v}_{in}(t) \in \mathbb{R}^p$ is the vector of the volumetric flow rates of the p inlets to the reactor at t, the j^{th} column of $\mathbf{C}_{in}(t) \in \mathbb{R}^{n_x \times p}$ is the vector of species concentrations in the j^{th} inlet, and $\mathbf{1}^{\mathrm{T}}\mathbf{v}_{in}(t) = v_{out}(t)$ is the volumetric flow rate of the single outlet from the reactor.

For a system of this form, a linear transformation yields a system in terms of reaction "variants" and "invariants" [179]. For instance, if $\mathbf{S}$ is full column rank, the rows of $\mathbf{N}$ are left null vectors of $\mathbf{S}$, and $\mathbf{S}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{S}$ (see Ch. 1 of [17]), then letting $\mathbf{y}_1 = \mathbf{S}^+\mathbf{x}$ and $\mathbf{y}_2 = \mathbf{N}\mathbf{x}$ we obtain

$$\begin{aligned} \dot{\mathbf{y}}_1(t) &= \mathbf{r}_y(t, \mathbf{y}(t)) + (1/V)\left(\mathbf{S}^+\mathbf{C}_{in}(t)\mathbf{v}_{in}(t) - v_{out}(t)\mathbf{y}_1(t)\right), \qquad && \mathbf{y}_1(t_0) = \mathbf{S}^+\mathbf{x}_0, \\ \dot{\mathbf{y}}_2(t) &= (1/V)\left(\mathbf{N}\mathbf{C}_{in}(t)\mathbf{v}_{in}(t) - v_{out}(t)\mathbf{y}_2(t)\right), && \mathbf{y}_2(t_0) = \mathbf{N}\mathbf{x}_0, \end{aligned} \tag{6.15}$$

where the subscript y on $\mathbf{r}$ denotes that $\mathbf{r}_y$ is considered a function of the transformed variables. If the system has no inlets or outlets, i.e. is a batch reactor, then $\dot{\mathbf{y}}_2(t) = \mathbf{0}$ for all t, and so is constant. From the perspective of the original system, the solution must obey the affine constraints $\mathbf{N}\mathbf{x}(t) = \mathbf{N}\mathbf{x}_0$ for all t. This forms the basis of the *a priori* enclosures used in [74, 167, 168].

However, the addition of inlets and outlets complicates this kind of *a priori* enclosure. It is possible to salvage this enclosure, by noting that the linear transformation partially decouples the system of equations. If $\mathbf{C}_{in}$ and $\mathbf{v}_{in}$ are known, simple functions or are constant parameters, then an analytical solution for $\mathbf{y}_2$ can be obtained fairly easily. The result is that the *a priori* enclosure is now time-varying; specifically, the solution must obey $\mathbf{N}\mathbf{x}(t) = \mathbf{y}_2(t)$, where again $\mathbf{y}_2$ is now known explicitly. This kind of information still satisfies Assumption 6.3.3, and could be used in the current bounding theory. But again, matters are more complicated if, for instance, the values of $\mathbf{C}_{in}$ or $\mathbf{v}_{in}$ are subject to some unknown, but bounded time-varying disturbance.

Instead, the approach taken here will be to use a bounding polyhedron that will implicitly enforce this time-varying enclosure, without the need to determine explicitly $\mathbf{y}_2$, or some other functions that take the role of $\mathbf{b}_G$. Since the solution $\mathbf{x}$ describes concentrations, the components must be nonnegative, and so for this example the only *a priori* enclosure used is given by these nonnegativity constraints. In general, the bounding matrix is given by

$$\mathbf{A} = \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \\ -\mathbf{D}^+ \\ \mathbf{D}^+ \\ -\mathbf{N} \\ \mathbf{N} \end{bmatrix}, \tag{6.16}$$

where $\mathbf{D}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{D}$, a matrix formed from a maximal set of linearly independent columns of $\mathbf{S}$, and the rows of $\mathbf{N}$ are linearly independent and span the left null space of $\mathbf{S}$.

Before considering the specifics of the example, the merits of this form of polyhedral bounds, as determined by the matrix $\mathbf{A}$ above, are discussed. It helps to write the dynamics of the system (6.14) in the following general form:

$$\mathbf{f}(t, \mathbf{p}, \mathbf{z}) = \begin{bmatrix} \mathbf{S} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{r}_p(t, \mathbf{p}, \mathbf{z}) \\ \mathbf{g}_1(t, \mathbf{p}) - g_2(t, \mathbf{p})\mathbf{z} \end{bmatrix} = \widehat{\mathbf{S}}\widehat{\mathbf{r}}(t, \mathbf{p}, \mathbf{z}),$$

where the functions $\mathbf{g}_1$ and g_2 take into account the possibility that the inlet flow rates and concentrations are modeled as controls, parameters, or disturbances, and $\mathbf{r}_p$ takes into account that the reaction kinetics might not be known exactly. In general, the more rows that the matrix $\mathbf{A}$ has, the tighter the bounds. Of course, too many superfluous rows slows down the calculation and does little to improve the bounds. To a certain extent, the bounds on the individual components (that is, the interval bounds) are improved the most when linear combinations of the states that are "estimated well" are included in the bounds. Inspired by the previous discussion, the linear combinations $\mathbf{y}_2 = \mathbf{N}\mathbf{z}$, which no longer depend on the reaction rates, are a good candidate. Part of this relates to the specifics of how the affine overestimators of the dynamics are constructed. The affine relaxation method described in Ch. 3 requires interval arithmetic, and it is well known that the effectiveness of interval arithmetic is diminished by the dependency problem (see for instance §1.4 in [135]). As a general observation, the "simpler" the expression, the more effective interval arithmetic is at generating a tight estimate of its range. Thus, the quantities $\mathbf{y}_2$, whose dynamics no longer depend on the potentially nonlinear rate function $\mathbf{r}_p$, have a good chance of being estimated well by interval arithmetic and the affine relaxation method. Further, the dynamics for $\mathbf{y}_2$ are decoupled. This is significant since the value of $\mathbf{a}_i^{\mathrm{T}}\mathbf{z}$ is unique for $\mathbf{z} \in M_i(t, \mathbf{d})$, and it is over $M_i(t, \mathbf{d})$ which the dynamics must be estimated. This means that if $\mathbf{a}_i$ is the j^{th} row of $\mathbf{N}$, then $\mathbf{a}_i^{\mathrm{T}}\mathbf{z} = y_{2,j}$ and $\mathbf{a}_i^{\mathrm{T}}\mathbf{f}(t, \mathbf{p}, \mathbf{z}) = \mathbf{a}_i^{\mathrm{T}}\mathbf{g}_1(t, \mathbf{p}) - g_2(t, \mathbf{p})y_{2,j}$. In a loose sense, uncertainty with respect to the states has been removed, and overestimating the dynamics in this case only requires overestimation with respect to the inputs.

Similar reasoning supports why the quantities $\mathbf{y}_1 = \mathbf{D}^{+}\mathbf{z}$ also are estimated well. If $\mathbf{S}$ is full column rank, one can choose $\mathbf{D} = \mathbf{S}$ and then $\mathbf{D}^{+} = \mathbf{S}^{+}$, and so $\mathbf{D}^{+}\mathbf{S} = \mathbf{I}$. The result is that the dynamics of these quantities $\mathbf{y}_1$ only depend on a single component of the rate function $\mathbf{r}_p$ (in fact, in a batch system, this motivates their interpretation as "extents of reaction"). As before, the simpler the expression, the more likely it is to be estimated well via the affine relaxation procedure. If $\mathbf{S}$ is not full column rank, for instance $\mathbf{S} = [\mathbf{D} \;\; \mathbf{E}]$, then $\mathbf{D}^{+}\mathbf{S} = [\mathbf{I} \;\; \mathbf{D}^{+}\mathbf{E}]$, and again, the expression for the dynamics of the quantities $\mathbf{y}_1$ is potentially simplified.

At this point it is reasonable to wonder why not apply an interval-based bounding method to the transformed system (6.15). The complicating fact is that this requires explicitly rewriting the rate function in terms of the transformed variables to obtain $\mathbf{r}_y$. In general, this is not a trivial task. Although it is possible to automate the evaluation of $\mathbf{r}_y$, since the transformation from $\mathbf{x}$ to $\mathbf{y}$ is invertible in certain cases [179], it is likely that extending this evaluation to interval arithmetic will suffer from dependency issues. For this reason, explicitly bounding the original variables, in terms of which the rate function is originally written, helps reduce the overestimation of the original variables, and in turn reduce the overestimation of the range of the rate function on the various sets it must be estimated. Finally, as upper and lower bounds on the original variables are most likely the ultimate goal of estimating the reachable set of this system, there is little reason to exclude bounding them in the definition of $\mathbf{A}$.

Now, consider the specifics of the example. Let the components of the solution be $\mathbf{x} = (x_{\mathrm{A}}, x_{\mathrm{B}}, x_{\mathrm{C}}, x_{\mathrm{D}})$, which are the concentration profiles (in M) of the four chemical species A, B, C, and D, respectively. Let

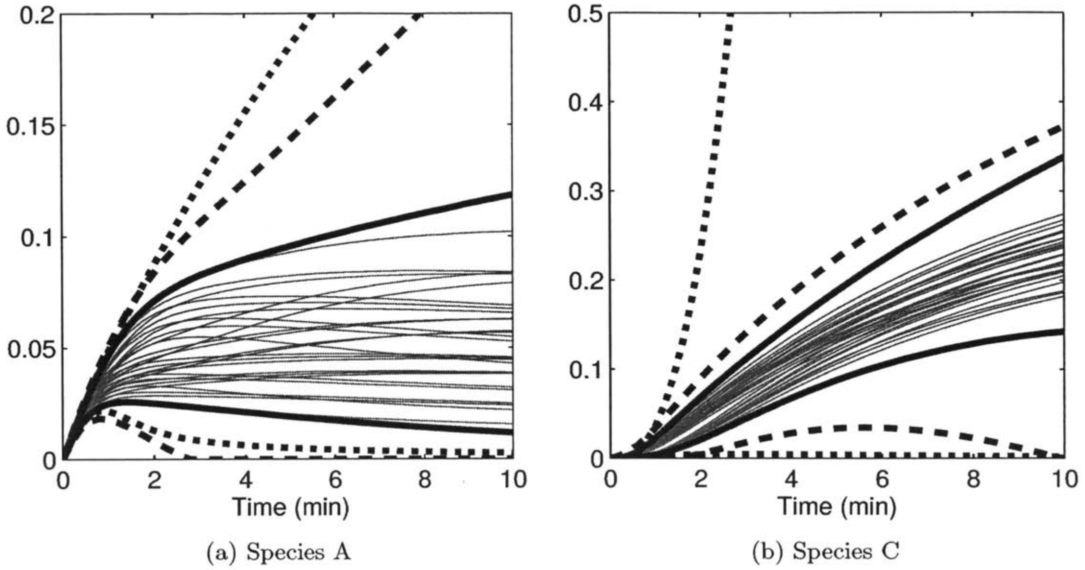
$$\dot{\mathbf{x}}(t,\mathbf{u}) = \begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_3(t)x_{\mathrm{A}}(t,\mathbf{u})x_{\mathrm{B}}(t,\mathbf{u}) \\ k_2 x_{\mathrm{A}}(t,\mathbf{u})x_{\mathrm{C}}(t,\mathbf{u}) \\ (1/V)(u_1(t)v_{\mathrm{A}} - x_{\mathrm{A}}(t,\mathbf{u})(v_{\mathrm{A}}+v_{\mathrm{B}})) \\ (1/V)(u_2(t)v_{\mathrm{B}} - x_{\mathrm{B}}(t,\mathbf{u})(v_{\mathrm{A}}+v_{\mathrm{B}})) \\ (1/V)(-x_{\mathrm{C}}(t,\mathbf{u})(v_{\mathrm{A}}+v_{\mathrm{B}})) \\ (1/V)(-x_{\mathrm{D}}(t,\mathbf{u})(v_{\mathrm{A}}+v_{\mathrm{B}})) \end{bmatrix}. \tag{6.17}$$

The known parameters are $V = 20$ (L), $k_2 = 0.4$ (M^{-1}min^{-1}), $v_{\mathrm{A}} = v_{\mathrm{B}} = 1$ (L(min)$^{-1}$). The time-varying uncertainties are the inlet concentration of species A, $u_1(t) \in [0.9, 1.1]$ (M), the inlet concentration of species B, $u_2(t) \in [0.8, 1.0]$ (M), and the rate constant of the first reaction, $u_3(t) \in [10, 50]$ (M^{-1}min^{-1}). Initially, the concentration of each species is zero, and at $t = 0$, A and B begin to flow in. The time period of interest is $T = [0, 10]$ (min). The first two columns of the matrix in Eqn. (6.17) correspond to the stoichiometry matrix $\mathbf{S}$. It columns are linearly independent, and so let

$$\mathbf{D}^{+} = \mathbf{S}^{+} = \begin{bmatrix} -1/3 & -1/3 & 1/3 & 0 \\ -1/3 & 0 & -1/3 & 1/3 \end{bmatrix} \text{ and } \mathbf{N} = \begin{bmatrix} -1 & 2 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix}.$$

Results for two representative species are in Fig. 6-2. For comparison, the interval hull of the enclosures that result from using

$$\mathbf{A}' = \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} \text{ or } \mathbf{A}'' = \begin{bmatrix} -\mathbf{D}^{+} \\ \mathbf{D}^{+} \\ -\mathbf{N} \\ \mathbf{N} \end{bmatrix} \tag{6.18}$$



(a) Species A

(b) Species C

Figure 6-2: Interval hull of enclosures versus time for the stirred-tank reactor (Eqn. (6.17)). Solution trajectories for various constant inputs are thin solid lines. Results from **A** in Eqn. (6.16) are solid black lines, while results from $\mathbf{A}'$ and $\mathbf{A}''$ as in Eqn. (6.18) are dotted lines and dashed lines, respectively.

as the matrix that defines the polyhedral enclosure are included. These are interval bounds on the original system (6.14), and (roughly) the transformed system (6.15), respectively. The interval hull of the polyhedral enclosures are calculated in a post-processing step for the purpose of comparing the different results on an equal footing. It is clear that the bounds that result from using **A** in Eqn. (6.16) are superior, and much tighter than just the intersection of the bounds resulting from the other enclosures. Finally, with LP feasibility and optimality tolerances of 10^{-5} and 10^{-6}, respectively, and absolute and relative integration tolerances of 10^{-6}, the CPU time required to solve the bounding system is 0.030 seconds.

6.5.3 Piecewise affine relaxations

An interesting application of this theory is to the construction of piecewise affine relaxations of the solutions of initial value problems in parametric ordinary differential equations. The initial value problem in parametric ordinary differential equations is a special case of the problem of interest (6.1), when the uncertainty is fixed, i.e. not time-varying. Certainly, the theory as it stands can handle this case already, but intuitively the bounds produced may not be as tight as those produced by a method that explicitly takes advantage of the fact that the uncertain inputs have a constant value in time, such as the Taylor-model methods described in [107].

The idea is fairly straightforward; the fixed uncertain parameters are treated as extra state variables with zero time derivatives and an uncertain set of initial values. Of course, nothing is gained from this reformulation if one can only propagate interval bounds on the states, but if one propagates polyhedral bounds the reformulation is meaningful.

To demonstrate this, an example adapted from Example 2 in [168] is considered, involving the following enzymatic reaction network:

$$\begin{aligned} \mathrm{A} + \mathrm{F} &\rightleftharpoons \mathrm{F{:}A} \rightarrow \mathrm{F} + \mathrm{A}', \\ \mathrm{A}' + \mathrm{R} &\rightleftharpoons \mathrm{R{:}A'} \rightarrow \mathrm{R} + \mathrm{A}. \end{aligned}$$

The dynamic equations governing the evolution of the species concentrations

$$\mathbf{x} = (x_{\mathrm{A}}, x_{\mathrm{F}}, x_{\mathrm{F:A}}, x_{\mathrm{A'}}, x_{\mathrm{R}}, x_{\mathrm{R:A'}})$$

in a closed system are

$$\begin{aligned} \dot{x}_{\mathrm{A}} &= -k_1 x_{\mathrm{F}} x_{\mathrm{A}} + k_2 x_{\mathrm{F:A}} + k_6 x_{\mathrm{R:A'}}, \\ \dot{x}_{\mathrm{F}} &= -k_1 x_{\mathrm{F}} x_{\mathrm{A}} + k_2 x_{\mathrm{F:A}} + k_3 x_{\mathrm{F:A}}, \\ \dot{x}_{\mathrm{F:A}} &= k_1 x_{\mathrm{F}} x_{\mathrm{A}} - k_2 x_{\mathrm{F:A}} - k_3 x_{\mathrm{F:A}}, \\ \dot{x}_{\mathrm{A'}} &= k_3 x_{\mathrm{F:A}} - k_4 x_{\mathrm{A'}} x_{\mathrm{R}} + k_5 x_{\mathrm{R:A'}}, \\ \dot{x}_{\mathrm{R}} &= -k_4 x_{\mathrm{A'}} x_{\mathrm{R}} + k_5 x_{\mathrm{R:A'}} + k_6 x_{\mathrm{R:A'}}, \\ \dot{x}_{\mathrm{R:A'}} &= k_4 x_{\mathrm{A'}} x_{\mathrm{R}} - k_5 x_{\mathrm{R:A'}} - k_6 x_{\mathrm{R:A'}}. \end{aligned} \tag{6.19}$$

The time interval of interest is $T = [0, 0.04]$ (s). For the original states $\mathbf{x}$, the initial conditions are $\mathbf{x}_0 = (34, 20, 0, 0, 16, 0)$ (M). Let the uncertain, but constant, parameters be $(p_1, p_2) = (k_1, k_6) \in [0.1, 1] \times [0.3, 3] = P$. The other k_i are known: $(k_2, k_3, k_4, k_5) = (0.1815, 88, 27.5, 2.75)$. To obtain the reformulated system, append the equations $\dot{\mathbf{p}} = \mathbf{0}$ to Equations (6.19) and now for the reformulated system, the initial conditions are uncertain: $(\mathbf{x}(0), \mathbf{p}(0)) \in \{\mathbf{x}_0\} \times P$. As in [168], the following *a priori* enclosure is available for the reformulated system:

$$G \equiv \{(\mathbf{z},\mathbf{r}) \in \mathbb{R}^6 \times \mathbb{R}^2 : \mathbf{0} \leq \mathbf{z} \leq \bar{\mathbf{x}}, \mathbf{N}\mathbf{z} = \mathbf{N}\mathbf{x}_0, \mathbf{r} \in P\}, \quad \text{with}$$

$$\mathbf{N} = \begin{bmatrix} 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 1 & -1 & 0 \end{bmatrix},$$

$$\bar{\mathbf{x}} = (34, 20, 20, 34, 16, 16).$$

The bounding matrix $\mathbf{A}$ used in this example is

$$\mathbf{A} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{M} \\ \mathbf{I} & -\mathbf{M} \end{bmatrix}$$

where $\mathbf{M}$ is the matrix whose columns are (approximately) the sensitivities of $\mathbf{x}$ with respect to each p_i, evaluated at the final time $t_f = 0.04$ and at the midpoint of the interval P. These sensitivities, as calculated numerically with CVODES, are

$$\begin{bmatrix} -14.4 & -3.12 & 3.12 & 1.99 & -9.28 & 9.28 \\ 0.105 & -0.00577 & 0.00577 & -0.00748 & 0.103 & -0.103 \end{bmatrix}^{\mathrm{T}},$$

however, any value with magnitude less than 10^{-2} is set to zero to construct $\mathbf{M}$. The reasoning behind this form of $\mathbf{A}$ is that the first half of its rows give interval bounds, while the second half of its rows give affine under and overestimators of each original state with respect to $\mathbf{p}$, and specifically, these should be "good" estimators at the final time point t_f.

Fig. 6-3 shows the piecewise affine underestimator (maximum of the lower bound and affine underestimator) and overestimator (minimum of the upper bound and affine overestimator) for a certain concentration on the set P. When only interval bounds are propagated, the interval bound on the state $x_{\mathrm{F:A}}$ at t_f is $[0.517, 4.79]$, while the use of the affine relaxations reduces this to $[0.582, 4.09]$, corresponding to an 18% reduction in the width of the enclosure. Thus, using the extra bounds in the form of affine under and overestimators also improves the interval bounds. This contrasts with the methods in [169] and [176], where the

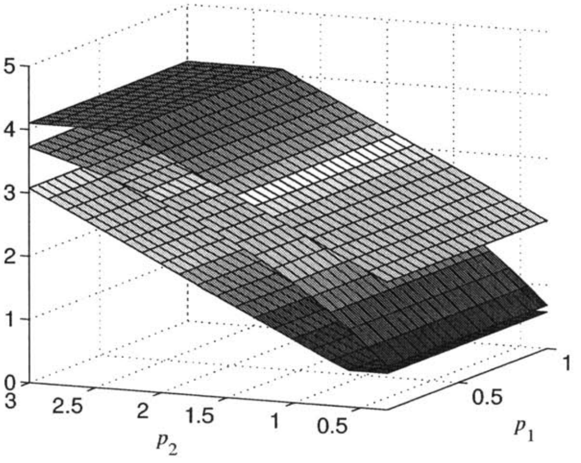


Figure 6-3: Piecewise affine under and overestimators of $x_{\mathrm{F:A}}$ at t_f on P; the sampled parametric solution surface is in the middle.

benefit is one-way; relaxations with respect to the parameters cannot improve the interval bounds.

With LP feasibility and optimality tolerances of 10^{-5} and 10^{-6}, respectively, and absolute and relative integration tolerances of 10^{-6}, the CPU time required to solve the bounding system is 0.15 seconds. In comparison, the methods for constructing convex and concave relaxations of the solutions of parametric ordinary differential equations presented in [169, 170] also involve the solution of an auxiliary dynamic system, but this system must be solved at each parameter value of interest to determine the value of the relaxations. The current method only requires that the auxiliary dynamic system is solved once to obtain the value of the relaxation on the entire parameter range.

6.6 Conclusions

This work has presented a general theory, as well as an efficient numerical implementation, for the construction of polyhedral bounds on the reachable set of a dynamic system subject to time-varying inputs and uncertain initial conditions. Some more fine-tuning of the current numerical implementation of the bounding method is a subject for future research. For instance, the function $\mathbf{q}$ defining the dynamics of the bounding system is nonsmooth, and it may be beneficial to supply approximate or "locked" Jacobian information to the numerical integrator. As mentioned in §6.4.2, to ensure that one does not run into domain issues, the reformulation in [60] could be used. Alternative implementations of the theory are also a subject for future research; that is, defining the general mappings M_i differently provides avenues for different numerical implementations. These different numerical implementations might avoid some of the cost of the solution of the linear programs, and provide a slightly faster method, although potentially at the cost of producing more conservative bounds. Nevertheless, the current work as is stands as an effective method.

Chapter 7

Polyhedral bounds for nonlinear control systems with constraints

7.1 Introduction

This chapter considers the theoretical and numerical aspects of the construction of enclosures (or "bounds") of the reachable set of nonlinear control systems. First, a general theory is proven which gives sufficient conditions for a time-varying polyhedron to enclose all solutions of a constrained dynamic system subject to uncertain inputs and initial conditions (see §7.2.1 for an exact statement). This theory is in the vein of a comparison theorem involving differential inequalities. Such theorems have a long history, going back to "Müller's theorem" [203], which was subsequently generalized to control systems in [72]. These theorems give conditions under which one can construct componentwise upper and lower, or interval, bounds on the solutions. Recent work in [168, 202] has expanded these theorems. This chapter continues those developments; in fact, it will be shown that the theories in [168] and Ch. 6 are special cases of the theory developed here.

One of the main contributions of this work is the extension of these differential inequality-based theorems to include dynamic systems with constraints. Although the theoretical developments of [168] and Ch. 6 are very similar to those of this work, neither of these previous theorems as stated can handle constraints on the states. General reachability analysis with constraints has been addressed in [104], where the focus is on linear systems and ellipsoidal enclosures of the reachable sets. Related work deals with control problems with state constraints [31, 102], in which the theoretical basis of the approaches is on the Hamilton-Jacobi-Bellman partial differential equation (PDE); the authors of [102] note that the solution of such a PDE is in general complicated for nonlinear systems. Meanwhile, [88, 99, 98] deal with interval bounds on the reachable set in the context of parameter and state estimation problems.

This work's treatment of constraints is independent of any particular setting and as a result a number of interesting connections to other problems and topics arise. For instance, in the context of dynamic optimization problems, such constraints are called path constraints [214]; these constraints can be used in the reachable set estimation problem to tighten the enclosure. Thus, if the enclosure is used to construct a relaxation of the dynamic optimization problem, the result is a tighter relaxation. A more exact discussion of this is in §7.2.2. Constraints are also discussed in the context of differential-algebraic equations (DAEs) in §7.5.4 and continuous-time measurements in state estimation problems in §7.7.1.

Another contribution is that this work provides a new method for the construction of relaxations of the solutions of initial value problems (IVPs) in parametric ordinary differential equations (ODEs). Previous work dealing with this includes [169, 176, 202]. The relaxation theory developed here is inspired the most by [202]. However, that work focuses on the case of IVPs in parametric ODEs, while the basic theorem in this work deals with control systems (time-varying inputs), and derives relaxations for the solutions of parametric ODEs as a special case. Neither theory is more general than the other; however, some interesting overlap is discussed, and the two theories provide different approaches to similar problems.

Numerical methods for constructing polyhedral bounds and relaxations are discussed, and the performances of these methods are assessed with examples. At the heart of these methods is the construction of an auxiliary system of ODEs, whose solution yields the parameters describing the polyhedral enclosure. As a result, the proposed methods benefit from the ability to use powerful methods for numerical integration.

There are many other connections to previous work; these are discussed throughout this chapter as the connections become apparent. The rest of this chapter is organized as follows. Section 7.2 provides the formal problem statement. A brief discussion of path constraints in dynamic optimization in §7.2.2 motivates this work's consideration of constrained dynamic systems. Section 7.3 provides the core theoretical developments. Subsequently, §7.4 discusses specific instances of this theory when constraints are absent. This yields theories for interval bounds (§7.4.1), polyhedral bounds (with time-varying normals for each face, §7.4.2), and affine relaxations (§7.4.3). Next, §7.5 specializes the theory to construct polyhedral bounds (Sections 7.5.1 and 7.5.2) and affine relaxations (§7.5.3) for constrained systems. As mentioned, there is also a discussion of the connections to DAEs (§7.5.4). Section 7.6 provides more specific information on the numerical implementation of two of the theories for constrained systems from the previous section. Section 7.7 then looks at the efficiency of these methods and tightness of the resulting bounds. One of these examples shows that the affine relaxation method developed has empirical convergence order of two. Finally, §7.8 concludes with some final thoughts.

7.2 Problem statement

7.2.1 Problem statement

Let $(n_x, n_u) \in \mathbb{N}^2$, nonempty interval $T = [t_0, t_f] \subset \mathbb{R}$, $D_x \subset \mathbb{R}^{n_x}$, and $D_u \subset \mathbb{R}^{n_u}$ be given. For $U : T \rightrightarrows D_u$, let the set of time-varying inputs be

$$\mathcal{U} = \left\{ \mathbf{u} \in L^1(T, \mathbb{R}^{n_u}) : \mathbf{u}(t) \in U(t), a.e.\ t \in T \right\},$$

and let the set of possible initial conditions be $X_0 \subset D_x$. Let the state constraints be given by $X_C : T \rightrightarrows \mathbb{R}^{n_x}$. Given $\mathbf{f} : T \times D_u \times D_x \to \mathbb{R}^{n_x}$, the problem of interest is the constrained initial value problem in ODEs

$$\dot{\mathbf{x}}(t, \mathbf{u}) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t, \mathbf{u})), \quad a.e.\ t \in T, \tag{7.1a}$$

$$\mathbf{x}(t_0, \mathbf{u}) \in X_0, \tag{7.1b}$$

$$\mathbf{x}(t, \mathbf{u}) \in X_C(t), \quad \forall t \in T. \tag{7.1c}$$

For given $\mathbf{u} \in \mathcal{U}$, a *solution of IVP* (7.1) is an absolutely continuous mapping $\mathbf{x}(\cdot, \mathbf{u}) : T \to D_x$ which satisfies Conditions (7.1). The goal of this work is to construct a polyhedral-valued mapping $B : T \rightrightarrows \mathbb{R}^{n_x}$ such that for all $\mathbf{u} \in \mathcal{U}$ and any solution $\mathbf{x}(\cdot, \mathbf{u})$ (if one exists for this $\mathbf{u}$), $\mathbf{x}(t, \mathbf{u}) \in B(t)$, for all $t \in T$.

Also of interest is a solution of the *unconstrained IVP* (7.1), which for given $\mathbf{u} \in \mathcal{U}$, is an absolutely continuous mapping $\mathbf{x}(\cdot, \mathbf{u}) : T \to D_x$ which satisfies Conditions (7.1a) and (7.1b), but not necessarily (7.1c). A solution of the unconstrained IVP will also be called an unconstrained solution of IVP (7.1). With this terminology, all solutions are also unconstrained solutions, but not the other way around.

7.2.2 Dynamic optimization

A more precise motivation for the current problem can now be described. Consider the simple dynamic optimization problem over the time period $T = [t_0, t_f]$

$$\begin{aligned} q^* = \inf_{\mathbf{u}} \; & x_1(t_f, \mathbf{u}) \\ \text{s.t. } & \mathbf{u} \in \mathcal{U}, \\ & \mathbf{g}(\mathbf{x}(t,\mathbf{u})) \leq \mathbf{0}, \quad \forall t \in T, \\ & \dot{\mathbf{x}}(t,\mathbf{u}) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t,\mathbf{u})), \quad a.e.\ t \in T, \\ & \mathbf{x}(t_0, \mathbf{u}) \in X_0, \end{aligned} \tag{7.2}$$

for appropriate mappings $\mathbf{g}$, $\mathbf{f}$, and sets X_0 and $\mathcal{U}$. The path constraints in this problem are $\mathbf{g}(\mathbf{x}(t,\mathbf{u})) \leq \mathbf{0}$, for all $t \in T$. If we calculate an interval enclosure of the solutions of the (unconstrained) initial value problem embedded in the constraints of the optimization problem (7.2), i.e. $(\mathbf{x}^L, \mathbf{x}^U)$ such that $\mathbf{x}(t,\mathbf{u}) \in [\mathbf{x}^L(t), \mathbf{x}^U(t)]$ for all $t \in T$ and $\mathbf{u} \in \mathcal{U}$, then $x_1^L(t_f)$ is a lower bound for the optimal solution value. Obtaining such a lower bound is an important part of solving dynamic optimization problems to global optimality with a deterministic method such as branch and bound [166, 175].

However, the path constraint information can be used to improve the tightness of the lower bound. Instead, assume we calculate $(\tilde{\mathbf{x}}^L, \tilde{\mathbf{x}}^U)$ such that $\mathbf{x}(t,\mathbf{u}) \in [\tilde{\mathbf{x}}^L(t), \tilde{\mathbf{x}}^U(t)]$ for all $t \in T$ and $\mathbf{u} \in \mathcal{U}$ such that $\mathbf{x}(\cdot,\mathbf{u})$ satisfies $\dot{\mathbf{x}}(t,\mathbf{u}) = \mathbf{f}(t,\mathbf{u}(t),\mathbf{x}(t,\mathbf{u}))$, almost every $t \in T$, *and* $\mathbf{g}(\mathbf{x}(t,\mathbf{u})) \leq \mathbf{0}$ for all $t \in T$. In other words, $\mathbf{x}(\cdot,\mathbf{u})$ is in effect the solution of a constrained initial value problem. In this case, we still have $\tilde{x}_1^L(t_f) \leq q^*$. In addition, a bounding method which uses the constraints effectively should give $x_1^L(t_f) \leq \tilde{x}_1^L(t_f)$. That is, a tighter lower bound on the optimal objective value of the dynamic optimization problem is obtained.

7.3 Bounding theory

Lemma 7.3.1. *Let $T \subset \mathbb{R}$ be an interval, and let $b : T \to \mathbb{R}$, $\mathbf{a} : T \to \mathbb{R}^n$, and $\mathbf{x} : T \to \mathbb{R}^n$ be absolutely continuous mappings. Then, the real-valued function $g : t \mapsto \max\{0, \mathbf{a}(t)^{\mathrm{T}}\mathbf{x}(t) - b(t)\}$ is absolutely continuous. Further, for almost all t such that $\mathbf{a}(t)^{\mathrm{T}}\mathbf{x}(t) > b(t)$ and for any $(\mathbf{v}, \mathbf{z}) \in \mathbb{R}^n \times \mathbb{R}^n$ such that $\mathbf{a}(t)^{\mathrm{T}}\mathbf{v} + \dot{\mathbf{a}}(t)^{\mathrm{T}}\mathbf{z} \le \dot{b}(t)$,*

$$\dot{g}(t) \le \|\dot{\mathbf{a}}(t)\|_* \|\mathbf{z} - \mathbf{x}(t)\| + \|\mathbf{a}(t)\|_* \|\mathbf{v} - \dot{\mathbf{x}}(t)\|.$$

Proof. Note that $g_1 : t \mapsto \mathbf{a}(t)^{\mathrm{T}}\mathbf{x}(t) - b(t)$ is absolutely continuous, as the sum of the product of absolutely continuous functions. Obviously, $g_2 : t \mapsto 0$ is absolutely continuous, and so g, as the maximum of the two, can be written as $g(t) = 1/2(g_1(t) + g_2(t) + |g_1(t) - g_2(t)|)$. One notes this is absolutely continuous, since the composition of a Lipschitz continuous function with an absolutely continuous function is absolutely continuous, and again the sum of absolutely continuous functions is absolutely continuous.

On the set of t such that $\mathbf{a}(t)^{\mathrm{T}}\mathbf{x}(t) > b(t)$, we have $g(\cdot) = \mathbf{a}(\cdot)^{\mathrm{T}}\mathbf{x}(\cdot) - b(\cdot)$. Since g is absolutely continuous, we have that for almost all t such that $\mathbf{a}(t)^{\mathrm{T}}\mathbf{x}(t) > b(t)$, $\dot{g}(t) = \dot{\mathbf{a}}(t)^{\mathrm{T}}\mathbf{x}(t) + \mathbf{a}(t)^{\mathrm{T}}\dot{\mathbf{x}}(t) - \dot{b}(t)$. Thus, for any $(\mathbf{v}, \mathbf{z})$ such that $\mathbf{a}(t)^{\mathrm{T}}\mathbf{v} + \dot{\mathbf{a}}(t)^{\mathrm{T}}\mathbf{z} \le \dot{b}(t)$, we have $\dot{g}(t) + \mathbf{a}(t)^{\mathrm{T}}\mathbf{v} + \dot{\mathbf{a}}(t)^{\mathrm{T}}\mathbf{z} \le \dot{\mathbf{a}}(t)^{\mathrm{T}}\mathbf{x}(t) + \mathbf{a}(t)^{\mathrm{T}}\dot{\mathbf{x}}(t) - \dot{b}(t) + \dot{b}(t)$. It follows that $\dot{g}(t) \le \dot{\mathbf{a}}(t)^{\mathrm{T}}(\mathbf{x}(t) - \mathbf{z}) + \mathbf{a}(t)^{\mathrm{T}}(\dot{\mathbf{x}}(t) - \mathbf{v})$. Finally, from the generalization of the Cauchy-Schwarz inequality (that is, from the definition of the dual norm), we have $\dot{g}(t) \le \|\dot{\mathbf{a}}(t)\|_* \|\mathbf{z} - \mathbf{x}(t)\| + \|\mathbf{a}(t)\|_* \|\mathbf{v} - \dot{\mathbf{x}}(t)\|$. □

The following assumptions and theorem form the core of the general bounding theory. The way in which the matrix-valued mapping $\mathbf{A}$ and the set-valued mappings M_i satisfying Assumption 7.3.2 are defined provides the flexibility of the theory. Conceptually, the mappings can be thought of in the following ways: In the unconstrained case, M_i can be thought of as mapping to the i^{th} face of the polyhedral bounds. Meanwhile, in the constrained case, the constraints may be used to restrict the value that each M_i might take.

Assumption 7.3.1. *For any $\mathbf{z} \in D_x$, there exists a neighborhood $N(\mathbf{z})$ and $\alpha \in L^1(T)$ such that for almost every $t \in T$ and every $\mathbf{p}_t \in U(t)$*

$$\|\mathbf{f}(t, \mathbf{p}_t, \mathbf{z}_1) - \mathbf{f}(t, \mathbf{p}_t, \mathbf{z}_2)\| \le \alpha(t) \|\mathbf{z}_1 - \mathbf{z}_2\|,$$

for every $\mathbf{z}_1$ *and* $\mathbf{z}_2$ *in* $N(\mathbf{z}) \cap D_x$.

Assumption 7.3.2. *Consider the problem stated in Section 7.2.1. For some* $m \in \mathbb{N}$, *assume that for each* $i \in \{1, \dots, m\}$, $\mathbf{a}_i : T \to \mathbb{R}^{n_x}$ *is absolutely continuous, and* $\mathbf{A} : T \ni t \mapsto [\mathbf{a}_i(t)^{\mathrm{T}}] \in \mathbb{R}^{m \times n_x}$. *Assume* $D_M \subset T \times \mathbb{R}^m$, *and* $M_i : D_M \rightrightarrows \mathbb{R}^{n_x}$ *satisfy the following conditions for each* $i \in \{1, \dots, m\}$:

1. *For any* $\mathbf{d} \in \mathbb{R}^m$, *if there exists* $(t, \mathbf{u}) \in T \times \mathcal{U}$ *such that* $\mathbf{A}(t)\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ *and* $\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$ *for some solution* $\mathbf{x}(\cdot, \mathbf{u})$ *of IVP* (7.1), *then* $(t, \mathbf{d}) \in D_M$ *and* $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$.
2. *For any* $(t, \mathbf{d}) \in D_M$, *there exists a neighborhood* $N(\mathbf{d})$ *of* $\mathbf{d}$, $t' > t$, *and* $L_M > 0$ *such that for any* $(s, \mathbf{d}_1)$ *and* $(s, \mathbf{d}_2)$ *in* $((t, t') \times N(\mathbf{d})) \cap D_M$ *and* $\mathbf{z}_1 \in M_i(s, \mathbf{d}_1)$, *there exists a* $\mathbf{z}_2 \in M_i(s, \mathbf{d}_2)$ *such that*

$$\|\mathbf{z}_1 - \mathbf{z}_2\| \leq L_M \|\mathbf{d}_1 - \mathbf{d}_2\|_1 .$$

Theorem 7.3.1. *Let Assumptions 7.3.1 and 7.3.2 hold. If*

1. $\mathbf{b} : T \to \mathbb{R}^m$ *is absolutely continuous and* $B : T \ni t \mapsto \{\mathbf{z} : \mathbf{A}(t)\mathbf{z} \leq \mathbf{b}(t)\}$,
2. $X_0 \subset B(t_0)$,
3. *for almost every* $t \in T$ *and each* $i \in \{1, \dots, m\}$, $(t, \mathbf{b}(t)) \in D_M$ *and* $M_i(t, \mathbf{b}(t)) \subset D_x$,
4. *for almost every* $t \in T$ *and each* $i \in \{1, \dots, m\}$,

$$\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{f}(t, \mathbf{p}, \mathbf{z}) + \dot{\mathbf{a}}_i(t)^{\mathrm{T}}\mathbf{z} \leq \dot{b}_i(t), \quad \forall (\mathbf{p}, \mathbf{z}) \in U(t) \times M_i(t, \mathbf{b}(t)),$$

then for all $\mathbf{u} \in \mathcal{U}$ *and any solution* $\mathbf{x}(\cdot, \mathbf{u})$ *of IVP* (7.1), $\mathbf{x}(t, \mathbf{u}) \in B(t)$, *for all* $t \in T$.

Proof. Fix $\mathbf{u} \in \mathcal{U}$. If no solution of IVP (7.1) exists for this $\mathbf{u}$, then the conclusion of the theorem holds trivially. Otherwise, choose some solution and for convenience use the abbreviation $\mathbf{x}(t) \equiv \mathbf{x}(t, \mathbf{u})$. For each $t \in T$ and $i \in \{1, \dots, m\}$, let $g_i(t) = \max\{0, \mathbf{a}_i(t)^{\mathrm{T}}\mathbf{x}(t) - b_i(t)\}$. By Lemma 7.3.1, each g_i is absolutely continuous. It follows that $\mathbf{A}(t)\mathbf{x}(t) \leq \mathbf{b}(t) + \mathbf{g}(t)$. Consequently, $\mathbf{g}(t) = \mathbf{0}$ implies $\mathbf{x}(t) \in B(t)$, and by the contrapositive $\mathbf{x}(t) \notin B(t)$ implies $\mathbf{g}(t) \neq \mathbf{0}$. Thus, for a contradiction, suppose that there exists a $\widetilde{t} \in T$ such that $\mathbf{x}(\widetilde{t}) \notin B(\widetilde{t})$. Then the set $T_v = \{t \in T : \|\mathbf{g}(t)\|_1 > 0\}$ is nonempty.

Let $t_1 = \inf T_v$. By Hypothesis 2, $\mathbf{g}(t_0) = \mathbf{0}$ and so by continuity of $\mathbf{g}$, $\|\mathbf{g}(t_1)\|_1 = 0$. Furthermore, there exists $t_2 > t_1$ and index set I such that $g_i(t) = 0$ for $i \notin I$ and $t \in [t_1, t_2)$, and $\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{x}(t) = b_i(t) + g_i(t)$ for $i \in I$ and $t \in [t_1, t_2)$. Explicitly, for each i define $T_i \equiv \{t : g_i(t) > 0\}$. By continuity of $\mathbf{g}$, each T_i is open. Let $I = \{i : t_1 = \inf T_i\}$ (which must be nonempty) and then choose $t_2 > t_1$ such that $(t_1, t_2) \subset \bigcap_{i \in I} T_i$ and $(t_1, t_2) \cap (\bigcup_{i \notin I} T_i) = \varnothing$.

Then by Assumption 7.3.2, $(t, \mathbf{b}(t) + \mathbf{g}(t)) \in D_M$ and $\mathbf{x}(t) \in M_i(t, \mathbf{b}(t) + \mathbf{g}(t))$ for $i \in I$, $t \in [t_1, t_2)$. Without loss of generality, let $N(\mathbf{b}(t_1))$, $t_3 > t_1$, and $L_M > 0$ satisfy Condition 2 of Assumption 7.3.2 at the point $\mathbf{b}(t_1)$, for each $i \in I$. Since $\mathbf{b}$ and $\mathbf{g}$ are continuous, there exists a $t_4 \in (t_1, \min\{t_2, t_3\})$ such that $\mathbf{b}(t), (\mathbf{b}(t) + \mathbf{g}(t)) \in N(\mathbf{b}(t_1))$ for each $t \in (t_1, t_4)$. Along with Hypothesis 3, it follows that for $i \in I$ and almost every $t \in (t_1, t_4)$, there exists an element $\mathbf{z}_i(t) \in M_i(t, \mathbf{b}(t))$ with

$$\|\mathbf{z}_i(t) - \mathbf{x}(t)\| \leq L_M \|\mathbf{g}(t)\|_1 . \tag{7.3}$$

Let $N(\mathbf{x}(t_1))$, and $\alpha \in L^1(T)$ satisfy Assumption 7.3.1 at the point $\mathbf{x}(t_1)$. Since $\mathbf{x}$ and $\|\mathbf{g}\|_1$ are continuous, using Inequality (7.3) and the triangle inequality

$$\|\mathbf{z}_i(t) - \mathbf{x}(t_1)\| \leq \|\mathbf{z}_i(t) - \mathbf{x}(t)\| + \|\mathbf{x}(t) - \mathbf{x}(t_1)\| ,$$

there exists a $t_5 \in (t_1, t_4)$ such that $\mathbf{z}_i(t)$, $\mathbf{x}(t) \in N(\mathbf{x}(t_1))$, for all $i \in I$ and almost every $t \in (t_1, t_5)$. Consequently,

$$\|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t))\| \leq \alpha(t) \|\mathbf{z}_i(t) - \mathbf{x}(t)\| , \quad a.e.\ t \in (t_1, t_5). \tag{7.4}$$

But by Hypothesis 4, $\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) + \dot{\mathbf{a}}_i(t)^{\mathrm{T}}\mathbf{z}_i(t) \leq \dot{b}_i(t)$ which by Lemma 7.3.1 means

$$\begin{aligned} \dot{g}_i(t) &\leq \|\dot{\mathbf{a}}_i(t)\|_* \|\mathbf{z}_i(t) - \mathbf{x}(t)\| + \|\mathbf{a}_i(t)\|_* \|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \dot{\mathbf{x}}(t)\| \\ &= \|\dot{\mathbf{a}}_i(t)\|_* \|\mathbf{z}_i(t) - \mathbf{x}(t)\| + \|\mathbf{a}_i(t)\|_* \|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t))\| . \end{aligned}$$

Combining this with Inequalities (7.3) and (7.4) we have

$$\dot{g}_i(t) \leq L_M \left(\|\dot{\mathbf{a}}_i(t)\|_* + \alpha(t) \|\mathbf{a}_i(t)\|_*\right) \|\mathbf{g}(t)\|_1 , \quad a.e.\ t \in (t_1, t_5).$$

Since this holds for each $i \in I$,

$$\sum_{i \in I} \dot{g}_i(t) \leq L_M \sum_{i \in I} \left(\|\dot{\mathbf{a}}_i(t)\|_* + \alpha(t) \|\mathbf{a}_i(t)\|_* \right) \|\mathbf{g}(t)\|_1 , \quad a.e.\ t \in (t_1, t_5).$$

Note that $\beta : t \mapsto \left|\sum_{i \in I} \left(\|\dot{\mathbf{a}}_i(t)\|_* + \alpha(t) \|\mathbf{a}_i(t)\|_* \right)\right|$ is in $L^1(T)$. Since $g_i(t) > 0$ for each $i \in I$ and $g_i(t) = 0$ for each $i \notin I$, we have $\|\mathbf{g}(t)\|_1 = \sum_{i \in I} g_i(t)$ and so

$$\sum_{i \in I} \dot{g}_i(t) \leq L_M \beta(t) \sum_{i \in I} g_i(t), \quad a.e.\ t \in (t_1, t_5),$$

to which we can apply Gronwall's inequality (see for instance [209]) to get

$$\sum_{i \in I} g_i(t) \leq \sum_{i \in I} g_i(t_1) \exp\left(\int_{[t_1, t]} L_M \beta(s) ds \right), \ \forall t \in [t_1, t_5].$$

But since $\sum_i g_i(t_1) = 0$, this yields $\sum_i g_i(t) \leq 0$, and since each g_i is nonnegative always and $g_i(t) = 0$ for each $i \notin I$, we have $g_i(t) = 0$ for all i and all $t \in (t_1, t_5) \subset T_v$, which is a contradiction. Since the choices of $\mathbf{u} \in \mathcal{U}$ and corresponding solution were arbitrary, the result follows. □

7.4 Implementations for unconstrained systems

In this section, various ways to satisfy Assumption 7.3.2 are discussed. Each subsection focuses on a specific implementation of the general theory, the implications for the class of systems to which the specific method applies, and the connections to previous work. However, none of these specific instances of the theory in this section use the state constraint information X_C, and so the resulting bounds enclose all unconstrained solutions of IVP (7.1). For concreteness, one can take $X_C(t) = \mathbb{R}^{n_x}$ for all $t \in T$, so that an unconstrained solution is equivalent to a solution.

7.4.1 Interval bounds

Perhaps the most simple way to define $\mathbf{A}$, D_M and M_i so that they satisfy Assumption 7.3.2 is to let

$$\begin{aligned}
&\mathbf{A} : t \mapsto \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} \\
&D_M = T \times \{\mathbf{d} \in \mathbb{R}^{2n_x} : -d_i \leq d_{i+n_x}, \forall i \in \{1, \dots, n_x\}\}, \text{ and} \\
&M_i : (t, \mathbf{d}) \mapsto \{\mathbf{z} : \mathbf{A}(t)\mathbf{z} \leq \mathbf{d}, \mathbf{a}_i(t)^{\mathrm{T}}\mathbf{z} = d_i\}.
\end{aligned}$$

The result is that the bounds B constructed in Theorem 7.3.1 describe a time-varying interval enclosure of the reachable sets of (the unconstrained) IVP (7.1). A rigorous proof that these definitions satisfy Assumption 7.3.2 is postponed until §7.5.1, as it is a special case of the definitions introduced in that section. On the other hand, it is not particularly difficult to verify this specific case.

Writing $\mathbf{b} = (-\mathbf{x}^L, \mathbf{x}^U)$, the key hypothesis (Hypothesis 4) in Theorem 7.3.1 becomes

$$\dot{x}_j^L(t) \leq \inf\{f_j(t, \mathbf{p}, \mathbf{z}) : \mathbf{p} \in U(t), \mathbf{x}^L(t) \leq \mathbf{z} \leq \mathbf{x}^U(t), z_j = x_j^L(t)\}, \tag{7.5a}$$

$$\dot{x}_j^U(t) \geq \sup\{f_j(t, \mathbf{p}, \mathbf{z}) : \mathbf{p} \in U(t), \mathbf{x}^L(t) \leq \mathbf{z} \leq \mathbf{x}^U(t), z_j = x_j^U(t)\}, \tag{7.5b}$$

for each j and almost every t. Assuming $\mathbf{f}$ is continuous and U is compact-valued, the infima and suprema are finite, and IVPs in ODEs can be constructed whose solutions (if they exist) give $(\mathbf{x}^L, \mathbf{x}^U)$ satisfying (7.5).

More specifically, if U is interval-valued, then the feasible sets of the optimization problems in Inequalities (7.5) are intervals, and tools from interval arithmetic can be used to estimate the value of the optimization problems efficiently. This forms the basis of many early bounding methods for the unconstrained problem, such as those described in [72]. The shortcomings of interval arithmetic motivate a lot of subsequent work, some of which will be discussed in later sections. The details of numerical implementations based on this theory are also a large part of the literature, typically when properties such as monotonicity and other problem structure can be used; see [97, 139, 152] for further work.

7.4.2 Polyhedral bounds

If we allow $\mathbf{A}$ to be time-varying, other possible definitions of D_M and M_i consistent with Assumption 7.3.2 are given in the following result.

Proposition 7.4.1. *Given $m \in \mathbb{N}$, and $\mathbf{a}_i : T \to \mathbb{R}^{n_x}$ for $i \in \{1, \dots, m\}$ which are absolutely continuous, suppose that for all $i \in \{1, \dots, m\}$, $\mathbf{a}_i(t) \neq \mathbf{0}$, for all $t \in T$. Then $\mathbf{A} : t \mapsto [\mathbf{a}_i(t)^{\mathrm{T}}] \in \mathbb{R}^{m \times n_x}$,*

$$D_M = T \times \mathbb{R}^m \text{ and} \tag{7.6}$$

$$M_i : (t, \mathbf{d}) \mapsto \{\mathbf{z} : \mathbf{a}_i(t)^{\mathrm{T}}\mathbf{z} = d_i\} \tag{7.7}$$

satisfy Assumption 7.3.2.

Proof. To see that Condition 1 of Assumption 7.3.2 holds, choose any $i \in \{1, \dots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ with $\mathbf{A}(t)\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$ (assuming some unconstrained solution of IVP (7.1) exists for this $\mathbf{u}$). It is clear that $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$ and $(t, \mathbf{d}) \in D_M$.

To see that Condition 2 holds, choose any $(s, \mathbf{d}^1)$ and $(s, \mathbf{d}^2) \in D_M$ and $\mathbf{z}_1 \in M_i(s, \mathbf{d}^1)$. If $M_i(s, \mathbf{d}^1)$ is empty, then the condition holds trivially. Otherwise, $\mathbf{z}_2 = \mathbf{z}_1 + \frac{(d_i^2 - d_i^1)}{\|\mathbf{a}_i(s)\|_2^2}\mathbf{a}_i(s)$ is in $M_i(s, \mathbf{d}^2)$. Thus

$$\|\mathbf{z}_2 - \mathbf{z}_1\| \leq \frac{\|\mathbf{a}_i(s)\|}{\|\mathbf{a}_i(s)\|_2^2} \left\|\mathbf{d}^2 - \mathbf{d}^1\right\|_1.$$

Since $\mathbf{a}_i$ is continuous and nonzero on T, there exists a $L_M > 0$ such that $\frac{\|\mathbf{a}_i(s)\|}{\|\mathbf{a}_i(s)\|_2^2} \leq L_M$, for all $s \in T$ and all i. □

Note that M_i as defined in Eqn. (7.7) is unbounded. Consequently, only systems with a special structure and specially constructed $\mathbf{a}_i$ will be able to satisfy Hypothesis 4 in Theorem 7.3.1. One such instance is when $\mathbf{f}$ is affine with respect to the states; i.e. when it has the form $\mathbf{f}(t, \mathbf{p}, \mathbf{z}) = \mathbf{F}(t)\mathbf{z} + \mathbf{g}(t, \mathbf{p})$. In this case, if we let $\mathbf{a}$ be the solution of an adjoint-like system

$$\dot{\mathbf{a}}(t)^{\mathrm{T}} = -\mathbf{a}(t)^{\mathrm{T}}\mathbf{F}(t), \quad a.e.\ t \in T,$$

then for almost every t

$$\begin{aligned}\mathbf{a}(t)^{\mathrm{T}}\mathbf{f}(t,\mathbf{p},\mathbf{z}) + \dot{\mathbf{a}}(t)^{\mathrm{T}}\mathbf{z} &= \mathbf{a}(t)^{\mathrm{T}}\mathbf{g}(t,\mathbf{p}) + \mathbf{a}(t)^{\mathrm{T}}\mathbf{F}(t)\mathbf{z} + \dot{\mathbf{a}}(t)^{\mathrm{T}}\mathbf{z}\\ &= \mathbf{a}(t)^{\mathrm{T}}\mathbf{g}(t,\mathbf{p}).\end{aligned}$$

This leads to the following result.

Proposition 7.4.2. *Suppose that $D_x = \mathbb{R}^{n_x}$ and $\mathbf{f}$ has the form*

$$\mathbf{f}(t,\mathbf{p},\mathbf{z}) = \mathbf{F}(t)\mathbf{z} + \mathbf{g}(t,\mathbf{p}),$$

where $\mathbf{g} : T \times D_u \to \mathbb{R}^{n_x}$, and $\mathbf{F} : T \to \mathbb{R}^{n_x \times n_x}$ has bounded induced norm: $\|\mathbf{F}(t)\| \le L_f$, for almost every $t \in T$. Let $m \in \mathbb{N}$. Assume $\mathbf{b} : T \to \mathbb{R}^m$ and $\mathbf{a}_i : T \to \mathbb{R}^{n_x}$, for $i \in \{1, \dots, m\}$, are absolutely continuous mappings with $\mathbf{a}_i(t) \neq \mathbf{0}$ for all t and each i. Let $B : t \mapsto \{\mathbf{z} : \mathbf{a}_i(t)^{\mathrm{T}}\mathbf{z} \le b_i(t), \forall i \in \{1, \dots, m\}\}$. If in addition, for $i \in \{1, \dots, m\}$,

$$\begin{aligned}&\dot{\mathbf{a}}_i(t)^{\mathrm{T}} = -\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{F}(t), \quad a.e.\ t \in T,\\ &\dot{b}_i(t) \ge \sup\{\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{g}(t,\mathbf{p}) : \mathbf{p} \in U(t)\}, \quad a.e.\ t \in T,\\ &X_0 \subset B(t_0),\end{aligned}$$

then for all $\mathbf{u} \in \mathcal{U}$ and any unconstrained solution $\mathbf{x}(\cdot,\mathbf{u})$ of IVP (7.1), $\mathbf{x}(t,\mathbf{u}) \in B(t)$, for all $t \in T$.

Proof. Note that Assumption 7.3.1 is satisfied, and although they are not actually used in the proposition, we can define $\mathbf{A}$, D_M, and M_i as in Proposition 7.4.1 so that Assumption 7.3.2 holds. Then, it is clear that all the hypotheses of Theorem 7.3.1 hold, since Hypothesis 4 becomes

$$\dot{b}_i(t) \ge \mathbf{a}_i(t)^{\mathrm{T}}\mathbf{g}(t,\mathbf{p}), \quad \forall \mathbf{p} \in U(t),$$

for almost every $t \in T$, which is satisfied by assumption. The result follows from Theorem 7.3.1. □

It is interesting to note that the definitions of D_M and M_i from Proposition 7.4.1 are not explicitly used in the proof of Proposition 7.4.2; merely their existence was needed. Meanwhile, further assumptions on $\mathbf{F}$, $\mathbf{g}$, and U are required to ensure that some $\mathbf{a}_i$ and $\mathbf{b}$

actually exist which satisfy the assumptions of Proposition 7.4.2.

Previous work that follows along these lines includes [73, 84]. The results in [84] are largely the same, although the derivation follows from arguments involving the Hamilton-Jacobi-Isaacs PDE. As a result, stronger conclusions can be made, such as the claim that in the completely linear case ($\mathbf{f}(t, \mathbf{p}, \mathbf{z}) = \mathbf{F}_x(t)\mathbf{z} + \mathbf{F}_u(t)\mathbf{p}$), the hyperplanes defining B are supporting hyperplanes to the exact reachable set of the unconstrained IVP. Another result from [84] allows an extra, bounded, nonlinear term to be added to the dynamics: $\mathbf{f}(t, \mathbf{p}, \mathbf{z}) = \mathbf{F}_x(t)\mathbf{z} + \mathbf{F}_u(t)\mathbf{p} + \phi(t, \mathbf{z})$, with $\|\phi(t, \mathbf{z})\| \leq \beta(t)$ for all $(t, \mathbf{z}) \in T \times D_x$ and for some bounded $\beta : T \to \mathbb{R}$. Modification of Proposition 7.4.2 to take this into account is straightforward.

Meanwhile, the work in [73] also constructs polyhedral bounds for an affine system, with the normals of the defining hyperplanes determined from the solution of an adjoint system. However, that work essentially deals with specific constraint information; only solutions with $\mathbf{x}(t, \mathbf{u}) \geq \mathbf{0}$ are of interest, and the proof of the main result depends on this fact. Currently, it is unclear how to use the constraint information with the present theory when the $\mathbf{a}_i$ are time-varying.

7.4.3 Affine relaxations

Another implementation of the theory permits the propagation of affine relaxations of the solutions of parametric ODEs.

Proposition 7.4.3. *Given $(n_y, n_p) \in \mathbb{N}^2$ and $P \subset \mathbb{R}^{n_p}$, let $n_x = n_y + n_p$ and P be nonempty. Suppose that for all $\mathbf{u} \in \mathcal{U}$ and for all unconstrained solutions of IVP* (7.1)*, $\mathbf{x}(\cdot, \mathbf{u}) = (\mathbf{x}_y(\cdot, \mathbf{u}), \mathbf{x}_p(\cdot, \mathbf{u}))$, satisfy $\mathbf{x}_p(t, \mathbf{u}) \in P$, for all $t \in T$. Given $\mathbf{a}_{p,j} : T \to \mathbb{R}^{n_p}$ for $j \in \{1, \dots, n_y\}$ which are absolutely continuous, suppose $\mathbf{A}_p : t \mapsto [\mathbf{a}_{p,j}(t)^{\mathrm{T}}]$, $m = 2n_y$, and*

$$\mathbf{A} : t \mapsto [\mathbf{a}_i(t)^{\mathrm{T}}] = \begin{bmatrix} -\mathbf{I} & \mathbf{A}_p(t) \\ \mathbf{I} & -\mathbf{A}_p(t) \end{bmatrix}.$$

Then $\mathbf{A}$,

$$D_M = \{(t, \mathbf{d}) \in T \times \mathbb{R}^m : -d_j \leq d_{j+n_y}, \forall j \in \{1, \dots, n_y\}\}, \quad \textit{and}$$

$$M_i : (t, \mathbf{d}) \mapsto \{\mathbf{z} = (\mathbf{y}, \mathbf{p}) : \mathbf{A}(t)\mathbf{z} \leq \mathbf{d}, \mathbf{a}_i(t)^{\mathrm{T}}\mathbf{z} = d_i, \mathbf{p} \in P\}$$

satisfy Assumption 7.3.2.

Proof. To see that Condition 1 of Assumption 7.3.2 holds, choose any $i \in \{1, \dots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ with $\mathbf{A}(t)\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$ (assuming some unconstrained solution of IVP (7.1) exists for this $\mathbf{u}$). It is clear that $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$, and from the form of $\mathbf{A}$, we have $(t, \mathbf{d}) \in D_M$.

Next, for $(t, \mathbf{d}, \mathbf{p}) \in T \times \mathbb{R}^m \times \mathbb{R}^{n_p}$, define $F_i(t, \mathbf{d}, \mathbf{p})$ for $i \leq n_y$ by

$$F_i(t, \mathbf{d}, \mathbf{p}) = \left\{ \widetilde{\mathbf{y}} : \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} \widetilde{\mathbf{y}} + \begin{bmatrix} \mathbf{A}_p(t) \\ -\mathbf{A}_p(t) \end{bmatrix} \mathbf{p} \leq \mathbf{d}, -\widetilde{y}_i + \mathbf{a}_{p,i}(t)^{\mathrm{T}}\mathbf{p} = d_i \right\}$$

and similarly for $i > n_y$. We establish that for all $(t, \mathbf{d}) \in D_M$, $\mathbf{p} \in P$, and for all i, $F_i(t, \mathbf{d}, \mathbf{p})$ is nonempty. Choose $(t, \mathbf{d}) \in D_M$ and $\mathbf{p} \in P$, and assume $i \leq n_y$. Then construct $\mathbf{y} \in \mathbb{R}^{n_y}$ by letting $y_j = -(d_j - \mathbf{a}_{p,j}(t)^{\mathrm{T}}\mathbf{p})$ for all j. Then $\mathbf{y}$ is in $F_i(t, \mathbf{d}, \mathbf{p})$: Clearly the equality constraint $-y_i + \mathbf{a}_{p,i}(t)^{\mathrm{T}}\mathbf{p} = d_i$ holds. Further, all the inequality constraints $-y_j + \mathbf{a}_{p,j}(t)^{\mathrm{T}}\mathbf{p} \leq d_j$ hold with equality, and by definition of D_M the other constraints also hold: $y_j - \mathbf{a}_{p,j}(t)^{\mathrm{T}}\mathbf{p} = -d_j \leq d_{j+n_y}$.

Now, to see that Condition 2 of Assumption 7.3.2 holds, choose any $(s, \mathbf{d})$, $(s, \mathbf{d}') \in D_M$ and $(\mathbf{y}, \mathbf{p}) \in M_i(s, \mathbf{d})$. Note that $F_i(s, \mathbf{d}, \mathbf{p})$ and $F_i(s, \mathbf{d}', \mathbf{p})$ are nonempty by the argument above. Note that $\mathbf{y} \in F_i(s, \mathbf{d}, \mathbf{p})$. Since F_i is a polyhedral set (in fact, an interval set) parameterized by the right-hand sides of its constraints, by Lemma 2.4.2, there exists $L_i > 0$ (which is independent of s and $\mathbf{p}$) such that for some $\mathbf{y}' \in F_i(s, \mathbf{d}', \mathbf{p})$

$$\left\|\mathbf{y} - \mathbf{y}'\right\| \leq L_i \left\|\mathbf{d} - \mathbf{d}'\right\|_1 .$$

Finally, note that $(\mathbf{y}', \mathbf{p}) \in M_i(s, \mathbf{d}')$. A similar argument establishes the case when $i > n_y$ and so choosing $L_M \geq L_i$ for all i yields the result. □

One application of the definitions in Proposition 7.4.3 is to the case when the dynamics depend on uncertain, but constant in time, parameters. Certainly these inputs can be incorporated in the set of time-varying inputs $\mathcal{U}$, but enforcing the fact that they are constant in time might yield better bounds. Using the notation from Proposition 7.4.3, these parameters are interpreted as states with zero time derivative:

$$\dot{\mathbf{x}}(t,\mathbf{u}) = \begin{bmatrix} \dot{\mathbf{x}}_y(t,\mathbf{u}) \\ \dot{\mathbf{x}}_p(t,\mathbf{u}) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_y(t,\mathbf{u}(t),\mathbf{x}_y(t,\mathbf{u}),\mathbf{x}_p(t,\mathbf{u})) \\ \mathbf{0} \end{bmatrix} = \mathbf{f}(t,\mathbf{u}(t),\mathbf{x}(t,\mathbf{u})), \quad a.e.\ t \in T. \tag{7.8}$$

Thus for any unconstrained solution of (7.8), $\mathbf{x}_p(t,\mathbf{u}) = \mathbf{x}_p(t_0,\mathbf{u})$ for all $t \in T$. Then assuming $X_0 \subset X_0^y \times P$ for some $X_0^y \subset \mathbb{R}^{n_y}$ and $P \subset \mathbb{R}^{n_p}$, the condition $\mathbf{x}_p(t,\mathbf{u}) \in P$ for all t must be satisfied.

Using the definitions in Proposition 7.4.3, the differential inequalities in Hypothesis 4 of Theorem 7.3.1 become

$$-f_{y,j}(t,\mathbf{v},\mathbf{y},\mathbf{p}) + \dot{\mathbf{a}}_{p,j}(t)^{\mathrm{T}}\mathbf{p} \le \dot{b}_j(t), \qquad \forall \mathbf{v} \in U(t), (\mathbf{y},\mathbf{p}) \in M_j(t,\mathbf{b}(t)), \tag{7.9a}$$

$$f_{y,j}(t,\mathbf{v},\mathbf{y},\mathbf{p}) - \dot{\mathbf{a}}_{p,j}(t)^{\mathrm{T}}\mathbf{p} \le \dot{b}_{j+n_y}(t), \qquad \forall \mathbf{v} \in U(t), (\mathbf{y},\mathbf{p}) \in M_{j+n_y}(t,\mathbf{b}(t)), \tag{7.9b}$$

for all $j \in \{1,\dots,n_y\}$. With the definitions in Proposition 7.4.3, the bounds $B : t \mapsto \{\mathbf{z} : \mathbf{A}(t)\mathbf{z} \le \mathbf{b}(t)\}$ imply

$$\mathbf{a}_{p,j}(t)^{\mathrm{T}}\mathbf{p} - b_j(t) \le x_{y,j}(t,\mathbf{u},\mathbf{p}) \le \mathbf{a}_{p,j}(t)^{\mathrm{T}}\mathbf{p} + b_{j+n_y}(t), \quad \forall j \in \{1,\dots,n_y\}, \tag{7.10}$$

for all $t \in T$, and for any $(\mathbf{u},\mathbf{p}) \in \mathcal{U} \times P$ for which an unconstrained solution of

$$\dot{\mathbf{x}}_y(t,\mathbf{u}) = \mathbf{f}_y(t,\mathbf{u}(t),\mathbf{x}_y(t,\mathbf{u}),\mathbf{p}), \quad a.e.\ t \in T, \tag{7.11a}$$

$$(\mathbf{x}_y(t_0,\mathbf{u}),\mathbf{p}) \in X_0, \tag{7.11b}$$

exists. That is to say, we obtain affine relaxations of the solutions of the IVP in parametric ODEs (7.11). In practice, satisfaction of Inequalities (7.9) can be achieved using affine relaxations of $\mathbf{f}_y$; see also Sections 7.5.3 and 7.6.2 for further discussion.

To explore connections with other work, simplify the problem by assuming that there are not any time-varying inputs $\mathbf{u}$ and drop them from the notation. For some reference $\widehat{\mathbf{p}} \in P$, let $\mathbf{x}_y^{\mathcal{P}} : T \times P \to \mathbb{R}^{n_y}$ be defined by

$$\mathbf{x}_y^{\mathcal{P}}(t,\mathbf{p}) = \mathbf{x}_y(t,\widehat{\mathbf{p}}) + \mathbf{A}_p(t)(\mathbf{p} - \widehat{\mathbf{p}})$$

and $R : t \mapsto [\mathbf{r}^L(t), \mathbf{r}^U(t)]$, with $\mathbf{r}^L$ and $\mathbf{r}^U$ defined componentwise by

$$r_j^L(t) = -b_j(t) - x_{y,j}(t, \widehat{\mathbf{p}}) + \mathbf{a}_{p,j}(t)^{\mathrm{T}}\widehat{\mathbf{p}}, \tag{7.12a}$$

$$r_j^U(t) = b_{j+n_y}(t) - x_{y,j}(t, \widehat{\mathbf{p}}) + \mathbf{a}_{p,j}(t)^{\mathrm{T}}\widehat{\mathbf{p}}. \tag{7.12b}$$

Then by Inequalities (7.10), we have $\mathbf{x}_y(t, \mathbf{p}) \in \{\mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p})\} + R(t)$ for all t (where Minkowski addition is being used), or

$$\mathbf{x}_y(t, \mathbf{p}) \in [\mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{r}^L(t), \mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{r}^U(t)].$$

Similarly, one can confirm that

$$\begin{aligned} &M_j(t, \mathbf{b}(t)) = \\ &\quad \left\{(\mathbf{y}, \mathbf{p}) : \mathbf{y} \in [\mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{r}^L(t), \mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{r}^U(t)], y_j = x_{y,j}^{\mathcal{P}}(t, \mathbf{p}) + r_j^L(t), \mathbf{p} \in P\right\}, \\ &M_{j+n_y}(t, \mathbf{b}(t)) = \\ &\quad \left\{(\mathbf{y}, \mathbf{p}) : \mathbf{y} \in [\mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{r}^L(t), \mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{r}^U(t)], y_j = x_{y,j}^{\mathcal{P}}(t, \mathbf{p}) + r_j^U(t), \mathbf{p} \in P\right\}, \end{aligned}$$

for all j. Going further, taking the time derivative of Equations (7.12) and using Inequalities (7.9) we obtain

$$\begin{aligned} &\dot{r}_j^L(t) \le \inf\left\{f_{y,j}(t, \mathbf{y}, \mathbf{p}) - f_{y,j}(t, \mathbf{x}_y(t, \widehat{\mathbf{p}}), \widehat{\mathbf{p}}) - \dot{\mathbf{a}}_{p,j}(t)^{\mathrm{T}}(\mathbf{p} - \widehat{\mathbf{p}}) : (\mathbf{y}, \mathbf{p}) \in M_j(t, \mathbf{b}(t))\right\}, \\ &\dot{r}_j^U(t) \ge \sup\left\{f_{y,j}(t, \mathbf{y}, \mathbf{p}) - f_{y,j}(t, \mathbf{x}_y(t, \widehat{\mathbf{p}}), \widehat{\mathbf{p}}) - \dot{\mathbf{a}}_{p,j}(t)^{\mathrm{T}}(\mathbf{p} - \widehat{\mathbf{p}}) : (\mathbf{y}, \mathbf{p}) \in M_{j+n_y}(t, \mathbf{b}(t))\right\}. \end{aligned}$$

Thus $(\mathbf{r}^L, \mathbf{r}^U)$ satisfy

$$\begin{aligned} &\dot{r}_j^L(t) \le \inf\left\{f_{y,j}(t, \mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{q}, \mathbf{p}) - \dot{x}_{y,j}^{\mathcal{P}}(t, \mathbf{p}) : (\mathbf{p}, \mathbf{q}) \in P \times [\mathbf{r}^L(t), \mathbf{r}^U(t)], q_j = r_j^L(t)\right\}, \\ &\dot{r}_j^U(t) \ge \sup\left\{f_{y,j}(t, \mathbf{x}_y^{\mathcal{P}}(t, \mathbf{p}) + \mathbf{q}, \mathbf{p}) - \dot{x}_{y,j}^{\mathcal{P}}(t, \mathbf{p}) : (\mathbf{p}, \mathbf{q}) \in P \times [\mathbf{r}^L(t), \mathbf{r}^U(t)], q_j = r_j^U(t)\right\}, \end{aligned}$$

for all j and for almost every t. These are the same conditions established in [37], in the specific case that

$$\mathbf{A}_p(t) = \frac{\partial \mathbf{x}_y}{\partial \mathbf{p}}(t, \widehat{\mathbf{p}})$$

(assuming that these sensitivities exist). In this case, $(\mathbf{x}_y^{\mathcal{P}}(t, \cdot), R(t))$ constitutes a first-order

Taylor model of $\mathbf{x}_y(t,\cdot)$, using the terminology from [37, 107, 202].

The derivation of this result in [37] proceeds by applying the interval bounding method of §7.4.1 to the remainder function

$$\mathbf{r}(t,\mathbf{p}) = \mathbf{x}_y(t,\mathbf{p}) - \mathbf{x}_y^{\mathcal{P}}(t,\mathbf{p}),$$

which satisfies the differential equation

$$\dot{\mathbf{r}}(t,\mathbf{p}) = \mathbf{f}_y\left(t, \mathbf{x}_y^{\mathcal{P}}(t,\mathbf{p}) + \mathbf{r}(t,\mathbf{p}), \mathbf{p}\right) - \dot{\mathbf{x}}_y^{\mathcal{P}}(t,\mathbf{p}), \quad a.e.\ t \in T.$$

This holds even when $\mathbf{x}_y^{\mathcal{P}}(t,\cdot)$ is a Taylor polynomial approximation of $\mathbf{x}_y(t,\cdot)$ of order greater than one. A generalization in [202] bounds the remainder function with either intervals or ellipsoids. The convergence properties of these Taylor models are discussed in [202]; §7.7.3 also provides an empirical convergence study for this method and others.

7.5 Implementations for constrained systems

In §7.4.2, the M_i mappings were not compact-valued, and as a consequence only certain systems could non-trivially satisfy the hypotheses of Theorem 7.3.1. Meanwhile, the M_i mappings in §7.4.1 are interval-valued, and thus also compact; however there is some inflexibility in only using intervals. These shortcomings motivate the instances of the general theory discussed in this section. At the expense of being restricted to using a constant $\mathbf{A}$, these implementations of the theory can use the information from each hyperplane to make the M_i, in essence, as small as possible. Along these same lines, the state constraints can be used to further restrict the size of each M_i. In this way, bounds which enclose all solutions, but not necessarily all unconstrained solutions, can be obtained.

7.5.1 Fast polyhedral bounds

Another instance of the mappings which satisfy Assumption 7.3.2 is given. These mappings allow one to use polyhedral-valued state constraint information X_C. The specific conditions are formalized in the following result.

Proposition 7.5.1. *Given* $m_c \in \mathbb{N}$, $\mathbf{A}_C \in \mathbb{R}^{m_c \times n_x}$, *and* $\mathbf{b}_C : T \to \mathbb{R}^{m_c}$, *assume that*

$X_C : t \mapsto \{\mathbf{z} : \mathbf{A}_C\mathbf{z} \le \mathbf{b}_C(t)\}$. *Let* $m \in \mathbb{N}$ *and* $\widehat{\mathbf{A}} = [\widehat{\mathbf{a}}_i^{\mathrm{T}}] \in \mathbb{R}^{m \times n_x}$ *be given. Let*

$$P_M : (t, \mathbf{d}) \mapsto \{\mathbf{z} : \widehat{\mathbf{A}}\mathbf{z} \le \mathbf{d}, \mathbf{A}_C\mathbf{z} \le \mathbf{b}_C(t)\}.$$

Then $\mathbf{A} : t \mapsto \widehat{\mathbf{A}}$,

$$D_M = \{(t, \mathbf{d}) \in T \times \mathbb{R}^m : P_M(t, \mathbf{d}) \neq \varnothing\}, \text{ and} \tag{7.13}$$

$$M_i : (t, \mathbf{d}) \mapsto \arg\max\{\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} : \widehat{\mathbf{A}}\mathbf{z} \le \mathbf{d}, \mathbf{A}_C\mathbf{z} \le \mathbf{b}_C(t)\} \tag{7.14}$$

satisfy Assumption 7.3.2.

Proof. To see that Condition 1 of Assumption 7.3.2 holds, choose any $i \in \{1, \dots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\widehat{\mathbf{A}}\mathbf{x}(t, \mathbf{u}) \le \mathbf{d}$ and $\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$. Since a solution $\mathbf{x}(\cdot, \mathbf{u})$ must satisfy $\mathbf{A}_C\mathbf{x}(t, \mathbf{u}) \le \mathbf{b}_C(t)$, it holds that $\mathbf{x}(t, \mathbf{u}) \in P_M(t, \mathbf{d})$, and thus $(t, \mathbf{d}) \in D_M$. Further, since $\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$, and any $\mathbf{z}$ such that $\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} > d_i$ would be infeasible in LP (7.14), we must have $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$.

Next, note that if $P_M(t, \mathbf{d})$ is nonempty, then $M_i(t, \mathbf{d})$ is nonempty for all i ($M_i(t, \mathbf{d})$ is the solution set of a linear program that must be feasible and bounded). Then to see that Condition 2 of Assumption 7.3.2 holds, choose any $(s, \mathbf{d}_1), (s, \mathbf{d}_2) \in D_M$. By definition of D_M and the previous observation, $M_i(s, \mathbf{d}_j)$ is nonempty for $i \in \{1, \dots, m\}$ and $j \in \{1, 2\}$. Applying Lemma 2.4.2, we have that there exists a $L > 0$ and for each $\mathbf{z}_1 \in M_i(s, \mathbf{d}_1)$, there exists a $\mathbf{z}_2 \in M_i(s, \mathbf{d}_2)$ such that

$$\|\mathbf{z}_1 - \mathbf{z}_2\| \le L \left\|(\mathbf{d}_1, \mathbf{b}_C(s)) - (\mathbf{d}_2, \mathbf{b}_C(s))\right\|_1 = L \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1.$$

□

Ignoring the state constraints and defining $\mathbf{A}$ appropriately, the interval bounds from §7.4.1 are regained. Connections with other theories are apparent when one interprets the state constraints instead as an *a priori* enclosure of the reachable set. Assume that it is known beforehand that for all $\mathbf{u} \in \mathcal{U}$ and all *unconstrained* solutions of IVP (7.1), $\mathbf{x}(t, \mathbf{u}) \in X_C(t)$ for all $t \in T$. In this case, X_C is called an *a priori* enclosure of the reachable set of the unconstrained IVP. With this interpretation, defining $\mathbf{A}$ so that interval bounds are obtained and assuming $\mathbf{b}_C$ is constant on T (i.e. assuming the *a priori* enclosure is constant) yields the theory in [75]. Allowing a more general (but still constant) $\mathbf{A}$ and permitting a time-varying $\mathbf{b}_C$ yields the theory in Ch. 6. Compared to the developments in §7.4.2, the benefit is that a more general form of $\mathbf{f}$ is allowed.

The critical hypothesis of Theorem 7.3.1 becomes

$$\dot{b}_i(t) \geq \sup\left\{\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{f}(t,\mathbf{p},\mathbf{z}) : \mathbf{p} \in U(t), \widehat{\mathbf{A}}\mathbf{z} \leq \mathbf{b}(t), \mathbf{A}_C\mathbf{z} \leq \mathbf{b}_C(t), \widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} = b_i^*(t,\mathbf{b}(t))\right\}, \quad (7.15)$$

for all i and almost every t, where $b_i^*(t,\mathbf{d}) = \sup\{\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} : \widehat{\mathbf{A}}\mathbf{z} \leq \mathbf{d}, \mathbf{A}_C\mathbf{z} \leq \mathbf{b}_C(t)\}$. To obtain an auxiliary IVP in ODEs whose solution is $\mathbf{b}$, we need (at least) to assume $\widehat{\mathbf{A}}$, $\mathbf{A}_C$ and $\mathbf{b}_C$ are chosen so that M_i in Eqn. (7.14) is compact-valued, and that in addition $\mathbf{f}$ is continuous and U is compact-valued. Further, one needs to assume that $\mathbf{b}_C$ is piecewise continuous to ensure that the auxiliary IVP in ODEs actually has a solution. The details of a numerical method based on this implementation are given in Ch. 6.

However the optimization problems in Inequality (7.15) are estimated, the definition of the M_i mappings involves the solution of linear programs (LPs). Although this can be done fairly efficiently, it cannot be made quite as efficient as the method in [168], which made the most significant step toward generalizing the idea of using *a priori* enclosures. This inspires the following section's development of another implementation of the general theory, which avoids the solution of linear programs.

7.5.2 Faster polyhedral bounds

In the previous section, the implementation of the theory calls for the repeated solution of LPs. In this section, another instance of the general theory is given which avoids this.

To do so, the interval-tightening operator originally discussed in §6.4.1 is required. For convenience, its definition and properties are restated in Algorithm 5 and Proposition 7.5.2, respectively.

Proposition 7.5.2. *For any* $(n_m, n) \in \mathbb{N}^2$*, let* $\mathbf{M} \in \mathbb{R}^{n_m \times n}$*. For any* $(\mathbf{v},\mathbf{w},\mathbf{d}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n_m}$ *with* $\mathbf{v} \leq \mathbf{w}$*, the interval-tightening operator* I_t *defined in Algorithm 5 satisfies* $I_t([\mathbf{v},\mathbf{w}],\mathbf{d};\mathbf{M}) \neq \varnothing$ *and*

$$\{\mathbf{z} \in [\mathbf{v},\mathbf{w}] : \mathbf{M}\mathbf{z} \leq \mathbf{d}\} \subset I_t([\mathbf{v},\mathbf{w}],\mathbf{d};\mathbf{M}) \subset [\mathbf{v},\mathbf{w}].$$

Algorithm 5 Definition of the interval-tightening operator I_t

```
Require: $(n_m, n) \in \mathbb{N}^2$, $\mathbf{M} = [m_{i,j}] \in \mathbb{R}^{n_m \times n}$, $\mathbf{d} \in \mathbb{R}^{n_m}$, $(\mathbf{v}, \mathbf{w}) \in \mathbb{R}^n \times \mathbb{R}^n$, $\mathbf{v} \leq \mathbf{w}$
  $(\widehat{\mathbf{v}}, \widehat{\mathbf{w}}) \leftarrow (\mathbf{v}, \mathbf{w})$
  for $i \in \{1, \dots, n_m\}$ do
    for $j \in \{1, \dots, n\}$ do
      if $m_{i,j} \neq 0$ then
        $\gamma \leftarrow \text{median}\left\{\widehat{v}_j, \widehat{w}_j, {}^1\!/_{m_{i,j}}\big(d_i + \sum_{k \neq j} \max\{-m_{i,k}\widehat{v}_k, -m_{i,k}\widehat{w}_k\}\big)\right\}$
        if $m_{i,j} > 0$ then
          $\widehat{w}_j \leftarrow \gamma$
        end if
        if $m_{i,j} < 0$ then
          $\widehat{v}_j \leftarrow \gamma$
        end if
      end if
    end for
  end for
  return $I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}) \leftarrow [\widehat{\mathbf{v}}, \widehat{\mathbf{w}}]$
```

Further, let $\mathbf{v}_{It}$ and $\mathbf{w}_{It}$ be the endpoints of I_t:

$$[\mathbf{v}_{It}(\mathbf{v}, \mathbf{w}, \mathbf{d}; \mathbf{M}), \mathbf{w}_{It}(\mathbf{v}, \mathbf{w}, \mathbf{d}; \mathbf{M})] = I_t([\mathbf{v}, \mathbf{w}], \mathbf{d}; \mathbf{M}).$$

Then there exists a $L_{\mathbf{M}} > 0$ such that for $(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1)$ and $(\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2)$ in $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n_m}$ with $\mathbf{v}_1 \leq \mathbf{w}_1$ and $\mathbf{v}_2 \leq \mathbf{w}_2$,

$$\|\mathbf{v}_{It}(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1; \mathbf{M}) - \mathbf{v}_{It}(\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2; \mathbf{M})\| \leq L_{\mathbf{M}} \|(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1) - (\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2)\|,$$

$$\|\mathbf{w}_{It}(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1; \mathbf{M}) - \mathbf{w}_{It}(\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2; \mathbf{M})\| \leq L_{\mathbf{M}} \|(\mathbf{v}_1, \mathbf{w}_1, \mathbf{d}_1) - (\mathbf{v}_2, \mathbf{w}_2, \mathbf{d}_2)\|.$$

More definitions of D_M and M_i satisfying Assumption 7.3.2 can be stated. In the following, a somewhat specific form of the matrix $\mathbf{A}$ is assumed. As a result, interval bounds are always available. This is not strictly necessary, but it simplifies the required constructions.

Proposition 7.5.3. *Given $m_c \in \mathbb{N}$, $\mathbf{A}_C \in \mathbb{R}^{m_c \times n_x}$, and $\mathbf{b}_C : T \to \mathbb{R}^{m_c}$, suppose that $X_C : t \mapsto \{\mathbf{z} : \mathbf{A}_C\mathbf{z} \leq \mathbf{b}_C(t)\}$. Given $m_e \in \mathbb{N}$ and $\mathbf{A}_e \in \mathbb{R}^{m_e \times n_x}$, let $m = 2n_x + m_e$ and*

$\widehat{\mathbf{A}} = [\widehat{\mathbf{a}}_i^{\mathrm{T}}] \in \mathbb{R}^{m \times n_x}$ *be*

$$\widehat{\mathbf{A}} = \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \\ \mathbf{A}_e \end{bmatrix}.$$

Let $D_M = T \times \mathbb{R}^m$ *and define*

$$\mathbf{v} : \mathbb{R}^m \ni \mathbf{d} \mapsto \begin{bmatrix} \min\{-d_1, d_{n_x+1}\} \\ \vdots \\ \min\{-d_{n_x}, d_{n_x+n_x}\} \end{bmatrix}, \quad \mathbf{w} : \mathbb{R}^m \ni \mathbf{d} \mapsto \begin{bmatrix} \max\{-d_1, d_{n_x+1}\} \\ \vdots \\ \max\{-d_{n_x}, d_{n_x+n_x}\} \end{bmatrix},$$

$$\mathbf{A}_{F,i} = \begin{bmatrix} -\widehat{\mathbf{a}}_i^{\mathrm{T}} \\ \widehat{\mathbf{A}} \\ \mathbf{A}_C \end{bmatrix}, \quad \mathbf{d}_{F,i} : (t, \mathbf{d}) \mapsto \begin{bmatrix} -d_i \\ \mathbf{d} \\ \mathbf{b}_C(t) \end{bmatrix},$$

$$F_i : (t, \mathbf{d}) \mapsto I_t([\mathbf{v}(\mathbf{d}), \mathbf{w}(\mathbf{d})], \mathbf{d}_{F,i}(t, \mathbf{d}); \mathbf{A}_{F,i}),$$

$$b_i^L : (t, \mathbf{d}) \mapsto \min\{\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} : \mathbf{z} \in F_i(t, \mathbf{d})\}, \quad b_i^U : (t, \mathbf{d}) \mapsto \max\{\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} : \mathbf{z} \in F_i(t, \mathbf{d})\},$$

$$M_i : (t, \mathbf{d}) \mapsto F_i(t, \mathbf{d}) \cap \left\{\mathbf{z} : \widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} = \operatorname{median}\left\{d_i, b_i^L(t, \mathbf{d}), b_i^U(t, \mathbf{d})\right\}\right\},$$

for $i \in \{1, \ldots, m\}$. *Then* $\mathbf{A} : t \mapsto \widehat{\mathbf{A}}$, D_M, *and* M_i *for* $i \in \{1, \ldots, m\}$ *satisfy Assumption 7.3.2.*

Proof. To see that Condition 1 of Assumption 7.3.2 holds, choose any $i \in \{1, \ldots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\widehat{\mathbf{A}}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$. This means we have $-\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) \leq -d_i$, and as well by assumption on the form of $\widehat{\mathbf{A}}$, we have $\mathbf{x}(t, \mathbf{u}) \in [\mathbf{v}(\mathbf{d}), \mathbf{w}(\mathbf{d})]$. Since a solution $\mathbf{x}(\cdot, \mathbf{u})$ must satisfy $\mathbf{A}_C\mathbf{x}(t, \mathbf{u}) \leq \mathbf{b}_C(t)$, by Proposition 7.5.2, $\mathbf{x}(t, \mathbf{u}) \in F_i(t, \mathbf{d})$. This means that $b_i^L(t, \mathbf{d}) \leq \widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) \leq b_i^U(t, \mathbf{d})$. It follows that $M_i(t, \mathbf{d}) = F_i(t, \mathbf{d}) \cap \{\mathbf{z} : \widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} = d_i\}$ and so $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$ (and clearly $(t, \mathbf{d}) \in D_M$).

By construction, $\mathbf{v}(\mathbf{d}) \leq \mathbf{w}(\mathbf{d})$ for any $\mathbf{d} \in \mathbb{R}^m$, and so by Proposition 7.5.2, $F_i(t, \mathbf{d})$ is nonempty. Next, since $b_i^L(t, \mathbf{d}) \leq b_i^U(t, \mathbf{d})$ for any $(t, \mathbf{d}) \in D_M$, one can analyze the three cases $d_i \leq b_i^L(t, \mathbf{d}) \leq b_i^U(t, \mathbf{d})$, $b_i^L(t, \mathbf{d}) \leq d_i \leq b_i^U(t, \mathbf{d})$, and $b_i^L(t, \mathbf{d}) \leq b_i^U(t, \mathbf{d}) \leq d_i$ to see that in each case $M_i(t, \mathbf{d})$ must be nonempty for all $(t, \mathbf{d})$. It is clear that $\mathbf{v}$ and $\mathbf{w}$ are Lipschitz continuous, and that $\mathbf{d}_{F,i}$ satisfies

$$\left\|\mathbf{d}_{F,i}(s, \mathbf{d}_1) - \mathbf{d}_{F,i}(s, \mathbf{d}_2)\right\| \leq 2\left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1,$$

for any $(s, \mathbf{d}_1)$, $(s, \mathbf{d}_2) \in D_M$. Combined with the Lipschitz continuity properties from Proposition 7.5.2, there exists $L_i > 0$ such that the lower endpoint $\mathbf{v}_{F,i}$ of F_i satisfies

$$\|\mathbf{v}_{F,i}(s, \mathbf{d}_1) - \mathbf{v}_{F,i}(s, \mathbf{d}_2)\| \le L_i \|\mathbf{d}_1 - \mathbf{d}_2\|_1, \tag{7.16}$$

for any $(s, \mathbf{d}_1)$, $(s, \mathbf{d}_2) \in D_M$, and similarly for the upper endpoint $\mathbf{w}_{F,i}$ of F_i. Since b_i^L and b_i^U are the optimal objective values of certain parametric LPs, the Lipschitz continuity of the endpoints of F_i combined with Lemma 2.4.2 establish that b_i^L and b_i^U satisfy inequalities similar to Inequality (7.16). The Lipschitz continuity of median $\{\cdot, \cdot, \cdot\}$ consequently establishes that $b_i^{med} : (t, \mathbf{d}) \mapsto \text{median}\{d_i, b_i^L(t, \mathbf{d}), b_i^U(t, \mathbf{d})\}$ also satisfies an inequality like Inequality (7.16). Writing $M_i(t, \mathbf{d})$ as

$$\left\{\mathbf{z} : -\mathbf{I}\mathbf{z} \le -\mathbf{v}_{F,i}(t, \mathbf{d}), \mathbf{I}\mathbf{z} \le \mathbf{w}_{F,i}(t, \mathbf{d}), -\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} \le -b_i^{med}(t, \mathbf{d}), \widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{z} \le b_i^{med}(t, \mathbf{d})\right\},$$

we can apply Lemma 2.4.2 again to see that Condition 2 of Assumption 7.3.2 holds. □

Note that we can change the definitions of $\mathbf{A}_{F,i}$ and $\mathbf{d}_{F,i}$ in Proposition 7.5.3 to

$$\mathbf{A}_{F,i} = \begin{bmatrix} \widehat{\mathbf{A}} \\ -\widehat{\mathbf{a}}_i^{\mathrm{T}} \\ \mathbf{A}_C \end{bmatrix}, \quad \mathbf{d}_{F,i} : (t, \mathbf{d}) \mapsto \begin{bmatrix} \mathbf{d} \\ -d_i \\ \mathbf{b}_C(t) \end{bmatrix}, \tag{7.17}$$

and the conclusion of the proposition still holds, with the proof unchanged. The original definition of M_i can be seen as an analog of the definitions in Eqn. (7) of [168] ("flattening then tightening"), while the alternate definition coming from using $\mathbf{A}_{F,i}$ and $\mathbf{d}_{F,i}$ in (7.17) can be seen as an analog of the definitions in Eqn. (6) of [168] ("tightening then flattening"). The definition of the M_i mappings in Proposition 7.5.3 as stated typically lead to better bounds, since the constraint information is used to tighten each face of the bounding polyhedron individually. However, this is *not* always the case, as an example in §7.7.1 demonstrates.

Although the definition of M_i in Proposition 7.5.3 involves the evaluation of b_i^L and b_i^U, defined by the optimal objective values of LPs, these LPs are over intervals and thus can be evaluated cheaply by inspecting the sign of each component of the objective $\widehat{\mathbf{a}}_i$. The disadvantage of this definition, compared to that in §7.5.1, is that M_i is not quite as "small." Nevertheless, a numerical method based on the definitions of this section proves to be effective and fast. The details of its implementation are discussed in §7.6, and numerical results are presented in Sections 7.7.1 and 7.7.2.

7.5.3 Simultaneous interval and affine relaxations

This section discusses a method for simultaneously calculating an interval enclosure and affine relaxations. In contrast to previous theories, the information from the interval enclosure can be used to improve the relaxations and vice versa. For example, in [170] or [176], tight relaxations cannot be used to tighten the interval enclosures.

The development is similar to that in §7.4.3, where some of the differential states are interpreted as unknown, but constant in time, parameters. The main result of this section is Proposition 7.5.6, which provides another definition of $\mathbf{A}$ and the M_i mappings which satisfy Assumption 7.3.2. The bounds $\{\mathbf{z} : \mathbf{A}(t)\mathbf{z} \leq \mathbf{b}(t)\}$ that result from Theorem 7.3.1 using the definitions of Proposition 7.5.6 imply that

$$\begin{gathered}
\mathbf{y}^L(t) \leq \mathbf{x}_y(t, \mathbf{u}, \mathbf{p}) \leq \mathbf{y}^U(t), \\
[\mathbf{a}^l_j(t)^{\mathrm{T}}]\mathbf{p} + \mathbf{b}^l(t) \leq \mathbf{x}_y(t, \mathbf{u}, \mathbf{p}) \leq [\mathbf{a}^u_j(t)^{\mathrm{T}}]\mathbf{p} + \mathbf{b}^u(t),
\end{gathered}$$

for all t, $\mathbf{p}$, and $\mathbf{u}$, where $(-\mathbf{y}^L, \mathbf{y}^U, -\mathbf{b}^l, \mathbf{b}^u) = \mathbf{b}$ and $\mathbf{x}_y$ is a solution of the IVP in parametric ODEs (7.11). As evidenced by the form of the bounds, the M_i mappings in Proposition 7.5.6 are also the intersection of an interval and an "affine" enclosure. Using the definitions in Proposition 7.5.6 and again letting $(-\mathbf{y}^L, \mathbf{y}^U, -\mathbf{b}^l, \mathbf{b}^u) = \mathbf{b}$, the differential inequalities in Hypothesis 4 of Theorem 7.3.1 become

$$\begin{aligned}
\dot{y}^L_j(t) &\leq \inf\{f_{y,j}(t, \mathbf{v}, \mathbf{y}, \mathbf{p}) : \mathbf{v} \in U(t), (\mathbf{y}, \mathbf{p}) \in M_j(t, \mathbf{b}(t))\}, \\
\dot{y}^U_j(t) &\geq \sup\{f_{y,j}(t, \mathbf{v}, \mathbf{y}, \mathbf{p}) : \mathbf{v} \in U(t), (\mathbf{y}, \mathbf{p}) \in M_{j+n_y}(t, \mathbf{b}(t))\}, \\
\dot{b}^l_j(t) &\leq \inf\{f_{y,j}(t, \mathbf{v}, \mathbf{y}, \mathbf{p}) - \dot{\mathbf{a}}^l_j(t)^{\mathrm{T}}\mathbf{p} : \mathbf{v} \in U(t), (\mathbf{y}, \mathbf{p}) \in M_{j+2n_y}(t, \mathbf{b}(t))\}, \\
\dot{b}^u_j(t) &\geq \sup\{f_{y,j}(t, \mathbf{v}, \mathbf{y}, \mathbf{p}) - \dot{\mathbf{a}}^u_j(t)^{\mathrm{T}}\mathbf{p} : \mathbf{v} \in U(t), (\mathbf{y}, \mathbf{p}) \in M_{j+3n_y}(t, \mathbf{b}(t))\},
\end{aligned}$$

for all j. When establishing that these conditions hold, a combination of interval arithmetic and some sort of "affine arithmetic" (see Ch. 3) may be used to take advantage of this form of the M_i.

Before proceeding, the interval-valued operator $\square(\cdot, \cdot)$ is introduced. In words, if $\mathbf{v} \leq \mathbf{w}$,

$\square(\mathbf{v},\mathbf{w})$ returns the nonempty interval $[\mathbf{v},\mathbf{w}]$. If $v_j > w_j$ for any j, the upper and lower bounds of the resulting interval in the j^{th} dimension equal the average value of v_j and w_j. The properties of $\square$ in Lemma 7.5.4 clearly hold.

Definition 7.5.1. For any $n \in \mathbb{N}$ and $D \subset \mathbb{R}^n$, let $\mathbb{I}D = \{[\mathbf{v},\mathbf{w}] \subset D : [\mathbf{v},\mathbf{w}] \neq \varnothing\}$.

Definition 7.5.2. For any $n \in \mathbb{N}$, the mapping $\square : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{IR}^n$ is defined by $\square(\mathbf{v},\mathbf{w}) = [\widehat{\mathbf{v}},\widehat{\mathbf{w}}]$ where $\widehat{\mathbf{v}}$, $\widehat{\mathbf{w}}$ are given componentwise by $\widehat{v}_j = \min\{v_j, (v_j + w_j)/2\}$ and $\widehat{w}_j = \max\{w_j, (v_j + w_j)/2\}$.

Lemma 7.5.4. *Let $\widehat{\mathbf{v}}$, $\widehat{\mathbf{w}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ be defined by the endpoints of $\square$; i.e.*

$$[\widehat{\mathbf{v}}(\mathbf{v},\mathbf{w}), \widehat{\mathbf{w}}(\mathbf{v},\mathbf{w})] = \square(\mathbf{v},\mathbf{w}).$$

Then there exists a $L > 0$ such that for all $(\mathbf{v},\mathbf{w})$ and $(\mathbf{v}',\mathbf{w}')$ in $\mathbb{R}^n \times \mathbb{R}^n$,

$$\left\|\widehat{\mathbf{v}}(\mathbf{v},\mathbf{w}) - \widehat{\mathbf{v}}(\mathbf{v}',\mathbf{w}')\right\| \leq L \left\|(\mathbf{v},\mathbf{w}) - (\mathbf{v}',\mathbf{w}')\right\|$$

and similarly for $\widehat{\mathbf{w}}$. Further, $\square$ is always nonempty-valued.

The definitions in Proposition 7.5.6 depend on Definition 7.5.4, which requires the following class of fundamental objects. Some necessary properties are subsequently stated and proved.

Definition 7.5.3. For $(n, q) \in \mathbb{N}^2$, let $Y \subset \mathbb{R}^n$ and $Q \subset \mathbb{R}^q$. Let $\mathbb{A}(Y, Q)$ be defined by the following: $A \in \mathbb{A}(Y, Q)$ if and only if

1. $A \subset Y \times Q$.
2. $\forall \mathbf{q} \in Q$, there exists $\mathbf{y} \in Y$ such that $(\mathbf{y},\mathbf{q}) \in A$.
3. There exist $\mathbf{v}$, $\mathbf{w}$, $\mathbf{d}^l$, $\mathbf{d}^u \in \mathbb{R}^n$, and $\mathbf{A}^l$, $\mathbf{A}^u \in \mathbb{R}^{n\times q}$ such that

$$A = \left\{(\mathbf{y},\mathbf{q}) \in [\mathbf{v},\mathbf{w}] \times Q : \mathbf{A}^l\mathbf{q} + \mathbf{d}^l \leq \mathbf{y} \leq \mathbf{A}^u\mathbf{q} + \mathbf{d}^u\right\}.$$

Definition 7.5.4. For any $(n_y, n_p, m_c) \in \mathbb{N}^3$, nonempty compact $P \subset P^I \in \mathbb{IR}^{n_p}$, $\mathbf{A}_C \in \mathbb{R}^{m_c\times(n_y+n_p)}$, and $i \in \{1,\ldots,4n_y\}$, define $A_i : \mathbb{R}^{4n_y} \times \mathbb{R}^{n_y\times n_p} \times \mathbb{R}^{n_y\times n_p} \times \mathbb{R}^{m_c} \rightrightarrows \mathbb{R}^{n_y} \times P$ by Algorithm 6.

Algorithm 6 Calculation of A_i mapping in Definition 7.5.4

```
Require: $\mathbf{d} \in \mathbb{R}^{4n_y}$, $\mathbf{A}^l = [(\mathbf{a}_j^l)^{\mathrm{T}}] \in \mathbb{R}^{n_y \times n_p}$, $\mathbf{A}^u = [(\mathbf{a}_j^u)^{\mathrm{T}}] \in \mathbb{R}^{n_y \times n_p}$, $\mathbf{b}_C \in \mathbb{R}^{m_c}$
  Set $\mathbf{y}^L$, $\mathbf{y}^U$, $\mathbf{b}^l$, $\mathbf{b}^u \in \mathbb{R}^{n_y}$ such that $(-\mathbf{y}^L, \mathbf{y}^U, -\mathbf{b}^l, \mathbf{b}^u) = \mathbf{d}$.
  Set $j = ((i-1) \mod n_y) + 1$.
  $(\mathbf{v}, \mathbf{w}) \leftarrow (\mathbf{y}^L, \mathbf{y}^U)$, $(\widehat{\mathbf{b}}^l, \widehat{\mathbf{b}}^u) \leftarrow (\mathbf{b}^l, \mathbf{b}^u)$, $(\widehat{\mathbf{A}}^l, \widehat{\mathbf{A}}^u) \leftarrow (\mathbf{A}^l, \mathbf{A}^u)$
  Flatten and Tighten:
  if $i \le n_y$ then
    (corresponds to an interval lower bound)
    $\widehat{y}_j^L \leftarrow v_j \leftarrow y_j^L$
    $\widehat{y}_j^U \leftarrow w_j \leftarrow y_j^L$
    $\widehat{b}_j^l \leftarrow \widehat{b}_j^u \leftarrow y_j^L$, $\widehat{\mathbf{a}}_j^l \leftarrow \widehat{\mathbf{a}}_j^u \leftarrow \mathbf{0}$
  else if $i \le 2n_y$ then
    (corresponds to an interval upper bound)
    $\widehat{y}_j^L \leftarrow v_j \leftarrow y_j^U$
    $\widehat{y}_j^U \leftarrow w_j \leftarrow y_j^U$
    $\widehat{b}_j^l \leftarrow \widehat{b}_j^u \leftarrow y_j^U$, $\widehat{\mathbf{a}}_j^l \leftarrow \widehat{\mathbf{a}}_j^u \leftarrow \mathbf{0}$
  else if $i \le 3n_y$ then
    (corresponds to an affine underestimator)
    $\widehat{y}_j^L \leftarrow v_j \leftarrow \min\{(\mathbf{a}_j^l)^{\mathrm{T}}\mathbf{p} : \mathbf{p} \in P\} + b_j^l$
    $\widehat{y}_j^U \leftarrow w_j \leftarrow \max\{(\mathbf{a}_j^l)^{\mathrm{T}}\mathbf{p} : \mathbf{p} \in P\} + b_j^l$
    $\widehat{b}_j^l \leftarrow \widehat{b}_j^u \leftarrow b_j^l$, $\widehat{\mathbf{a}}_j^l \leftarrow \widehat{\mathbf{a}}_j^u \leftarrow \mathbf{a}_j^l$
  else
    (corresponds to an affine overestimator)
    $\widehat{y}_j^L \leftarrow v_j \leftarrow \min\{(\mathbf{a}_j^u)^{\mathrm{T}}\mathbf{p} : \mathbf{p} \in P\} + b_j^u$
    $\widehat{y}_j^U \leftarrow w_j \leftarrow \max\{(\mathbf{a}_j^u)^{\mathrm{T}}\mathbf{p} : \mathbf{p} \in P\} + b_j^u$
    $\widehat{b}_j^l \leftarrow \widehat{b}_j^u \leftarrow b_j^u$, $\widehat{\mathbf{a}}_j^l \leftarrow \widehat{\mathbf{a}}_j^u \leftarrow \mathbf{a}_j^u$
  end if
  $[\mathbf{v}', \mathbf{w}'] \times [\mathbf{p}_{dummy}^L, \mathbf{p}_{dummy}^U] \leftarrow I_t(\square(\mathbf{v}, \mathbf{w}) \times P^I, \mathbf{b}_C; \mathbf{A}_C)$
  Rectify:
  for $k \in \{1, \dots, n_y\}$ do
    if $k \neq j$ then
      $q_k^{\text{diff}} \leftarrow \min\{(\mathbf{a}_k^u - \mathbf{a}_k^l)^{\mathrm{T}}\mathbf{p} : \mathbf{p} \in P\} + b_k^u - b_k^l$,
      $q_k^{l,\text{diff}} \leftarrow w_k' - \max\{(\mathbf{a}_k^l)^{\mathrm{T}}\mathbf{p} : \mathbf{p} \in P\} - b_k^l$,
      $q_k^{u,\text{diff}} \leftarrow \min\{(\mathbf{a}_k^u)^{\mathrm{T}}\mathbf{p} : \mathbf{p} \in P\} + b_k^u - v_k'$;
      if $i \le 2n_y$ then
        $\widehat{y}_k^L \leftarrow v_k'$,                        $\widehat{b}_k^l \leftarrow b_k^l + \min\{0, q_k^{\text{diff}}/2, q_k^{l,\text{diff}}\}$
        $\widehat{y}_k^U \leftarrow w_k'$,                        $\widehat{b}_k^u \leftarrow b_k^u - \min\{0, q_k^{\text{diff}}/2, q_k^{u,\text{diff}}\}$
      else
        $\widehat{y}_k^L \leftarrow v_k' + \min\{0, q_k^{u,\text{diff}}\}$,    $\widehat{b}_k^l \leftarrow b_k^l + \min\{0, q_k^{\text{diff}}/2\}$
        $\widehat{y}_k^U \leftarrow w_k' - \min\{0, q_k^{l,\text{diff}}\}$,    $\widehat{b}_k^u \leftarrow b_k^u - \min\{0, q_k^{\text{diff}}/2\}$
      end if
    end if
  end for
  return $A_i(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C) = \left\{(\mathbf{y}, \mathbf{p}) \in [\widehat{\mathbf{y}}^L, \widehat{\mathbf{y}}^U] \times P : \widehat{\mathbf{A}}^l\mathbf{p} + \widehat{\mathbf{b}}^l \le \mathbf{y} \le \widehat{\mathbf{A}}^u\mathbf{p} + \widehat{\mathbf{b}}^u\right\}$
```

Lemma 7.5.5. *For any $(n_y, n_p, m_c) \in \mathbb{N}^3$, nonempty compact $P \subset P^I \in \mathbb{IR}^{n_p}$, $\mathbf{A}_C \in \mathbb{R}^{m_c \times (n_y + n_p)}$, and $i \in \{1, \dots, 4n_y\}$ the following holds:*

1. *A_i defined in Definition 7.5.4 is a mapping into $\mathbb{A}(\mathbb{R}^{n_y}, P)$.*
2. *There exists $L_i > 0$ such that for any $\mathbf{A}^l \in \mathbb{R}^{n_y \times n_p}$, $\mathbf{A}^u \in \mathbb{R}^{n_y \times n_p}$, and $\mathbf{b}_C \in \mathbb{R}^{m_c}$, and for any $(\mathbf{d}_1, \mathbf{d}_2) \in \mathbb{R}^{4n_y} \times \mathbb{R}^{4n_y}$, and $(\mathbf{y}_1, \mathbf{p}_1) \in A_i(\mathbf{d}_1, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$, there exists $(\mathbf{y}_2, \mathbf{p}_2) \in A_i(\mathbf{d}_2, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$ such that $\|(\mathbf{y}_1, \mathbf{p}_1) - (\mathbf{y}_2, \mathbf{p}_2)\| \le L_i \|\mathbf{d}_1 - \mathbf{d}_2\|$.*

Proof.

1. It is clear from the return value of A_i in Algorithm 6 that Conditions 1 and 3 of Definition 7.5.3 are satisfied. The main challenge is establishing that for any $(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$ in the domain of A_i and $\mathbf{p} \in P$, there exists a $\mathbf{y} \in \mathbb{R}^{n_y}$ such that $(\mathbf{y}, \mathbf{p}) \in A_i(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$. However, this is still not too hard to see. In the j^{th} dimension (where j is defined as in Algorithm 6), it is clear that the "Flatten and Tighten" step in Algorithm 6 ensures that $[\widehat{y}_j^L, \widehat{y}_j^U]$ contains $[(\widehat{\mathbf{a}}_j^l)^{\mathrm{T}}\mathbf{p} + \widehat{b}_j^l, (\widehat{\mathbf{a}}_j^u)^{\mathrm{T}}\mathbf{p} + \widehat{b}_j^u]$ for each $\mathbf{p} \in P$. Meanwhile, before the "Rectify" step in Algorithm 6, for $k \neq j$ and fixed $\mathbf{p} \in P$, the intersection $[v_k', w_k'] \cap [(\mathbf{a}_k^l)^{\mathrm{T}}\mathbf{p} + b_k^l, (\mathbf{a}_k^u)^{\mathrm{T}}\mathbf{p} + b_k^u]$ may be empty. One can verify that the definitions of $\widehat{y}_k^L$, $\widehat{y}_k^U$, $\widehat{b}_k^l$, $\widehat{b}_k^u$ ensure that $[\widehat{y}_k^L, \widehat{y}_k^U] \cap [(\mathbf{a}_k^l)^{\mathrm{T}}\mathbf{p} + \widehat{b}_k^l, (\mathbf{a}_k^u)^{\mathrm{T}}\mathbf{p} + \widehat{b}_k^u]$ is nonempty for each $\mathbf{p} \in P$ and $k \neq j$. This establishes that Condition 2 of Definition 7.5.3 holds.
2. Choose any $\mathbf{A}^l \in \mathbb{R}^{n_y \times n_p}$, $\mathbf{A}^u \in \mathbb{R}^{n_y \times n_p}$, and $\mathbf{b}_C \in \mathbb{R}^{m_c}$. Let $\widehat{\mathbf{y}}^L$, $\widehat{\mathbf{y}}^U$, $\widehat{\mathbf{b}}^l$, $\widehat{\mathbf{b}}^u$ be mappings $\mathbb{R}^{4n_y} \to \mathbb{R}^{n_y}$ which define $A_i(\cdot, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$; i.e.

   $$A_i(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C) = \{(\mathbf{y}, \mathbf{p}) \in [\widehat{\mathbf{y}}^L(\mathbf{d}), \widehat{\mathbf{y}}^U(\mathbf{d})] \times P : \widehat{\mathbf{A}}^l\mathbf{p} + \widehat{\mathbf{b}}^l(\mathbf{d}) \le \mathbf{y} \le \widehat{\mathbf{A}}^u\mathbf{p} + \widehat{\mathbf{b}}^u(\mathbf{d})\}.$$

   We claim that there exists $\widehat{L}_i > 0$ (independent of $\mathbf{A}^l$, $\mathbf{A}^u$, $\mathbf{b}_C$) such that for all $\mathbf{d}_1$, $\mathbf{d}_2 \in \mathbb{R}^{4n_y}$, we have

   $$\left\|\widehat{\mathbf{y}}^L(\mathbf{d}_1) - \widehat{\mathbf{y}}^L(\mathbf{d}_2)\right\| \le \widehat{L}_i \|\mathbf{d}_1 - \mathbf{d}_2\|, \tag{7.18}$$

   and similarly for $\widehat{\mathbf{y}}^U$, $\widehat{\mathbf{b}}^l$, $\widehat{\mathbf{b}}^u$. This is fairly clear from inspection of the operations in Algorithm 6 and application of Lemma 7.5.4 and Proposition 7.5.2.

   Now, choose any $(\mathbf{d}_1, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$ and $(\mathbf{d}_2, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$ in the domain of A_i, and any $(\mathbf{y}_1, \mathbf{p}_1) \in A_i(\mathbf{d}_1, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$. By the first claim A_i is a mapping into $\mathbb{A}(\mathbb{R}^{n_y}, P)$. By definition, for $j \in \{1, 2\}$, $A_i(\mathbf{d}_j, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C) \cap (\mathbb{R}^{n_y} \times \{\mathbf{p}_1\})$ is nonempty, and furthermore, an interval. So by definition of the Hausdorff distance (or application of Lemma 2.4.2), we have that there exists $\mathbf{y}_2$ such that $(\mathbf{y}_2, \mathbf{p}_1) \in A_i(\mathbf{d}_2, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C)$ and

$$\begin{aligned}\|(\mathbf{y}_1, \mathbf{p}_1) - (\mathbf{y}_2, \mathbf{p}_1)\|_\infty = \|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \\ \max\Big\{ \|\widehat{\mathbf{y}}^L(\mathbf{d}_1) - \widehat{\mathbf{y}}^L(\mathbf{d}_2)\|_\infty, \|\widehat{\mathbf{y}}^U(\mathbf{d}_1) - \widehat{\mathbf{y}}^U(\mathbf{d}_2)\|_\infty, \\ \left\|\widehat{\mathbf{b}}^l(\mathbf{d}_1) - \widehat{\mathbf{b}}^l(\mathbf{d}_2)\right\|_\infty, \left\|\widehat{\mathbf{b}}^u(\mathbf{d}_1) - \widehat{\mathbf{b}}^u(\mathbf{d}_2)\right\|_\infty \Big\}.\end{aligned}$$

Combined with Inequality (7.18) (and the others for $\widehat{\mathbf{y}}^U$, $\widehat{\mathbf{b}}^l$, $\widehat{\mathbf{b}}^u$) , we have the result, applying equivalence of norms if necessary.

□

The following result forms the basis of the numerical method discussed in §7.6.2.

Proposition 7.5.6. *Given $(n_y, n_p) \in \mathbb{N}^2$, $P \subset \mathbb{R}^{n_p}$, and $P^I \in \mathbb{IR}^{n_p}$, let $n_x = n_y + n_p$, and suppose that P is nonempty and compact, and $P \subset P^I$. Given $m_c \in \mathbb{N}$, $\mathbf{A}_C \in \mathbb{R}^{m_c \times n_x}$, and $\mathbf{b}_C : T \to \mathbb{R}^{m_c}$, let $X_C : t \mapsto \{\mathbf{z} : \mathbf{A}_C \mathbf{z} \leq \mathbf{b}_C(t)\}$. For $i \in \{1, \dots, 4n_y\}$, let A_i be defined as in Definition 7.5.4 (with n_y, n_p, m_c, P, P^I, and $\mathbf{A}_C$ as given here). Suppose that for all $\mathbf{u} \in \mathcal{U}$ and for all solutions of IVP (7.1), $\mathbf{x}(\cdot, \mathbf{u}) = (\mathbf{x}_y(\cdot, \mathbf{u}), \mathbf{x}_p(\cdot, \mathbf{u}))$, satisfy $\mathbf{x}_p(t, \mathbf{u}) \in P$, for all $t \in T$. Given $(\mathbf{a}_j^l, \mathbf{a}_j^u) : T \to \mathbb{R}^{n_p} \times \mathbb{R}^{n_p}$ for $j \in \{1, \dots, n_y\}$ which are absolutely continuous, suppose $\mathbf{A}^l : t \mapsto [\mathbf{a}_j^l(t)^{\mathrm{T}}] \in \mathbb{R}^{n_y \times n_p}$, $\mathbf{A}^u : t \mapsto [\mathbf{a}_j^u(t)^{\mathrm{T}}] \in \mathbb{R}^{n_y \times n_p}$, $m = 4n_y$, and*

$$\mathbf{A} : t \mapsto \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{A}^l(t) \\ \mathbf{I} & -\mathbf{A}^u(t) \end{bmatrix}.$$

For each i, define $M_i : (t, \mathbf{d}) \mapsto A_i(\mathbf{d}, \mathbf{A}^l(t), \mathbf{A}^u(t), \mathbf{b}_C(t))$. Then $\mathbf{A}$, $D_M = T \times \mathbb{R}^m$, and M_i satisfy Assumption 7.3.2.

Proof. Begin by checking that Condition 1 of Assumption 7.3.2 holds; choose any $i \in \{1, \dots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\mathbf{A}(t)\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{x}(t, \mathbf{u}) = d_i$. Trivially, we have that $(t, \mathbf{d}) \in D_M$. Choose $(\mathbf{y}^L, \mathbf{y}^U, \mathbf{b}^l, \mathbf{b}^u)$ so that $(-\mathbf{y}^L, \mathbf{y}^U, -\mathbf{b}^l, \mathbf{b}^u) = \mathbf{d}$ (as in Algorithm 6). Then $\mathbf{x}(t, \mathbf{u}) \in \{(\mathbf{y}, \mathbf{p}) \in [\mathbf{y}^L, \mathbf{y}^U] \times P : \mathbf{A}^l(t)\mathbf{p} + \mathbf{b}^l \leq \mathbf{y} \leq \mathbf{A}^u(t)\mathbf{p} + \mathbf{b}^u\}$.

Assume, for example, that $i \le n_y$, so that the constraint $\mathbf{a}_i(t)^{\mathrm{T}}\mathbf{x}(t,\mathbf{u}) = d_i$ corresponds to $x_{y,i}(t,\mathbf{u}) = y_i^L$. At the end of the "Flatten and Tighten" step in Algorithm 6, it holds that $\mathbf{x}_y(t,\mathbf{u})$ is in $\square(\mathbf{v},\mathbf{w})$, and so $\mathbf{x}_y(t,\mathbf{u})$ is in $[\mathbf{v}',\mathbf{w}']$ (using the properties of the interval-tightening operator from Proposition 7.5.2). The rest of the steps in the algorithm only widen this interval; i.e. $\widehat{y}_k^L \le v'_k$ and $w'_k \le \widehat{y}_k^U$ for $k \neq j$. Similarly, $\widehat{b}_k^l \le b_k^l$ and $b_k^u \le \widehat{b}_k^u$, and so it is clear that

$$\mathbf{x}(t,\mathbf{u}) \in \left\{(\mathbf{y},\mathbf{p}) : \mathbf{p} \in P, \widehat{y}_k^L \le y_k \le \widehat{y}_k^U, \mathbf{a}_k^l(t)^{\mathrm{T}}\mathbf{p} + \widehat{b}_k^l \le y_k \le \mathbf{a}_k^u(t)^{\mathrm{T}}\mathbf{p} + \widehat{b}_k^u, \forall k \neq j\right\},$$

where $j = ((i-1) \mod n_y) + 1$ as in Algorithm 6. Along with the constraint $x_{y,i}(t,\mathbf{u}) = \widehat{y}_i^L = y_i^L$, this is precisely the definition of $A_i(\mathbf{d}, \mathbf{A}^l(t), \mathbf{A}^u(t), \mathbf{b}_C(t))$ which equals $M_i(t,\mathbf{d})$. Similar reasoning establishes the cases when $i > n_y$.

Finally, Condition 2 of Assumption 7.3.2 follows from Lemma 7.5.5. □

7.5.4 Connections to DAEs

To end this section on constrained ODEs, some interesting connections to DAEs are noted. Consider the semi-explicit DAE

$$\begin{aligned}\dot{\mathbf{x}}_z(t) &= \mathbf{f}_z(t, \mathbf{u}(t), \mathbf{x}_z(t), \mathbf{x}_y(t)),\\ \mathbf{0} &= \mathbf{h}(t, \mathbf{x}_z(t), \mathbf{x}_y(t)).\end{aligned}$$

Under the assumptions that $\mathbf{h}$ is sufficiently smooth and that this is index-one, the derivative of $\mathbf{h}$ with respect to the algebraic variables ($\mathbf{x}_y$) is invertible (in a neighborhood of the solution), and so the time derivative of the algebraic variables satisfies

$$\dot{\mathbf{x}}_y(t) = -\left(\frac{\partial \mathbf{h}}{\partial \mathbf{y}}(t, \mathbf{x}_z(t), \mathbf{x}_y(t))\right)^{-1}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{z}}(t, \mathbf{x}_z(t), \mathbf{x}_y(t))\dot{\mathbf{x}}_z(t) + \frac{\partial \mathbf{h}}{\partial t}(t, \mathbf{x}_z(t), \mathbf{x}_y(t))\right).$$

Writing $X_C(t) = \{(\mathbf{z},\mathbf{y}) : \mathbf{h}(t,\mathbf{z},\mathbf{y}) = \mathbf{0}\}$, we can think of an index-one DAE as a constrained ODE. Assuming constant $\mathbf{a}_i = (\mathbf{a}_{z,i}, \mathbf{a}_{y,i})$, Hypothesis 4 of Theorem 7.3.1 would look like

$$\mathbf{a}_{z,i}^{\mathrm{T}}\mathbf{f}_z(t,\mathbf{p},\mathbf{z},\mathbf{y}) - \mathbf{a}_{y,i}^{\mathrm{T}}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{y}}(t,\mathbf{z},\mathbf{y})\right)^{-1}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{z}}(t,\mathbf{z},\mathbf{y})\mathbf{f}(t,\mathbf{p},\mathbf{z},\mathbf{y}) + \frac{\partial \mathbf{h}}{\partial t}(t,\mathbf{z},\mathbf{y})\right) \le \dot{b}_i(t),$$

for all i, almost every t, and all $\mathbf{p} \in U(t)$ and $(\mathbf{z},\mathbf{y}) \in M_i(t,\mathbf{b}(t))$ (for some definition of M_i). Ensuring this holds is maybe a little tedious; the C++ automatic differentiation code FADBAD++ [180] could be overloaded with interval arithmetic to obtain interval extensions of ${}^{\partial \mathbf{h}}/_{\partial \mathbf{y}}$, ${}^{\partial \mathbf{h}}/_{\partial \mathbf{z}}$, and ${}^{\partial \mathbf{h}}/_{\partial t}$, which could be combined with an interval Newton method. Meanwhile, Ch. 6 of [166] considers many of the details of another DAE bounding method and may contain some insight on the specifics of such an approach. Using the theory in this chapter, it might be possible to use interval relaxations of $\mathbf{h}$ to tighten or refine the value of each M_i mapping based on the constraints $X_C(t) = \{(\mathbf{z},\mathbf{y}) : \mathbf{h}(t,\mathbf{z},\mathbf{y}) = \mathbf{0}\}$.

7.6 Numerical methods

This section presents the details of numerical methods based on the implementations in §7.5.2 and §7.5.3 of the general theory ("Faster polyhedral bounds" and "Simultaneous interval and affine relaxations," respectively). These implementations will be the focus of the examples in §7.7.

7.6.1 Method for faster polyhedral bounds

Given the constant-matrix-valued $\mathbf{A}(\cdot) \equiv \widehat{\mathbf{A}}$ and information $\mathbf{A}_C$ and $\mathbf{b}_C$ defining the constraints X_C, the goal is to construct a related initial value problem in ordinary differential equations whose solution, if one exists, is the mapping $\mathbf{b}$ so that $B : t \mapsto \{\mathbf{z} : \widehat{\mathbf{A}}\mathbf{z} \le \mathbf{b}(t)\}$ bounds all (constrained) solutions of IVP (7.1).

In §7.5.2, specifically Proposition 7.5.3, assumptions on the form of X_C and $\mathbf{A}$ are made and definitions of the M_i mappings are given. With this definition, the value of an M_i mapping takes the general form of an interval intersected with a hyperplane. Since $\mathbf{A}$ is assumed to be the constant matrix $\widehat{\mathbf{A}}$, the differential inequality in Hypothesis 4 of Theorem 7.3.1 becomes

$$\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{f}(t,\mathbf{p},\mathbf{z}) \le \dot{b}(t), \quad \forall(\mathbf{p},\mathbf{z}) \in U(t) \times M_i(t,\mathbf{b}(t)),$$

for all i and for almost every $t \in T$. Thus, the values of the potentially nonlinear optimization problems

$$q_i(t,\mathbf{d}) = \sup\left\{\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{f}(t,\mathbf{p},\mathbf{z}) : (\mathbf{p},\mathbf{z}) \in U(t) \times M_i(t,\mathbf{d})\right\}$$

must be estimated.

One approach is to use an affine relaxation of the objective and solve the resulting linear programming relaxation exactly. This forms the basis of the method in Ch. 6. However, as mentioned earlier, the goal of the definitions in Proposition 7.5.3 is to avoid the solution of LPs. Consequently, an affine relaxation of the objective is still used, but the resulting LP relaxation is only solved approximately, taking advantage of the specific form of the M_i mappings.

Before stating the next result, a simplifying assumption is made. In addition to assuming that U is interval-valued, it is assumed that affine relaxations of $\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{f}(t,\cdot,\cdot)$ are available. As discussed in Ch. 3, affine relaxations could be obtained from first-order Taylor models [107], subgradients of convex and concave relaxations [124], or the method in Ch. 3 (although see the discussion at the end of this section).

Assumption 7.6.1. *Let $m \in \mathbb{N}$ and $\{\widehat{\mathbf{a}}_i \in \mathbb{R}^{n_x} : i \in \{1,\dots,m\}\}$ be given. Assume that for each $i \in \{1,\dots,m\}$, there exist $\widetilde{\mathbf{c}}_i \equiv (\widetilde{\mathbf{c}}_i^u, \widetilde{\mathbf{c}}_i^x) : T \times \mathbb{I}D_x \to \mathbb{R}^{n_u} \times \mathbb{R}^{n_x}$ and $\widetilde{h}_i : T \times \mathbb{I}D_x \to \mathbb{R}$ such that for each $Z \in \mathbb{I}D_x$ and $t \in T$,*

$$\widehat{\mathbf{a}}_i^{\mathrm{T}}\mathbf{f}(t,\mathbf{p},\mathbf{z}) \le (\widetilde{\mathbf{c}}_i^u(t,Z))^{\mathrm{T}}\mathbf{p} + (\widetilde{\mathbf{c}}_i^x(t,Z))^{\mathrm{T}}\mathbf{z} + \widetilde{h}_i(t,Z),$$

for all $(\mathbf{p},\mathbf{z}) \in U(t) \times Z$.

Proposition 7.6.1. *Let Assumption 7.3.1 hold. Given $m_e \in \mathbb{N}$ and $\mathbf{A}_e = [\mathbf{a}_{e,i}^{\mathrm{T}}] \in \mathbb{R}^{m_e \times n_x}$, assume $\mathbf{a}_{e,i} \neq \mathbf{0}$ for all i. Given $m_c \in \mathbb{N}$, $\mathbf{A}_C \in \mathbb{R}^{m_c \times n_x}$, and $\mathbf{b}_C : T \to \mathbb{R}^{m_c}$, assume that $X_C : t \mapsto \{\mathbf{z} : \mathbf{A}_C\mathbf{z} \le \mathbf{b}_C(t)\}$. Define $\widehat{\mathbf{A}} = [\widehat{\mathbf{a}}_i^{\mathrm{T}}]$, b_i^L, b_i^U, and F_i as in Proposition 7.5.3. Let Assumption 7.6.1 hold for m and $\{\widehat{\mathbf{a}}_i\}$. Assume $U : T \rightrightarrows \mathbb{R}^{n_u}$ is interval-valued. Assume $\mathbf{b} : T \to \mathbb{R}^m$ is an absolutely continuous function satisfying*

1. *$X_0 \subset \{\mathbf{z} : \widehat{\mathbf{A}}\mathbf{z} \le \mathbf{b}(t_0)\}$,*
2. *for all $i \in \{1,\dots,m\}$ and for almost every $t \in T$, $F_i(t,\mathbf{b}(t)) \subset D_x$,*
3. *for almost every $t \in T$ and all $i \in \{1,\dots,m\}$*

$$\begin{aligned} \dot{b}_i(t) = {} & \widetilde{h}_i(t, F_i(t,\mathbf{b}(t)) + \alpha_i(t,\mathbf{b}(t)) b_i^m(t,\mathbf{b}(t)) + \\ & \max\left\{(\mathbf{c}_i^u(t,\mathbf{b}(t)))^{\mathrm{T}}\mathbf{p} + (\mathbf{c}_i^x(t,\mathbf{b}(t)) - \alpha_i(t,\mathbf{b}(t))\widehat{\mathbf{a}}_i)^{\mathrm{T}}\mathbf{z} : (\mathbf{p},\mathbf{z}) \in U(t) \times F_i(t,\mathbf{b}(t))\right\}, \end{aligned} \tag{7.19}$$

where

$$\begin{aligned}
\alpha_i &: (t, \mathbf{d}) \mapsto \frac{(\mathbf{c}_i^x(t, \mathbf{d}))^{\mathrm{T}} \widehat{\mathbf{a}}_i}{\|\widehat{\mathbf{a}}_i\|_2^2}, \\
(\mathbf{c}_i^u, \mathbf{c}_i^x) &: (t, \mathbf{d}) \mapsto (\widetilde{\mathbf{c}}_i^u(t, F_i(t, \mathbf{d})), \widetilde{\mathbf{c}}_i^x(t, F_i(t, \mathbf{d}))), \text{ and} \\
b_i^m &: (t, \mathbf{d}) \mapsto \operatorname{median}\{d_i, b_i^L(t, \mathbf{d}), b_i^U(t, \mathbf{d})\}.
\end{aligned}$$

Then for all $\mathbf{u} \in \mathcal{U}$ *and any solution* $\mathbf{x}(\cdot, \mathbf{u})$ *of IVP* (7.1), $\widehat{\mathbf{A}}\mathbf{x}(t, \mathbf{u}) \le \mathbf{b}(t)$, *for all* $t \in T$.

Proof. The goal is to construct the bounds $B : t \mapsto \{\mathbf{z} : \mathbf{A}(t)\mathbf{z} \le \mathbf{b}(t)\}$ and establish that all the assumptions and hypotheses of Theorem 7.3.1 are satisfied, specifically using the definitions of $\mathbf{A}$, D_M and M_i from Proposition 7.5.3.

First, since $D_M = T \times \mathbb{R}^m$, we have $(t, \mathbf{b}(t)) \in D_M$ for all $t \in T$. Also, since $M_i(t, \mathbf{d}) \subset F_i(t, \mathbf{d})$ for all $(t, \mathbf{d})$ and $F_i(t, \mathbf{b}(t)) \subset D_x$ for almost every t, we also have $M_i(t, \mathbf{b}(t)) \subset D_x$ for all i and almost every t. Further, we clearly have $X_0 \subset B(t_0)$.

The final step is to establish that the differential inequality in Hypothesis 4 of Theorem 7.3.1 holds. As mentioned earlier, this reduces to establishing that

$$\dot{b}_i(t) \ge q_i(t, \mathbf{b}(t)) = \sup\left\{\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z}) : (\mathbf{p}, \mathbf{z}) \in U(t) \times M_i(t, \mathbf{b}(t))\right\} \tag{7.20}$$

for all i and almost every t. Since $F_i(t, \mathbf{b}(t))$ is an interval subset of D_x for almost every t, we have $F_i(t, \mathbf{b}(t)) \in \mathbb{I}D_x$. Thus the values of $\mathbf{c}_i^x$ and $\mathbf{c}_i^u$ are defined at $(t, \mathbf{b}(t))$. In addition, the maximum in the right-hand side of Eqn. (7.19) is indeed achieved, since $U(t) \times F_i(t, \mathbf{b}(t))$ is compact and the objective of the maximization in Eqn. (7.19) is linear and thus continuous. Overall the right-hand side of Eqn. (7.19) is well defined for almost every $t \in T$.

Note that for any $\widetilde{\alpha} \in \mathbb{R}$, we can rewrite q_i defined in Eqn. (7.20) as

$$\begin{aligned}
q_i(t, \mathbf{b}(t)) &= \max\left\{\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z}) - \widetilde{\alpha}\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{z} + \widetilde{\alpha}\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{z} : (\mathbf{p}, \mathbf{z}) \in U(t) \times M_i(t, \mathbf{b}(t))\right\} \\
&= \max\left\{\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z}) - \widetilde{\alpha}\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{z} : (\mathbf{p}, \mathbf{z}) \in U(t) \times M_i(t, \mathbf{b}(t))\right\} + \widetilde{\alpha} b_i^m(t, \mathbf{b}(t)),
\end{aligned}$$

where the second equality follows from the fact that $\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{z} = b_i^m(t, \mathbf{d})$ for all $\mathbf{z} \in M_i(t, \mathbf{d})$. If follows that we can relax this maximization problem to get the inequality

$$q_i(t, \mathbf{b}(t)) \le \max\left\{\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{f}(t, \mathbf{p}, \mathbf{z}) - \widetilde{\alpha}\widehat{\mathbf{a}}_i^{\mathrm{T}} \mathbf{z} : (\mathbf{p}, \mathbf{z}) \in U(t) \times F_i(t, \mathbf{b}(t))\right\} + \widetilde{\alpha} b_i^m(t, \mathbf{b}(t)).$$

Finally, using the affine relaxations of $\widetilde{\mathbf{a}}_i^{\mathrm{T}}\mathbf{f}(t,\cdot,\cdot)$ from Assumption 7.6.1, we have $\dot{b}_i(t) \geq q_i(t, \mathbf{b}(t))$ for all i and *a.e.* t, since the specific choice of $\widetilde{\alpha}$ at each t does not matter. The result follows from Theorem 7.3.1. □

As a brief note, Proposition 7.6.1 above holds for any choice of α_i. For instance, the claim still holds if each α_i was defined as identically zero instead. The specific form used aims to take advantage of the fact that the set $M_i(t, \mathbf{d})$ is a subset of an affine subspace. For concreteness, consider the following optimization problem:

$$\max\{\mathbf{c}^{\mathrm{T}}\mathbf{z} : \mathbf{z} \in Z, \mathbf{a}^{\mathrm{T}}\mathbf{z} = d\}.$$

If $\mathbf{c} = \alpha\mathbf{a}$ and assuming the feasible set is nonempty, the optimal objective value is clearly αd. In general, if one has the freedom to choose α so that $\|\mathbf{c} - \alpha\mathbf{a}\|$ is small, then the hope is that

$$\max\{(\mathbf{c} - \alpha\mathbf{a})^{\mathrm{T}}\mathbf{z} : \mathbf{z} \in Z\} + \alpha d$$

is a good approximation to the value of the original optimization problem. If $\mathbf{a} \neq \mathbf{0}$, one can then confirm that

$$\frac{\mathbf{c}^{\mathrm{T}}\mathbf{a}}{\|\mathbf{a}\|_2^2} \in \arg\min\left\{\|\mathbf{c} - \alpha\mathbf{a}\|_2^2 : \alpha \in \mathbb{R}\right\},$$

which is precisely the form used in Proposition 7.6.1.

As noted earlier, the goal of this implementation is to avoid solving linear programs. Although a linear program appears in the right-hand side of the differential equation (7.19), its feasible set is an interval and the optimal objective value can be evaluated by inspecting the signs of the components of the objective vectors.

Finally, to ensure that some $\mathbf{b}$ exists which satisfies the differential equation (7.19), and more importantly that a numerical approximation can be calculated with standard numerical integration algorithms, some regularity conditions on $\widetilde{\mathbf{c}}_i^x$, $\widetilde{\mathbf{c}}_i^u$ and $\widetilde{h}_i$ are required. As indicated in Sections 5.4.1 and 6.4.2, the specific method in Ch. 3 ensures these regularity properties.

7.6.2 Method for simultaneous interval and affine relaxations

The focus of this section are the specifics of a numerical method for constructing interval and affine relaxations by the theory in §7.5.3. For simplicity, only parametric dependence will be considered; i.e. the interval and affine relaxations are for the solutions of the constrained initial value problem in parametric ODEs

$$\dot{\mathbf{x}}_y(t,\mathbf{p}) = \mathbf{f}_y(t,\mathbf{x}_y(t,\mathbf{p}),\mathbf{p}), \quad a.e.\ t \in T, \tag{7.21a}$$

$$(\mathbf{x}_y(t_0,\mathbf{p}),\mathbf{p}) \in X_0, \tag{7.21b}$$

$$(\mathbf{x}_y(t,\mathbf{p}),\mathbf{p}) \in X_C(t), \quad \forall t \in T, \tag{7.21c}$$

for $\mathbf{p} \in P$. The main benefit of affine relaxations is the potential for better-than-first order convergence rate; including dependence on (unparameterized controls) would tend to reduce this convergence order. Consequently, in practice, one would only consider using this method if there was not any control dependence.

As in the previous section, the following assumption specifies the exact situation and provides a way to estimate the optimization problems that appear in Hypothesis 4 of Theorem 7.3.1.

Assumption 7.6.2.

1. *Assume that* $(n_y,n_p) \in \mathbb{N}^2$, $T = [t_0,t_f] \subset \mathbb{R}$, $D_y \subset \mathbb{R}^{n_y}$, $P \in \mathbb{IR}^{n_p}$, $X_0 \subset D_y \times P$, *and* $\mathbf{f}_y : T \times D_y \times P \to \mathbb{R}^{n_y}$ *are given such that for any* $(\mathbf{y},\mathbf{p}) \in D_y \times P$, *there exists a neighborhood* $N(\mathbf{y},\mathbf{p})$ *and* $\alpha \in L^1(T)$ *such that for almost every* $t \in T$

$$\|\mathbf{f}_y(t,\mathbf{y}_1,\mathbf{p}_1) - \mathbf{f}_y(t,\mathbf{y}_2,\mathbf{p}_2)\| \le \alpha(t)\,\|(\mathbf{y}_1,\mathbf{p}_1) - (\mathbf{y}_2,\mathbf{p}_2)\|\,,$$

   for every $(\mathbf{y}_1,\mathbf{p}_1)$, $(\mathbf{y}_2,\mathbf{p}_2) \in N(\mathbf{y},\mathbf{p}) \cap D_y \times P$.
2. *Given* $m_c \in \mathbb{N}$, $\mathbf{A}_C \in \mathbb{R}^{m_c \times (n_y+n_p)}$, *and* $\mathbf{b}_C : T \to \mathbb{R}^{m_c}$, *assume that* $X_C : t \mapsto \{\mathbf{z} : \mathbf{A}_C\mathbf{z} \le \mathbf{b}_C(t)\}$.
3. *Assume that for* $\mathbf{p}_r = \mathrm{mid}(P)$, *there exists* $\mathbf{y}_r : T \to \mathbb{R}^{n_y}$ *satisfying* $(\mathbf{y}_r(t_0),\mathbf{p}_r) \in X_0$, $\dot{\mathbf{y}}_r(t) = \mathbf{f}_y(t,\mathbf{y}_r(t),\mathbf{p}_r)$ *for a.e.* $t \in T$, *and* $\frac{\partial \mathbf{f}_y}{\partial \mathbf{y}}(t,\mathbf{y}_r(t),\mathbf{p}_r)$ *and* $\frac{\partial \mathbf{f}_y}{\partial \mathbf{p}}(t,\mathbf{y}_r(t),\mathbf{p}_r)$ *exist a.e.* $t \in T$.
4. *Assume that for each* $j \in \{1,\dots,n_y\}$, *there exist* $f^L_{y,j}$, $f^U_{y,j} : T \times \mathbb{A}(D_y,P) \to \mathbb{R}$, $f^{bl}_{y,j}$, $f^{bu}_{y,j} : T \times \mathbb{A}(D_y,P) \to \mathbb{R}$, *and* $\mathbf{f}^{al}_{y,j}$, $\mathbf{f}^{au}_{y,j} : T \times \mathbb{A}(D_y,P) \to \mathbb{R}^{n_p}$ *and such that for each*

$A \in \mathbb{A}(D_y, P)$ and $t \in T$

$$f^L_{y,j}(t, A) \leq f_{y,j}(t, \mathbf{y}, \mathbf{p}) \leq f^U_{y,j}(t, A),$$
$$(\mathbf{f}^{al}_{y,j}(t, A))^{\mathrm{T}}\mathbf{p} + f^{bl}_{y,j}(t, A) \leq f_{y,j}(t, \mathbf{y}, \mathbf{p}) \leq (\mathbf{f}^{au}_{y,j}(t, A))^{\mathrm{T}}\mathbf{p} + f^{bu}_{y,j}(t, A)$$

for all $(\mathbf{y}, \mathbf{p}) \in A$.

5. *For $i \in \{1, \dots, 4n_y\}$, let A_i be defined as in Definition 7.5.4 (with n_y, n_p, m_c, P, $P^I = P$, and $\mathbf{A}_C$ as assumed here).*
6. *For $j \in \{1, \dots, n_y\}$, assume A^L_j, A^U_j, A^l_j, A^u_j defined by*

$$A^L_j : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u) \mapsto A_j(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C(t)),$$
$$A^U_j : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u) \mapsto A_{j+n_y}(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C(t)),$$
$$A^l_j : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u) \mapsto A_{j+2n_y}(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C(t)),$$
$$A^u_j : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u) \mapsto A_{j+3n_y}(\mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, \mathbf{b}_C(t)),$$

are mappings $T \times \mathbb{R}^{4n_y} \times \mathbb{R}^{n_y \times n_p} \times \mathbb{R}^{n_y \times n_p} \to \mathbb{A}(D_y, P)$.

7. *Let $\mathbf{q}^L$, $\mathbf{q}^U$ be mappings $T \times \mathbb{R}^{4n_y} \times \mathbb{R}^{n_y \times n_p} \times \mathbb{R}^{n_y \times n_p} \to \mathbb{R}^{n_y}$ and let $\mathbf{q}^l$, $\mathbf{q}^u$ be mappings $T \times \mathbb{R}^{4n_y} \times \mathbb{R}^{n_y \times n_p} \times \mathbb{R}^{n_y \times n_p} \times \mathbb{R}^{n_y \times n_p} \to \mathbb{R}^{n_y}$ defined for each $j \in \{1, \dots, n_y\}$ by*

$$\begin{aligned}
&q^L_j : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u) \mapsto \\
&\quad \max\Big\{f^L_{y,j}(t, A^L_j(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)), \\
&\qquad \min\big\{\big(\mathbf{f}^{al}_{y,j}(t, A^L_j(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u))\big)^{\mathrm{T}}\mathbf{p} + f^{bl}_{y,j}(t, A^L_j(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) : \mathbf{p} \in P\big\}\Big\}, \\
&q^U_j : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u) \mapsto \\
&\quad \min\Big\{f^U_{y,j}(t, A^U_j(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)), \\
&\qquad \max\big\{\big(\mathbf{f}^{au}_{y,j}(t, A^U_j(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u))\big)^{\mathrm{T}}\mathbf{p} + f^{bu}_{y,j}(t, A^U_j(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) : \mathbf{p} \in P\big\}\Big\},
\end{aligned}$$

$$\begin{aligned}
&q_j^l : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, [\mathbf{s}_k^{\mathrm{T}}]) \mapsto \\
&\quad \max\Big\{ \min\big\{ -\mathbf{s}_j^{\mathrm{T}}\mathbf{p} + f_{y,j}^{L}(t, A_j^l(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) : \mathbf{p} \in P\big\}, \\
&\qquad \min\big\{ \big(\mathbf{f}_{y,j}^{al}(t, A_j^l(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) - \mathbf{s}_j\big)^{\mathrm{T}}\mathbf{p} + f_{y,j}^{bl}(t, A_j^l(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) : \mathbf{p} \in P\big\}\Big\}, \\
&q_j^u : (t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u, [\mathbf{s}_k^{\mathrm{T}}]) \mapsto \\
&\quad \min\Big\{ \max\big\{ -\mathbf{s}_j^{\mathrm{T}}\mathbf{p} + f_{y,j}^{U}(t, A_j^u(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) : \mathbf{p} \in P\big\}, \\
&\qquad \max\big\{ \big(\mathbf{f}_{y,j}^{au}(t, A_j^u(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) - \mathbf{s}_j\big)^{\mathrm{T}}\mathbf{p} + f_{y,j}^{bu}(t, A_j^u(t, \mathbf{d}, \mathbf{A}^l, \mathbf{A}^u)) : \mathbf{p} \in P\big\}\Big\}.
\end{aligned}$$

Proposition 7.6.2. *Let Assumption 7.6.2 hold. Let $\mathbf{a}_j^l$, $\mathbf{a}_j^u$ for $j \in \{1, \dots, n_y\}$ be absolutely continuous mappings $T \to \mathbb{R}^{n_p}$, and $\mathbf{y}^L$, $\mathbf{y}^U$, $\mathbf{b}^l$, $\mathbf{b}^u$ be absolutely continuous mappings $T \to \mathbb{R}^{n_y}$. Let $(\mathbf{A}^l, \mathbf{A}^u) : t \mapsto ([\mathbf{a}_j^l(t)^{\mathrm{T}}], [\mathbf{a}_j^u(t)^{\mathrm{T}}]) \in \mathbb{R}^{n_y \times n_p} \times \mathbb{R}^{n_y \times n_p}$ and $\mathbf{b} : t \mapsto (-\mathbf{y}^L(t), \mathbf{y}^U(t), -\mathbf{b}^l(t), \mathbf{b}^u(t))$. Assume*

$$X_0 \subset \{(\mathbf{y}, \mathbf{p}) \in [\mathbf{y}^L(t_0), \mathbf{y}^U(t_0)] \times P : \mathbf{A}^l(t_0)\mathbf{p} + \mathbf{b}^l(t_0) \leq \mathbf{y} \leq \mathbf{A}^u(t_0)\mathbf{p} + \mathbf{b}^u(t_0)\},$$

and for almost every $t \in T$

$$\begin{aligned}
\dot{\mathbf{A}}^l(t) &= \frac{\partial \mathbf{f}_y}{\partial \mathbf{y}}(t, \mathbf{y}_r(t), \mathbf{p}_r)\mathbf{A}^l(t) + \frac{\partial \mathbf{f}_y}{\partial \mathbf{p}}(t, \mathbf{y}_r(t), \mathbf{p}_r), \\
\dot{\mathbf{A}}^u(t) &= \frac{\partial \mathbf{f}_y}{\partial \mathbf{y}}(t, \mathbf{y}_r(t), \mathbf{p}_r)\mathbf{A}^u(t) + \frac{\partial \mathbf{f}_y}{\partial \mathbf{p}}(t, \mathbf{y}_r(t), \mathbf{p}_r), \\
\dot{\mathbf{y}}^L(t) &= \mathbf{q}^L(t, \mathbf{b}(t), \mathbf{A}^l(t), \mathbf{A}^u(t)), \\
\dot{\mathbf{y}}^U(t) &= \mathbf{q}^U(t, \mathbf{b}(t), \mathbf{A}^l(t), \mathbf{A}^u(t)), \\
\dot{\mathbf{b}}^l(t) &= \mathbf{q}^l(t, \mathbf{b}(t), \mathbf{A}^l(t), \mathbf{A}^u(t), \dot{\mathbf{A}}^l(t)), \\
\dot{\mathbf{b}}^u(t) &= \mathbf{q}^u(t, \mathbf{b}(t), \mathbf{A}^l(t), \mathbf{A}^u(t), \dot{\mathbf{A}}^u(t)).
\end{aligned}$$

Then for all $t \in T$

$$\begin{gathered}
\mathbf{y}^L(t) \leq \mathbf{x}_y(t, \mathbf{p}) \leq \mathbf{y}^U(t), \\
\mathbf{A}^l(t)\mathbf{p} + \mathbf{b}^l(t) \leq \mathbf{x}_y(t, \mathbf{p}) \leq \mathbf{A}^u(t)\mathbf{p} + \mathbf{b}^u(t),
\end{gathered}$$

for any $\mathbf{p} \in P$ and solution $\mathbf{x}_y(\cdot, \mathbf{p})$ of IVP (7.21).

Proof. Under Assumption 7.6.2, for any $\mathbf{p} \in P$, a solution $\mathbf{x}_y(\cdot, \mathbf{p})$ of (7.21) corresponds to a solution of (7.1) by letting $n_x = n_y + n_p$, $D_x = D_y \times P$, and $\mathbf{f} : (t, \mathbf{v}, \mathbf{y}, \mathbf{p}) \mapsto (\mathbf{f}_y(t, \mathbf{y}, \mathbf{p}), \mathbf{0})$. Further, Assumption 7.3.1 holds for this $\mathbf{f}$. Then for any $\mathbf{u} \in \mathcal{U}$, take $\mathbf{x}(\cdot, \mathbf{u}) = (\mathbf{x}_y(\cdot, \mathbf{p}), \mathbf{p})$. The goal, then, is to apply Theorem 7.3.1. All the hypotheses of Proposition 7.5.6 hold, so define $\mathbf{A}$ and M_i as in that result, using the values of $\mathbf{A}^l$, $\mathbf{A}^u$ assumed to exist in the present hypotheses. Thus, the bounds $B : t \mapsto \{\mathbf{z} : \mathbf{A}(t)\mathbf{z} \leq \mathbf{b}(t)\}$ for the original IVP (7.1) will yield the interval and affine relaxations at the conclusion of this proposition. It is not too hard to check that all the hypotheses of Theorem 7.3.1 hold. As usual, the main challenge is verifying Hypothesis 4. Consider, for instance, the definitions of q^u_j, A^u_j, and the properties of the relaxations $f^U_{y,j}$, $\mathbf{f}^{au}_{y,j}$, and $f^{bu}_{y,j}$. Then we have

$$f_{y,j}(t, \mathbf{y}, \mathbf{p}) - (\dot{\mathbf{a}}^u_j(t))^{\mathrm{T}}\mathbf{p} \leq q^u_j(t, \mathbf{b}(t), \mathbf{A}^l(t), \mathbf{A}^u(t), \dot{\mathbf{A}}^u(t)) = \dot{b}^u_j(t) = \dot{b}_{j+3n_y}(t),$$

for all $(\mathbf{y}, \mathbf{p}) \in M_{j+3n_y}(t, \mathbf{b}(t)) = A^u_j(t, \mathbf{b}(t), \mathbf{A}^l(t), \mathbf{A}^u(t))$. Similar logic establishes this for the other cases and the result follows. □

7.7 Examples

The performance of implementations of the numerical methods discussed in §7.6 is considered. At the hearts of these numerical methods are the initial value problems in Propositions 7.6.1 and 7.6.2. The implementations of these methods are C/C++ codes employing the CVODE component of the SUNDIALS suite [78] to solve these initial value problems. All numerical studies were performed on a 64-bit Linux virtual machine allocated a single core of a 3.07 GHz Intel Xeon processor and 1.28 GB RAM. Computational times are for GCC with the -O3 optimization flag.

7.7.1 State estimation with continuous-time measurements

This section considers a problem in which state constraint information greatly improves the constructed bounds. The specific bounding method used is the "faster polyhedral bounds" considered in Sections 7.5.2 and 7.6.1.

Consider the problem of state estimation: the goal is to determine or estimate the internal state of a real system. While only certain states or functions of the states can be measured directly, a mathematical model of the system is available. When the system is a dynamic one, a history of measurements is typically available. Estimating the state using a dynamic model and measurements goes back to the Kalman filter [90], where a more statistical estimate of the state is obtained. More recently, work has focused on estimates in the form of guaranteed bounds on the state in the presence of bounded uncertainties and measurement errors [48, 88, 119, 129, 199].

Consider now the specific case that the system is dynamic, the mathematical model is an IVP in ODEs, and a continuous history of measurements of the system (or more realistically, a history of measurements so frequent that interpolation is acceptable) is available, beginning at time t_0 up to the time t_f at which the state estimate is desired. Denote the state variables at a given time by $\mathbf{x}(t) \in \mathbb{R}^{n_x}$. Assume that the measurements are of some function $\mathbf{y} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ of the states, and that they have bounded error. Thus, for each $t \in T = [t_0, t_f]$ the measurements imply $\mathbf{y}(\mathbf{x}(t)) \in [\mathbf{y}^L(t), \mathbf{y}^U(t)]$, for some $\mathbf{y}^L, \mathbf{y}^U : T \to \mathbb{R}^{n_y}$. Furthermore, there may be uncertainty in inputs, initial conditions, and model parameters. The approach to state estimation taken here will be to use the mathematical model of the system to calculate guaranteed upper and lower bounds on the states at the current time t_f that are consistent with the measurements. In other words, bounds on the solutions of a constrained IVP in ODEs (such as (7.1)) are sought.

As a specific example, the bioreactor system from [128] is considered. The dynamic model describes the evolution in time of the concentrations of biomass and feed substrate. The dynamic equations on the time domain $T = [0, 20]$ (day) are

$$\begin{aligned}
\dot{x}(t) &= \left(\mu_0 \frac{s(t)}{s(t) + k_s + s(t)^2/k_i} - \alpha D(t)\right) x(t), && x(0) \in [0, 10] \text{ (mmol/L)},\\
\dot{s}(t) &= -k\mu_0 x(t) \frac{s(t)}{s(t) + k_s + s(t)^2/k_i} + D(t)(s_{in}(t) - s(t)), && s(0) \in [0, 100] \text{ (mmol/L)},
\end{aligned}$$

where $x(t)$ and $s(t)$ are the biomass and substrate concentrations, respectively, at time t, and

$$\begin{aligned}
&\widehat{s}_{in} : t \mapsto 50 + 15\cos(t/5) \text{ (mmol/L)},\\
&\mu_0 \in [0.703, 0.777] \text{ (day}^{-1}\text{)},\\
&s_{in}(t) \in [0.95\widehat{s}_{in}(t), 1.05\widehat{s}_{in}(t)], \quad \forall t \in T,\\
&(k_s, k_i) = (9.28, 256) \text{ (mmol/L)},\\
&(k, \alpha) = (42.14, 0.5),\\
&D(t) = \begin{cases} 2 \text{ (day}^{-1}\text{)}, & \text{if } t \in [0,5],\\ 0.5 \text{ (day}^{-1}\text{)}, & \text{if } t \in (5,10],\\ 1.067 \text{ (day}^{-1}\text{)}, & \text{if } t \in (10,20]. \end{cases}
\end{aligned}$$

In this case we take $U : t \mapsto [0.703, 0.777] \times [0.95\widehat{s}_{in}(t), 1.05\widehat{s}_{in}(t)]$ (although μ_0 is an uncertain constant, we treat it as a time-varying input). In addition, it is assumed that the biomass concentration is continuously measured, and so the measurements/constraints $x(t) \in [y^L(t), y^U(t)]$, for all $t \in T$ are available; for this example, y^L and y^U are obtained from plus/minus 5% deviation of the biomass concentration of a nominal trajectory $(\widehat{x}, \widehat{s})$, with $\widehat{x}(0) = 5$, $\widehat{s}(0) = 40$, $\mu_0 = 0.74$, and $s_{in}(t) = \widehat{s}_{in}(t)$ (i.e. $y^L = 0.95\widehat{x}$ and $y^U = 1.05\widehat{x}$). Since concentrations cannot be negative, nonnegativity of the constraints is also included. Then $X_C : t \mapsto \{(x,s) : x \in [y^L(t), y^U(t)], x \geq 0, s \geq 0\}$.

The method from §7.6.1 is used to calculate interval bounds on the substrate concentration at the current time. Although bounds on the biomass concentration are also obtained, the measurements/constraints are already available and tighter. Since interval bounds only are propagated (and not a more general polyhedral enclosure), the method is similar to the one described in [168]. However, as stated, the theory in [168] cannot handle the type of constraint information considered in this example.

The results are seen in Fig. 7-1. The interval estimate of the substrate concentration is $s(t_f = 20) \in [20.6, 26.5]$ (mmol/L), which indeed encloses the value of the nominal trajectory $\widehat{s}$. This is quite good for a guaranteed estimate, considering the initial uncertainties. These results are of comparable quality to the interval observer-based methods from [119, 128]. For comparison, bounds were constructed using the same method, but ignoring the measurement/constraint information; that is, $X_C : t \mapsto \{(x,s) : x \geq 0, s \geq 0\}$, with which the

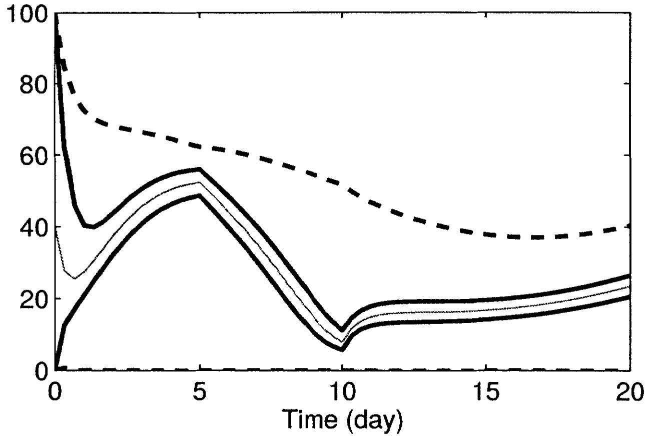


Figure 7-1: Upper and lower bounds on substrate concentration versus time for the bioreactor state estimation problem. Results of the method from §7.6.1 using state measurements as constraints are thick black lines, while the results of the same method ignoring the measurement information are dashed lines. The nominal value ($\widehat{s}$) is plotted with a thin solid line.

raw bounds (the solution of the IVP (7.19)) are intersected at each point in time. Even with this post-processing intersection, the bounds are clearly inferior; the estimate of the substrate concentration at $t_f = 20$ is $[0, 40.4]$ (mmol/L).

For this example, the method from §7.6.1 uses the Backwards Differentiation Formulae (BDF) implementation in CVODE, using a Newton iteration for the corrector, with relative and absolute integration tolerances equal to 10^{-6}. With these integration parameters, the method requires 0.001s to compute the estimate.

As mentioned in §7.5.2, there are different options for defining the mappings M_i; specifically, the definition in Proposition 7.5.3 could be modified to use $\mathbf{A}_{F,i}$ and $\mathbf{d}_{F,i}$ in Eqn. (7.17). The result of this is seen in Fig. 7-2. It is interesting to note that one method is not always better than the other for the biomass concentration (however, the substrate concentration bounds are the same). To get an idea of why this is the case, look at the dynamics of the biomass concentration, which can be written in the general form

$$\dot{x}(t) = g(t, \mu_0, s(t))x(t).$$

If $g(t, \mu_0, s(t)) \geq 0$, then x grows more slowly when $x(t)$ is small than when it is large. But if $g(t, \mu_0, s(t)) < 0$, then x grows more slowly when $x(t)$ is large than when it is small.

Consequently, let $\mathbf{d} = (-x^L, -s^L, x^U, s^U)$, with $[x^L, x^U] \supset [y^L(t), y^U(t)]$ for some $t \in T$

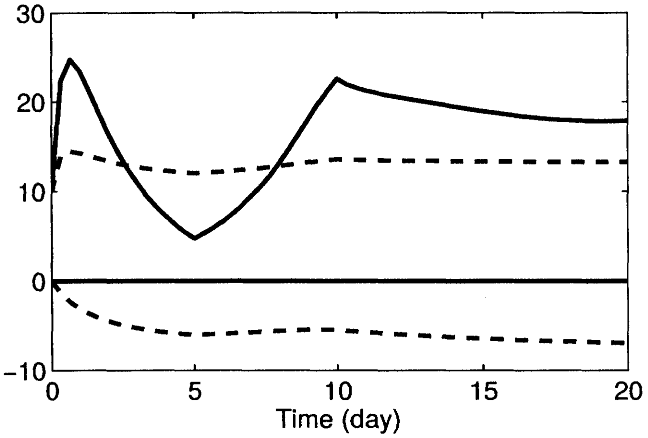


Figure 7-2: Upper and lower bounds on biomass concentration versus time for the bioreactor state estimation problem. Results of the method from §7.6.1 as stated are thick black lines, while the results of the modified method are dotted black lines.

and $s^L \geq 0$. Consider what happens when one constructs the set $M_3(t, \mathbf{d})$ corresponding to the upper bound of x. If one "flattens then tightens" (as in Proposition 7.5.3 as stated), $M_3(t, \mathbf{d})$ will have the form $[x^U, x^U] \times [s^L, s^U]$. Meanwhile, using the alternate definition ("tighten then flatten"), $M_3(t, \mathbf{d})$ will have the form $[y^U(t), y^U(t)] \times [s^L, s^U]$. Letting

$$\begin{aligned}
a(t, \mathbf{d}) &= \sup\left\{g(t, \mu_0, r)z : \mu_0 \in [0.703, 0.777], (z, r) \in [x^U, x^U] \times [s^L, s^U]\right\},\\
b(t, \mathbf{d}) &= \sup\left\{g(t, \mu_0, r)z : \mu_0 \in [0.703, 0.777], (z, r) \in [y^U(t), y^U(t)] \times [s^L, s^U]\right\},
\end{aligned}$$

then as noted above, one may have $a(t, \mathbf{d}) \geq b(t, \mathbf{d})$ or $a(t, \mathbf{d}) \leq b(t, \mathbf{d})$ at different points in time. As a result, one set of bounds is not always better than the other. This observation hints at an improved method; however, the method based on Proposition 7.5.3 as stated proves to be effective, as demonstrated by this example and the next.

7.7.2 Faster polyhedral bounds

The model of a stirred-tank reactor from §6.5.2 is considered to demonstrate the effectiveness of the "faster polyhedral bounds" considered in Sections 7.5.2 and 7.6.1. In particular, the faster polyhedral bounds are compared with the "fast polyhedral bounds" from §7.5.1, originally presented in Ch. 6.

The specific equations describing the evolution of the concentrations of chemical species

A, B, C, and D (denoted $\mathbf{x} = (x_{\mathrm{A}}, x_{\mathrm{B}}, x_{\mathrm{C}}, x_{\mathrm{D}})$) are

$$\dot{\mathbf{x}}(t,\mathbf{u}) = \begin{bmatrix} -1 & -1 \\ -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_3(t)x_{\mathrm{A}}(t,\mathbf{u})x_{\mathrm{B}}(t,\mathbf{u}) \\ k_2 x_{\mathrm{A}}(t,\mathbf{u})x_{\mathrm{C}}(t,\mathbf{u}) \end{bmatrix} + \begin{bmatrix} ({}^1\!/_V)(u_1(t)v_{\mathrm{A}} - x_{\mathrm{A}}(t,\mathbf{u})(v_{\mathrm{A}} + v_{\mathrm{B}})) \\ ({}^1\!/_V)(u_2(t)v_{\mathrm{B}} - x_{\mathrm{B}}(t,\mathbf{u})(v_{\mathrm{A}} + v_{\mathrm{B}})) \\ ({}^1\!/_V)(-x_{\mathrm{C}}(t,\mathbf{u})(v_{\mathrm{A}} + v_{\mathrm{B}})) \\ ({}^1\!/_V)(-x_{\mathrm{D}}(t,\mathbf{u})(v_{\mathrm{A}} + v_{\mathrm{B}})) \end{bmatrix}. \tag{7.22}$$

The known parameters are $V = 20$ (L), $k_2 = 0.4$ ($\mathrm{M}^{-1}\mathrm{min}^{-1}$), $v_{\mathrm{A}} = v_{\mathrm{B}} = 1$ ($\mathrm{L(min)}^{-1}$). The time-varying uncertainties are the inlet concentration of species A, $u_1(t) \in [0.9, 1.1]$ (M), the inlet concentration of species B, $u_2(t) \in [0.8, 1.0]$ (M), and the rate constant of the first reaction, $u_3(t) \in [10, 50]$ ($\mathrm{M}^{-1}\mathrm{min}^{-1}$). Initially, the concentration of each species is zero, and at $t = 0$, A and B begin to flow in. The time period of interest is $T = [0, 10]$ (min).

The bounds are given by a constant matrix

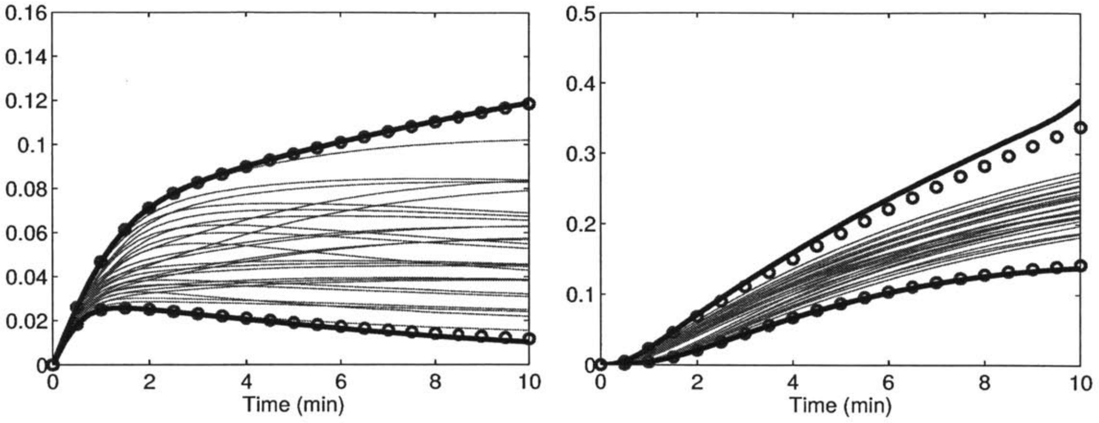
$$\mathbf{A} : t \mapsto \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \\ -\mathbf{D}^+ \\ \mathbf{D}^+ \\ -\mathbf{N} \\ \mathbf{N} \end{bmatrix},$$

where

$$\mathbf{D}^+ = \begin{bmatrix} -{}^1\!/_3 & -{}^1\!/_3 & {}^1\!/_3 & 0 \\ -{}^1\!/_3 & 0 & -{}^1\!/_3 & {}^1\!/_3 \end{bmatrix} \text{ and } \mathbf{N} = \begin{bmatrix} -1 & 2 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix}.$$

The merits of this form for the bounds are discussed in §6.5.2. Meanwhile, constraint information in the form of nonnegativity of the states is used ($X_C : t \mapsto \{\mathbf{z} : \mathbf{z} \geq \mathbf{0}\}$).

The results for two representative species are plotted in Fig. 7-3. As one can see, the bounds resulting from the numerical method from §7.6.1 are of comparable quality compared to the method from Ch. 6. Both methods use the same affine relaxations from Ch. 3 to overestimate the dynamics. Both methods employ the BDF implementation in CVODE, using a Newton iteration for the corrector, with relative and absolute integration tolerances equal to 10^{-6}. The method from Ch. 6, which must solve linear programs, requires 0.0275s, while the method from §7.6.1 requires 0.0055s, or a factor of 5 faster.



(a) Species A

(b) Species C

Figure 7-3: Upper and lower bounds on concentration versus time for the stirred-tank reactor (Eqn. (7.22)). Solution trajectories for various constant inputs are thin solid lines. Results from the polyhedral bounding method from Ch. 6 are plotted with circles, while the results from the method in §7.6.1 are plotted with thick black lines.

7.7.3 Simultaneous interval and affine relaxations

This section considers the numerical method from §7.6.2, for construction of simultaneous interval and affine relaxations.

The stirred-tank reactor model from §7.7.2 is considered again. To assess the performance of the affine relaxation method, empirical convergence is studied; relaxations are calculated on a sequence of intervals $\{P_n\}$. Since relaxations are desired in this case, the uncertain quantities will be constant in time; i.e. replace $u_1(t)$, $u_2(t)$, and $u_3(t)$ with p_1, p_2, and p_3, respectively, in Eqn. (7.22). Then $\mathbf{p} \in P_n = [0.9, p^U_{n,1}] \times [0.8, p^U_{n,2}] \times [10, p^U_{n,3}]$, with $\mathbf{p}^U_n$ initially equal to $(1.1, 1.0, 50)$ and decreasing to $\mathbf{p}^L = (0.9, 0.8, 10)$.

Constraint information coming from the stoichiometry of the reaction can be used in this example. This type of information inspires the form of the $\mathbf{A}$ mapping used in §7.7.2; consider the matrix $\mathbf{N}$ used in that definition of $\mathbf{A}$. Let $\mathbf{y}_N = \mathbf{N}\mathbf{x}$. Then $\mathbf{y}_N$ obeys the differential equation (multiply Eqn. (7.22) from the left by $\mathbf{N}$)

$$\dot{\mathbf{y}}_N(t, \mathbf{p}) = \frac{1}{V}\begin{bmatrix} -p_1 v_{\mathrm{A}} + 2p_2 v_{\mathrm{B}} \\ p_1 v_{\mathrm{A}} - p_2 v_{\mathrm{B}} \end{bmatrix} - \frac{v_{\mathrm{A}} + v_{\mathrm{B}}}{V}\mathbf{y}_N(t, \mathbf{p}),$$

with initial conditions $\mathbf{y}_N(0, \mathbf{p}) = \mathbf{0}$. This is a separable system of linear, first-order ODEs and has the solution

$$\mathbf{y}_N(t,\mathbf{p}) = \left(\exp\left(-\frac{v_{\mathrm{A}}+v_{\mathrm{B}}}{V}t\right)-1\right)\left(-\frac{1}{v_{\mathrm{A}}+v_{\mathrm{B}}}\right)\begin{bmatrix} -p_1v_{\mathrm{A}}+2p_2v_{\mathrm{B}} \\ p_1v_{\mathrm{A}}-p_2v_{\mathrm{B}} \end{bmatrix}.$$

At each point in time, we can evaluate this expression in interval arithmetic and obtain an enclosure of $\mathbf{y}_N(t,P_n)$; denote this $[\mathbf{y}_N^L(t,P_n),\mathbf{y}_N^U(t,P_n)]$. This yields an enclosure of two linear combinations of the original differential states, and we can take $X_C : t \mapsto \{\mathbf{z} : \mathbf{N}\mathbf{z} \le \mathbf{y}_N^U(t,P_n), -\mathbf{N}\mathbf{z} \le -\mathbf{y}_N^L(t,P_n), \mathbf{z} \ge \mathbf{0}\}$ (where nonnegativity of the states also has been included).

In addition to the method for simultaneous interval and affine relaxations from §7.6.2, a few other affine or interval relaxation methods are included for comparison; the specifics of the method and its name are listed in Table 7.1. All methods rely on the solution of an auxiliary IVP in ODEs to determine the bounds. In each case, the implementation of the BDF in CVODE with a Newton iteration was used with relative and absolute integration tolerances equal to 10^{-9}. A maximum of 10^5 integration steps was allowed. This is significant since the Affine only methods fail on the larger parameter intervals; the bounds become very large and the numerical integrator must take time steps on the order of machine precision, resulting in integration failure when the maximum number of steps is reached.

Fig. 7-4 summarizes the convergence results; these figures plot the "width" of the bounds at the final time against $\mathrm{diam}(P_n)$. For the interval bounding methods, the width is taken to be the diameter of the interval. For any of the affine relaxations, the width is defined as the maximum difference between the overestimator and underestimator in any dimension (at the final time). That is, for generic affine underestimators $[(\mathbf{a}_j^l)^{\mathrm{T}}]\mathbf{p} + \mathbf{b}^l$ and affine overestimators $[(\mathbf{a}_j^u)^{\mathrm{T}}]\mathbf{p} + \mathbf{b}^u$ on P_n, the width is

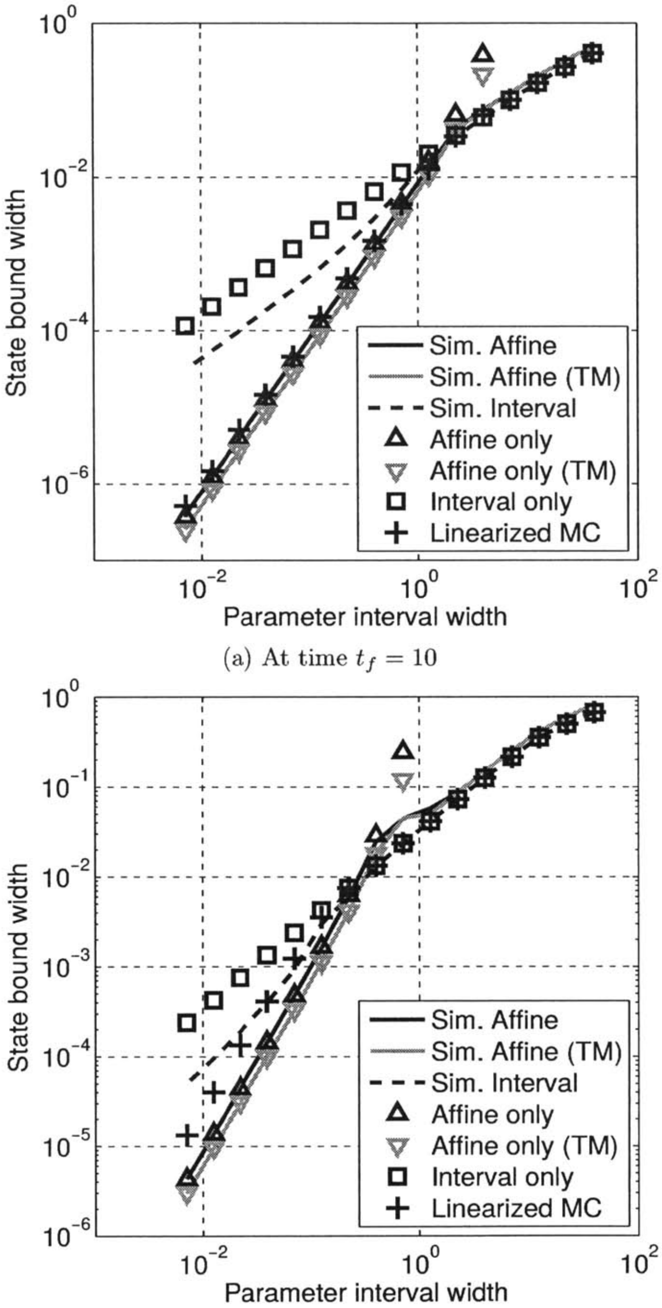
$$\max\left\{ \max\{(\mathbf{a}_j^u - \mathbf{a}_j^l)^{\mathrm{T}}\mathbf{p} + b_j^u - b_j^l : \mathbf{p} \in P_n\} : j \in \{1,\dots,n_y\}\right\}.$$

Thus, just by looking at these widths, one can make conclusions about the (empirical) pointwise and thus Hausdorff convergence order of the relaxations at the final time; see also Ch. 3 of [163].

The performances of the affine relaxation methods are similar on "small" intervals. Meanwhile, the interval methods (and the Linearized MC method) are similar on "large" intervals.

Table 7.1: Names and specifics of methods used in convergence study of §7.7.3.

Name	Description
Sim. Affine/Interval	Affine and interval parts of simultaneous interval and affine relaxation method from §7.6.2, using affine relaxations from Ch. 3 to satisfy Condition 4 of Assumption 7.6.2.
Sim. Affine (TM)	Same as Sim. Affine, but using first-order Taylor-model (and interval) arithmetic to satisfy Assumption 7.6.2 (implementation in MC++ version 0.7 with default options [36], which includes linearization at the midpoint of the parameter interval).
Affine only	Affine relaxation method from §7.4.3, with $\mathbf{A}_p(\cdot) = \frac{\partial \mathbf{x}_y}{\partial \mathbf{p}}(\cdot, \text{mid}(P_n))$ (i.e. sensitivities at a reference trajectory at the midpoint of the parameter interval), and using affine relaxations from Ch. 3 to satisfy Inequalities (7.9).
Affine only (TM)	Same as Affine only, but using first-order Taylor-model arithmetic to satisfy Inequalities (7.9) (implementation in MC++ version 0.7 with default options [36], which includes linearization at the midpoint of the parameter interval).
Interval only	Interval bounds from the method in §7.6.1 with $\widehat{\mathbf{A}} = [-\mathbf{I} \ \ \mathbf{I}]^{\mathrm{T}}$ (includes constraints, but treats parameters as time-varying uncertainty).
Linearized MC	Affine relaxations obtained from subgradients of convex and concave (McCormick) relaxations from [169]. Base interval bounds are "Interval only." Subgradients are evaluated at the midpoint of the parameter interval following the scheme in §10.3.2 of [166] and §2.1.3.3 of [163].



(a) At time $t_f = 10$

(b) At time $t_f = 100$

Figure 7-4: Empirical convergence results for the tank reactor (Eqn. (7.22)) at two different time points. See Table 7.1 for the meaning of the labels. All available data points are visible (i.e. the Affine only methods failed on the larger parameter intervals).

However, the prefactors of the Sim. Affine and Affine only methods seem to depend less heavily on time, compared to the Linearized MC relaxations (comparing Figures 7-4a and 7-4b). Still, all of the affine relaxation methods display second-order convergence once the parameter interval is small enough. This agrees with the analysis of §6 of [202], which establishes that the Affine Only (TM) method should have second-order Hausdorff convergence.

The better of the Interval only and Affine only methods for any parameter interval is of comparable quality to the better of the Sim. Affine and Sim. Interval methods. However, simultaneous calculation of the interval and affine relaxations indeed non-trivially improves the Sim. Affine relaxations compared to the Affine only method on large parameter intervals, as well as the Sim. Interval relaxations compared to the Interval only method on small parameter intervals. Furthermore, calculation of the Affine only and Interval only relaxations separately and then taking the better of the two can be potentially more computationally expensive than the simultaneous calculation as in §7.6.2. For example, consider the calculation of relaxations on the largest parameter interval $P_1 = [0.9, 1.1] \times [0.8, 1.0] \times [10, 50]$ on the time interval $T = [0, 10]$ and using the integration options listed earlier. The (simultaneous) calculation of the Sim. Affine and Sim. Interval relaxations takes 0.026s, while calculation of Affine only and Interval only relaxations separately (and intersecting) takes at least 0.73s, or a factor of almost 30 longer. As mentioned earlier, this relates to the fact that numerical integration for the Affine only relaxations fails, which in general is expensive.

7.8 Conclusions

This chapter has considered the problem of estimating the reachable set of constrained dynamic systems. Specifically, a theory was presented giving conditions under which a time-varying polyhedron bounds all solutions of a constrained dynamic system subject to uncertain initial conditions and inputs. This theory was then specialized to yield a number of specific theories, highlighting the connections to previous work. Even further, new methods for constructing polyhedral bounds and affine relaxations were discussed, and their numerical implementations were assessed with various examples. The state constraints yield a number of interesting connections.

Further work might include fine-tuning the numerical methods presented. These methods depend on overestimating the dynamics on various sets; this will most likely involve interval arithmetic or something similar. As a consequence, the dynamics of the auxiliary system yielding the bounds almost certainly will be nonsmooth. Taking advantage of recent advances in calculating elements of generalized derivatives [96], it might be possible to even further speed up the current methods. In the case of the simultaneous interval and affine relaxation method, a modification of the corrector iteration in BDF, along the lines of the staggered corrector in sensitivity calculation [50], might also speed up the method by taking advantage of the structure of the auxiliary system of ODEs in Proposition 7.6.2. However, the methods as presented are effective.

Chapter 8

Lower-level duality and the global solution of generalized semi-infinite programs

8.1 Introduction

The problem of interest is the generalized semi-infinite program (GSIP):

$$\begin{aligned} f^* = \inf_{\mathbf{x} \in X} \; & f(\mathbf{x}) && \text{(GSIP)} \\ \text{s.t.}\; & g(\mathbf{x},\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in \widehat{Y}(\mathbf{x}), \\ & \widehat{Y}(\mathbf{x}) \equiv \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}\}, \end{aligned}$$

where for $(n_x, n_y, m) \in \mathbb{N}^3$, $X \subset \mathbb{R}^{n_x}$, $Y \subset D_y \subset \mathbb{R}^{n_y}$, X and Y are nonempty, $f : X \to \mathbb{R}$, $g : X \times D_y \to \mathbb{R}$ and $\mathbf{h} : X \times D_y \to \mathbb{R}^m$. A "standard" semi-infinite program (SIP) differs from a GSIP in that the index set of the constraint (i.e. $\widehat{Y}$) does not depend on the decision variable $\mathbf{x}$. The challenge of semi-infinite programming comes from the fact that the cardinality of Y may be greater than $\aleph_0$, which is to say that there may be an uncountable number of constraints.

This chapter focuses on conditions when one can reformulate a GSIP into equivalent, but easier to solve problems, such as SIPs and (finite) nonlinear programs (NLPs). These conditions involve the convexity of the lower-level program as the reformulation involves duality arguments; see §8.2 for the definition of the lower-level program. Similar reformulations have been explored before in the literature, however it is not apparent in previous work that the full benefit of a duality-based reformulation has been realized. Much of this relates to a lack of work on the details or performance of associated numerical methods, especially with connections to global methods. In this chapter, it is demonstrated that a duality-based reformulation can be more readily obtained, is a more numerically tractable problem or can be solved via the solution of simple, numerically tractable problems, and is more flexible and applies to a broader class of GSIP.

The present work is most closely related to that in [42, 106, 184, 185]. These articles exemplify the approach wherein one assumes that $-g(\mathbf{x}, \cdot)$ and $\mathbf{h}(\mathbf{x}, \cdot)$ are convex functions, permitting one to make a global statement about minimizers of the lower-level program. In [184, 185], the lower-level program is replaced with its Karush-Kuhn-Tucker (KKT) conditions, which then yields a finite number of algebraic constraints. The result then is a finite NLP, but the complementarity constraints in the KKT conditions make this NLP a mathematical program with complementarity constraints (MPCC). Although some progress has been made, general-purpose solvers typically have trouble with these problems. Indeed, the majority of the effort in those papers goes to regularizing and solving the MPCC. These shortcomings of the MPCC reformulation motivate the work in [42, 106]. The reformulation obtained in [106] is a finite NLP, but it is for a very specific form of GSIP, and the necessary assumptions are quite restrictive. Meanwhile, [42] marks a movement toward duality-based reformulations of GSIP. A number of finite NLP reformulations are proposed, however, the focus of that work is on establishing the regularity of the local minimizers of those reformulations. This chapter continues the development of duality-based reformulations of GSIP, although the reformulations in this work are based on slightly different duality results, and the focus is on obtaining reformulations that can be solved globally in a tractable manner. In the case that the lower-level program has differentiable data, the reformulation is a finite NLP, while in general the reformulation is an SIP. The specific cases when the lower-level program is a linear or second-order cone program are also considered. These specific cases are interesting because stronger results hold.

Meanwhile, local reduction methods, described in e.g., [183, 187, 188, 189], rely on characterizing the set of local minimizers of the lower-level program. This is typically done by characterizing the set of KKT points. Under the appropriate conditions, this set is finite in a neighborhood around a specific $\mathbf{x}$, and thus describes a finite subset of the constraints which must hold at that point. Consequently, the GSIP is locally equivalent to an NLP. However, the "reduction ansatz" which must hold for all $\mathbf{x} \in X$ for the reduction to hold globally has only been shown to hold "most" of the time for linear problems [183], and it is an open question whether it can be expected to hold in a more general setting. In addition, the references above provide few numerical results for methods based on this approach.

Related to local reduction is the approach of formulating necessary or sufficient conditions for (local) optimality of the GSIP [89, 91, 157]. However, any numerical method developed from these conditions would be local methods. Discretization methods are another class of solution method. Fairly successful global methods for SIP based purely on discretization have a long history (see [30] for an early contribution). Some conceptual attempts to generalize these methods to GSIP have been presented in [67, 188]. Unfortunately, they are outer approximation methods; the solution furnished is not guaranteed to be feasible in the original problem, which can be unacceptable in certain applications, such as design centering. See also [67, 182, 189] for reviews of theory and numerical methods for GSIP. A recent advance for the solution of GSIP with nonconvex lower-level program has been presented in [127]. The numerical method in that work is similar to the method in [122], which provides the basis for the numerical method described in this chapter. As one might expect, the nonconvex lower-level program makes the solution method in [127] slightly more arduous; the method involves the solution of disjunctive or nonsmooth nonlinear programs, which in implementation are reformulated as mixed-integer nonlinear programs. In contrast, the solution method in this work does not introduce any additional nonsmoothness into the problems that must be solved.

The formulation (GSIP) omits additional (finite) inequality or equality constraints on $\mathbf{x}$ when considering the theoretical and numerical aspects, but these are easily included.

The rest of the chapter is organized as follows. Section 8.2 introduces notation and terminology, including the definition of the lower-level program (LLP), dual function, and dual problem. Section 8.3 focuses on the case when the LLP satisfies a strong duality result. In this case, (GSIP) is reformulated as an SIP. Section 8.4 provides a reformulation to a finite NLP. Specifically, Section 8.4.1 discusses the case when the LLP is convex and differentiable. Although an NLP is obtained, it involves the derivatives of functions defining the LLP. Some numerical disadvantages of this reformulation, such as how to obtain explicit expressions for the derivatives, are discussed. Section 8.4.2 discusses the special case of (GSIP) when it has a linear LLP. The merits of this reformulation, compared to the MPCC reformulation based on KKT conditions, are discussed. Sections 8.4.3 and 8.4.4 discuss the advantages and disadvantages of reformulations based on duality results for cone programs. Using the SIP reformulation, a solution method from [122] is adapted to solve the GSIP in Section 8.5. Section 8.6 presents some numerical examples. The examples in Sections 8.6.2 and 8.6.3 demonstrate the advantages of the duality-based reformulation in the case of the linear LLP. The examples in Sections 8.6.4 and 8.6.5 demonstrate the effectiveness of the numerical method from §8.5 for nonlinear LLP. Section 8.7 concludes with final remarks.

8.2 Definitions

Of central importance to (GSIP) is the corresponding *lower-level program* (LLP)

$$g^*(\mathbf{x}) = \sup\{g(\mathbf{x},\mathbf{y}) : \mathbf{h}(\mathbf{x},\mathbf{y}) \le \mathbf{0}, \mathbf{y} \in Y\}. \tag{LLP}$$

If $\widehat{Y}(\mathbf{x})$ is nonempty, then the infinite constraint of (GSIP) is equivalent to $g^*(\mathbf{x}) \le 0$. If $\widehat{Y}(\mathbf{x})$ is empty, then no constraints are required to hold; consequently $\mathbf{x} \in X$ is feasible in (GSIP). In this case, assigning $g^*(\mathbf{x}) = -\infty$ is consistent with the typical definition of the supremum of a real-valued function over an empty set, and permits the characterization of $\mathbf{x} \in X$ feasible in (GSIP) if and only if $g^*(\mathbf{x}) \le 0$.

The reformulations in this work are based on Lagrangian duality theory [21, Ch. 5]. Define the *dual function* of (LLP) as

$$q(\mathbf{x},\boldsymbol{\mu}) = \sup\{g(\mathbf{x},\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) : \mathbf{y} \in Y\} \tag{8.1}$$

for all $\boldsymbol{\mu} \in \mathbb{R}^m$ and $\mathbf{x} \in X$. Since $q(\mathbf{x},\boldsymbol{\mu})$ may equal $+\infty$ for some $(\mathbf{x},\boldsymbol{\mu})$, denote its (effective) domain

$$\mathrm{dom}(q(\mathbf{x},\cdot)) = \{\boldsymbol{\mu} \in \mathbb{R}^m : q(\mathbf{x},\boldsymbol{\mu}) < +\infty\}.$$

Subsequently, define the *dual problem* of (LLP) as

$$q^*(\mathbf{x}) = \inf\{q(\mathbf{x},\boldsymbol{\mu}) : \boldsymbol{\mu} \ge \mathbf{0}, \boldsymbol{\mu} \in \mathrm{dom}(q(\mathbf{x},\cdot))\}. \tag{8.2}$$

Duality results from convex programming will be used; for instance, under appropriate assumptions on g, $\mathbf{h}$, and Y, strong duality asserts that $g^*(\mathbf{x}) = q^*(\mathbf{x})$. Such assumptions can be found in, for instance, Proposition 5.3.1 in [23]. Weak duality will also be useful, which states that $g^*(\mathbf{x}) \leq q^*(\mathbf{x})$ always holds; see Proposition 5.1.3 and discussion in §5.1.4 of [21].

8.3 Reformulation as SIP

This section discusses the relation between (GSIP) and the following SIP for some $M \subset \mathbb{R}^m$:

$$\begin{aligned} f^*_{SIP} = \inf_{(\mathbf{x},\boldsymbol{\mu}) \in X \times M} & f(\mathbf{x}) \\ \text{s.t.}\ & g(\mathbf{x},\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in Y, \\ & \boldsymbol{\mu} \geq \mathbf{0}. \end{aligned} \tag{SIP}$$

Theorem 8.3.1 below provides the core of these theoretical developments; it relies on Assumption 8.3.1, which provides the assertion that strong duality holds for the LLP for a given set M containing the dual variables $\boldsymbol{\mu}$. Establishing specific conditions when Assumption 8.3.1 holds is the focus of much of this section.

The first result establishes that (SIP) is a restriction of (GSIP).

Proposition 8.3.1. *For any $M \subset \mathbb{R}^m$ and for any $(\mathbf{x}, \boldsymbol{\mu})$ feasible in* (SIP), *$\mathbf{x}$ is feasible in* (GSIP). *Consequently, $f^* \leq f^*_{SIP}$.*

Proof. If (SIP) is infeasible, then the result holds trivially. Otherwise, choose $(\mathbf{x}, \boldsymbol{\mu})$ feasible in (SIP). Then, $q(\mathbf{x}, \boldsymbol{\mu}) = \sup\{g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\} \leq 0$. Thus $q^*(\mathbf{x}) \leq q(\mathbf{x}, \boldsymbol{\mu}) \leq 0$ since $\boldsymbol{\mu} \geq \mathbf{0}$. By weak duality, $g^*(\mathbf{x}) \leq q^*(\mathbf{x}) \leq 0$, and so $\mathbf{x}$ is feasible in (GSIP). It follows that $f^* \leq f^*_{SIP}$. □

Under Assumption 8.3.1 below, a stronger conclusion can be made.

Assumption 8.3.1. *For given $M \subset \mathbb{R}^m$, assume that for each $\mathbf{x} \in X$ there exists $\boldsymbol{\mu} \in M$, $\boldsymbol{\mu} \geq \mathbf{0}$, such that*

1. *if $g^*(\mathbf{x})$ is finite, $g^*(\mathbf{x}) \leq 0 \implies q(\mathbf{x}, \boldsymbol{\mu}) = g^*(\mathbf{x})$,*
2. *if $g^*(\mathbf{x}) = -\infty$, then $q(\mathbf{x}, \boldsymbol{\mu}) \leq 0$.*

The following result is similar to an unproved claim in [189]. The result establishes the equivalence of (GSIP) and (SIP).

Theorem 8.3.1. *For $M \subset \mathbb{R}^m$, let Assumption 8.3.1 hold. Then for any* $\mathbf{x}$ *feasible in* (GSIP), *there exists* $\boldsymbol{\mu} \in \mathbb{R}^m$ *such that* $(\mathbf{x}, \boldsymbol{\mu})$ *is feasible in* (SIP), *and conversely for any* $(\mathbf{x}, \boldsymbol{\mu})$ *feasible in* (SIP), $\mathbf{x}$ *is feasible in* (GSIP). *Consequently,* $f^* = f^*_{SIP}$.

Proof. Consider $\mathbf{x}$ which is feasible in (GSIP). Then $g^*(\mathbf{x}) \leq 0$. By Assumption 8.3.1, there exists a $\boldsymbol{\mu} \in M$, $\boldsymbol{\mu} \geq \mathbf{0}$, such that $q(\mathbf{x}, \boldsymbol{\mu}) \leq 0$. It follows that

$$g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}} \mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in Y.$$

Thus $(\mathbf{x}, \boldsymbol{\mu})$ is feasible in (SIP).

Conversely, by Proposition 8.3.1 for any $(\mathbf{x}, \boldsymbol{\mu})$ feasible in (SIP), $\mathbf{x}$ is feasible in (GSIP). The equality of the optimal objective values follows. □

The rest of this section is devoted to establishing conditions under which Assumption 8.3.1 holds. Of specific interest are conditions when Assumption 8.3.1 holds for a bounded M. For unbounded M, the most obvious and immediate case is when strong duality holds for (LLP).

Lemma 8.3.2. *Suppose Y is convex, and for all* $\mathbf{x} \in X$, $g(\mathbf{x}, \cdot)$ *and* $-\mathbf{h}(\mathbf{x}, \cdot)$ *are concave on* Y, $g^*(\mathbf{x})$ *is finite, and there exists* $\mathbf{y}_s(\mathbf{x}) \in Y$ *such that* $\mathbf{h}(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) < \mathbf{0}$. *Then for* $M = \mathbb{R}^m$, *Assumption 8.3.1 holds.*

Proof. Under these assumptions, strong duality for the LLP holds for each $\mathbf{x} \in X$; see Proposition 5.3.1 in [23]. This states that $g^*(\mathbf{x}) = q^*(\mathbf{x})$, and the dual problem achieves its infimum. Thus for some $\boldsymbol{\mu} \geq \mathbf{0}$, $q^*(\mathbf{x}) = q(\mathbf{x}, \boldsymbol{\mu})$. □

As mentioned, showing that Assumption 8.3.1 holds for bounded M is of more interest. One reason is that many algorithms for SIP require that the decision variables are contained in a compact set [28, 29, 30, 53, 126, 122]. The numerical method for GSIP described in §8.5 is based on the method for SIP in [122], and indeed establishing that the SIP reformulation (SIP) holds for bounded M is critical. In Lemma 8.3.2 the condition that there exists a $\mathbf{y}_s(\mathbf{x}) \in Y$ such that $\mathbf{h}(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) < \mathbf{0}$ is known as the Slater condition. A slightly stronger assumption than the Slater condition allows a practical way of bounding the solution sets

$S(\mathbf{x})$ of the dual problem, and thus yields a bounded M. However, it should be noted that it is sufficient to show that there exists a bounded $M \subset \mathbb{R}^m$ such that $M \cap S(\mathbf{x}) \neq \varnothing$ for all $\mathbf{x} \in X$; this is a direction for future research.

Lemma 8.3.3. *Suppose that for all $\mathbf{x} \in X$, $g^*(\mathbf{x}) = q^*(\mathbf{x})$ and the dual problem achieves its infimum. Further, suppose there exist $\mathbf{y}_s(\mathbf{x}) \in Y$, $g_b > 0$, and $\mathbf{h}_b > \mathbf{0}$, such that $g(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) > -g_b$ and $\mathbf{h}(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) \leq -\mathbf{h}_b$ for all $\mathbf{x} \in X$. Then Assumption 8.3.1 holds for compact $M = [\mathbf{0}, \mathbf{b}^*]$, where $\mathbf{b}^*$ is given by $b_i^* = g_b/h_{b,i}$ for each $i \in \{1, \dots, m\}$.*

Proof. We wish to establish that M contains the solution set $S(\mathbf{x})$ of the dual problem for all $\mathbf{x}$ feasible in (GSIP). First let

$$\widehat{q}(\mathbf{x}, \boldsymbol{\mu}) = g(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}_s(\mathbf{x})),$$

which we note for all $\boldsymbol{\mu}$ satisfies $\widehat{q}(\mathbf{x}, \boldsymbol{\mu}) \leq q(\mathbf{x}, \boldsymbol{\mu})$, since $\mathbf{y}_s(\mathbf{x}) \in Y$ and by the definition of q as a supremum over Y. Next, let

$$\widehat{S}(\mathbf{x}) = \{\boldsymbol{\mu} \geq \mathbf{0} : \widehat{q}(\mathbf{x}, \boldsymbol{\mu}) \leq 0\}.$$

Assume now that $\mathbf{x}$ is feasible in (GSIP). Then $g^*(\mathbf{x}) \leq 0$. Note that if $\boldsymbol{\mu} \in S(\mathbf{x})$, then $\boldsymbol{\mu} \geq \mathbf{0}$ and by strong duality we have

$$\widehat{q}(\mathbf{x}, \boldsymbol{\mu}) \leq q(\mathbf{x}, \boldsymbol{\mu}) = q^*(\mathbf{x}) \leq 0 \implies \boldsymbol{\mu} \in \widehat{S}(\mathbf{x}).$$

Thus $S(\mathbf{x}) \subset \widehat{S}(\mathbf{x})$. Consequently, we could obtain various norm-bounds on $\widehat{S}(\mathbf{x})$ (and subsequently $S(\mathbf{x})$ as well) by solving $\max\{\|\boldsymbol{\mu}\| : \boldsymbol{\mu} \in \widehat{S}(\mathbf{x})\}$. However, it is easy and useful to obtain bounds on each component of $\boldsymbol{\mu}$ separately. So, consider

$$b_i(\mathbf{x}) = \max\left\{\mu_i : \boldsymbol{\mu} \geq \mathbf{0}, g(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) - \sum_j \mu_j h_j(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) \leq 0\right\}$$

for each i. Since $h_j(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) < 0$ for each j, the above program achieves its maximum if $\mu_j = 0$, $j \neq i$. Since $g(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) \leq g^*(\mathbf{x}) \leq 0$, we have

$$\mu_i \leq \frac{g(\mathbf{x}, \mathbf{y}_s(\mathbf{x}))}{h_i(\mathbf{x}, \mathbf{y}_s(\mathbf{x}))} \implies b_i(\mathbf{x}) = \frac{g(\mathbf{x}, \mathbf{y}_s(\mathbf{x}))}{h_i(\mathbf{x}, \mathbf{y}_s(\mathbf{x}))}.$$

Since $S(\mathbf{x})$ is a subset of the nonnegative orthant, it must be a subset of $[\mathbf{0}, \mathbf{b}(\mathbf{x})]$.

Finally, since $g(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) \leq g^*(\mathbf{x}) \leq 0$, and for each i, $h_i(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) < 0$, an upper bound for $b_i(\mathbf{x})$ on X is given by $b_i^* = g_b/h_{b,i}$. If follows that M is compact and that Assumption 8.3.1 holds. □

The assumptions for the numerical method described in §8.5 include continuity of the defining functions and compactness of X. Under these assumptions, if in addition one has knowledge of a continuously parameterized Slater point $\mathbf{y}_s$ (which naturally falls out of some design centering problems), then it is clear that the constants g_b and $\mathbf{h}_b$ required by Lemma 8.3.3 exist.

Lemma 8.3.4. *Let the assumptions of Lemma 8.3.2 hold. Assume X is compact, g and $\mathbf{h}$ are continuous, and that there exists continuous $\mathbf{y}_s : X \to Y$ such that $\mathbf{h}(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) < \mathbf{0}$ for each $\mathbf{x} \in X$. Then there exists $g_b > 0$ and $\mathbf{h}_b > \mathbf{0}$ such that the conclusion of Lemma 8.3.3 holds.*

8.4 Reformulation as NLP

This section discusses the case when one can reformulate (GSIP) as a finite NLP. Proposition 8.4.1 is similar to Corollary 2.4 in [42]; however it does not require that $g(\mathbf{x}, \cdot)$ is concave on all of $\mathbb{R}^{n_y}$ as in the latter result. Further, this section then specializes this result for a number of cases, such as when the lower-level program is a linear program. These special cases have important implications for global optimization; namely, the lower-level variables $\mathbf{y}$ do not appear in the reformulated problems, which in general improves the run time of branch and bound. While the advantages of obtaining an NLP (versus SIP) reformulation are clear, some potential numerical disadvantages are discussed.

8.4.1 General convex LLP

Since the sum of convex functions is convex, we see, under the assumptions of Lemma 8.3.2, that the lower-level program of (SIP), $\sup\{g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\}$, is a convex program when $\boldsymbol{\mu} \geq \mathbf{0}$. Thus, one might ask why not try to reformulate (SIP) as a simpler problem. If we have convex functions defining Y, duality arguments similar to the ones employed already do not reduce the SIP to an NLP; instead we merely dualize the constraints defining Y and obtain an SIP in which the index set of the infinite constraint is now $\mathbb{R}^{n_y}$.

However, a reformulation of (GSIP) to an NLP is possible. This is inspired by the observation that if $g(\mathbf{x}, \cdot) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \cdot)$ is concave and differentiable on open convex Y, then its maximum is achieved at $\mathbf{y}$ if and only if $\nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y})\boldsymbol{\mu} = \mathbf{0}$. Then the infinite constraint in (SIP), $g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0$ for all $\mathbf{y} \in Y$, can be replaced with $g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0$ for any $\mathbf{y}$ such that $\nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y})\boldsymbol{\mu} = \mathbf{0}$. The benefit is that the resulting reformulation is not an MPCC.

Proposition 8.4.1. *For $M \subset \mathbb{R}^m$, let Assumption 8.3.1 hold. Suppose D_y is an open set and Y is an open convex set. Suppose that for all $\mathbf{x} \in X$, $g(\mathbf{x}, \cdot)$ and $\mathbf{h}(\mathbf{x}, \cdot)$ are differentiable on D_y, $g(\mathbf{x}, \cdot)$ and $-\mathbf{h}(\mathbf{x}, \cdot)$ are concave on $\overline{Y}$, and* (LLP) *achieves its supremum. Consider the NLP*

$$\begin{aligned} \inf_{\mathbf{x},\mathbf{y},\boldsymbol{\mu}} \; & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0, \\ & \nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y})\boldsymbol{\mu} = \mathbf{0}, \\ & \boldsymbol{\mu} \geq \mathbf{0}, \\ & (\mathbf{x}, \mathbf{y}, \boldsymbol{\mu}) \in X \times \overline{Y} \times M. \end{aligned} \tag{8.3}$$

Then for any $\mathbf{x}$ feasible in (GSIP), *there exists $(\mathbf{y}, \boldsymbol{\mu}) \in Y \times \mathbb{R}^m$ such that $(\mathbf{x}, \mathbf{y}, \boldsymbol{\mu})$ is feasible in NLP* (8.3), *and conversely for any $(\mathbf{x}, \mathbf{y}, \boldsymbol{\mu})$ feasible in NLP* (8.3), *$\mathbf{x}$ is feasible in* (GSIP).

Proof. Choose $\mathbf{x}$ feasible in (GSIP). Then $g^*(\mathbf{x}) \leq 0$, and by assumption there exists a maximizer $\mathbf{y}_x$ of (LLP), so $g^*(\mathbf{x})$ is also finite. So by Assumption 8.3.1, there exists $\boldsymbol{\mu}_x \in M$, $\boldsymbol{\mu}_x \geq \mathbf{0}$, such that $q(\mathbf{x}, \boldsymbol{\mu}_x) = g^*(\mathbf{x})$. In other words, $\boldsymbol{\mu}_x$ is a duality (or Lagrange) multiplier for (LLP). Then by Proposition 5.1.1 in [21], for example, $\mathbf{y}_x \in \arg\max\{g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}_x^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\}$. This implies $g(\mathbf{x}, \mathbf{y}_x) - \boldsymbol{\mu}_x^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}_x) \leq 0$. And since $g(\mathbf{x}, \cdot)$ and $-\mathbf{h}(\mathbf{x}, \cdot)$ are differentiable and Y is open, this implies $\nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}_x) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y}_x)\boldsymbol{\mu}_x = \mathbf{0}$. It follows that $(\mathbf{x}, \mathbf{y}_x, \boldsymbol{\mu}_x)$ is feasible in NLP (8.3).

Conversely choose $(\mathbf{x}, \mathbf{y}, \boldsymbol{\mu})$ feasible in NLP (8.3). Again, since $g(\mathbf{x}, \cdot)$ and $-\mathbf{h}(\mathbf{x}, \cdot)$ are concave and differentiable and $\overline{Y}$ is convex, $\nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y})\boldsymbol{\mu} = \mathbf{0}$ implies $\mathbf{y}$ is a maximizer of $g(\mathbf{x}, \cdot) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \cdot)$ on $\overline{Y}$. It follows that $(\mathbf{x}, \boldsymbol{\mu})$ is feasible in (SIP), and so by Proposition 8.3.1, $\mathbf{x}$ is feasible in (GSIP). □

Although the reformulation in Proposition 8.4.1 is a finite NLP, it has a potential numerical disadvantage. This relates to the fact that the reformulation requires derivative information. Although not too difficult to obtain in many cases, many deterministic global optimization software (such as BARON [197, 159]) require an explicit form for the functions defining the constraints. In the case of (8.3), this means that an explicit form for the derivatives is necessary. While obtaining an explicit expression is not impossible, numerical derivative information is much easier to obtain. The solution method described in §8.5, based on the SIP reformulation, requires the solution of various NLP subproblems; these subproblems are defined in terms of the original functions f, g, and $\mathbf{h}$ defining the (GSIP), and their solution would typically require at most numerical derivative information.

As a specific case, consider applying Proposition 8.4.1 when the LLP is a convex quadratic program (QP):

$$g^*(\mathbf{x}) = \sup\left\{(1/2)\mathbf{y}^{\mathrm{T}}\mathbf{Q}(\mathbf{x})\mathbf{y} + \mathbf{c}(\mathbf{x})^{\mathrm{T}}\mathbf{y} + d(\mathbf{x}) : \mathbf{A}(\mathbf{x})\mathbf{y} \le \mathbf{b}(\mathbf{x})\right\},$$

for some $\mathbf{Q} : X \to \mathbb{R}^{n_y \times n_y}$ which is negative-definite-valued, $\mathbf{c} : X \to \mathbb{R}^{n_y}$, $d : X \to \mathbb{R}$, $\mathbf{A} : X \to \mathbb{R}^{m \times n_y}$, and $\mathbf{b} : X \to \mathbb{R}^m$ (and assuming $Y = \mathbb{R}^{n_y}$). Since $\mathbf{Q}(\mathbf{x})$ is negative-definite, $g^*(\mathbf{x})$ is finite and the LLP achieves its supremum for all $\mathbf{x}$ such that the LLP is feasible. Then by Proposition 5.2.1 in [21], for instance, strong duality holds for all such $\mathbf{x}$, and so Assumption 8.3.1 holds for $M = \mathbb{R}^m$. Further, the stationarity condition for the Lagrangian appearing in the constraints of NLP (8.3) becomes

$$\mathbf{Q}(\mathbf{x})\mathbf{y} + \mathbf{c}(\mathbf{x}) - \mathbf{A}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\mu} = \mathbf{0}.$$

The preceding discussion is formalized in the following.

Corollary 8.4.1. *Suppose that $Y = D_y = \mathbb{R}^{n_y}$ and $g : (\mathbf{x}, \mathbf{y}) \mapsto (1/2)\mathbf{y}^{\mathrm{T}}\mathbf{Q}(\mathbf{x})\mathbf{y} + \mathbf{c}(\mathbf{x})^{\mathrm{T}}\mathbf{y} + d(\mathbf{x})$ for some $\mathbf{Q} : X \to \mathbb{R}^{n_y \times n_y}$ which is negative-definite-valued, $\mathbf{c} : X \to \mathbb{R}^{n_y}$, $d : X \to \mathbb{R}$, and $\mathbf{h} : (\mathbf{x}, \mathbf{y}) \mapsto \mathbf{A}(\mathbf{x})\mathbf{y} - \mathbf{b}(\mathbf{x})$ for some $\mathbf{A} : X \to \mathbb{R}^{m \times n_y}$ and $\mathbf{b} : X \to \mathbb{R}^m$. Suppose that* *for all* $\mathbf{x} \in X$ *there exists* $\mathbf{y} \in \mathbb{R}^{n_y}$ *such that* $\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}$. *Consider the NLP*

$$\begin{aligned} \inf_{\mathbf{x},\mathbf{y},\boldsymbol{\mu}} \; & f(\mathbf{x}) \\ \text{s.t. } & (1/2)\mathbf{y}^{\mathrm{T}}\mathbf{Q}(\mathbf{x})\mathbf{y} + \mathbf{c}(\mathbf{x})^{\mathrm{T}}\mathbf{y} + d(\mathbf{x}) - \boldsymbol{\mu}^{\mathrm{T}}(\mathbf{A}\mathbf{y} - \mathbf{b}(\mathbf{x})) \leq 0, \\ & \mathbf{Q}(\mathbf{x})\mathbf{y} + \mathbf{c}(\mathbf{x}) - \mathbf{A}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\mu} = \mathbf{0}, \\ & \boldsymbol{\mu} \geq \mathbf{0}, \\ & (\mathbf{x}, \mathbf{y}, \boldsymbol{\mu}) \in X \times \mathbb{R}^{n_y} \times \mathbb{R}^m. \end{aligned} \tag{8.4}$$

Then for any $\mathbf{x}$ *feasible in* (GSIP), *there exists* $(\mathbf{y}, \boldsymbol{\mu}) \in \mathbb{R}^{n_y} \times \mathbb{R}^m$ *such that* $(\mathbf{x}, \mathbf{y}, \boldsymbol{\mu})$ *is feasible in NLP* (8.4), *and conversely for any* $(\mathbf{x}, \mathbf{y}, \boldsymbol{\mu})$ *feasible in NLP* (8.4), $\mathbf{x}$ *is feasible in* (GSIP).

The main reason for pointing out this specific case is the fact that it relies on a strong duality result (Proposition 5.2.1 in [21]) that does not require a Slater point, as in Lemma 8.3.2, to satisfy Assumption 8.3.1. More generally, in the case that $g^*(\mathbf{x})$ is finite, $g(\mathbf{x}, \cdot)$ is concave on $Y = \mathbb{R}^{n_y}$, and the feasible set of the LLP is nonempty and a polyhedron for all $\mathbf{x}$, we could applying Proposition 5.2.1 in [21] to show that Assumption 8.3.1 is satisfied for $M = \mathbb{R}^m$ and subsequently apply Proposition 8.4.1.

If $\mathbf{Q}$ is constant or diagonal-valued, it may be possible to explicitly invert it. Then the stationarity condition of the Lagrangian implies

$$\mathbf{y} = (\mathbf{Q}(\mathbf{x}))^{-1}(\mathbf{A}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{c}(\mathbf{x})).$$

In this case, the variables $\mathbf{y}$ can be removed from problem (8.4) by replacing them with the expression above. As mentioned, this reduction in the number of variables can have a significant impact on the runtime of branch and bound, which in general has worst-case exponential scaling in the number of decision variables.

8.4.2 Linear LLP

This section considers the case that g and $\mathbf{h}$ are affine in $\mathbf{y}$ for each $\mathbf{x} \in X$. The subsequent reformulation is a specific case of the reformulation in §8.4.1. As mentioned, a reformulation in [42] is similar to the one in §8.4.1, and both require the assumption that (LLP) is feasible for each $\mathbf{x} \in X$. However, in the specific case of a linear LLP, it is established below that the equivalence can still hold when the LLP is infeasible. Further, the numerical disadvantages mentioned in the previous section, that analytical derivative information is needed, no longer apply. Consequently, it is worthwhile to focus on this specific case. This is specified in the following assumption.

Assumption 8.4.1. *Assume that* $Y = D_y = \mathbb{R}^{n_y}$, $g : (\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}\mathbf{y} + d(\mathbf{x})$, *and* $\mathbf{h} : (\mathbf{x}, \mathbf{y}) \mapsto \mathbf{A}(\mathbf{x})\mathbf{y} - \mathbf{b}(\mathbf{x})$ *for some functions* $\mathbf{c} : X \to \mathbb{R}^{n_y}$, $d : X \to \mathbb{R}$, $\mathbf{A} : X \to \mathbb{R}^{m \times n_y}$, *and* $\mathbf{b} : X \to \mathbb{R}^m$.

Under Assumption 8.4.1 the LLP for (GSIP) is the linear program (LP)

$$g^*(\mathbf{x}) = d(\mathbf{x}) + \sup\{(\mathbf{c}(\mathbf{x}))^{\mathrm{T}}\mathbf{y} : \mathbf{A}(\mathbf{x})\mathbf{y} \leq \mathbf{b}(\mathbf{x})\}. \tag{8.5}$$

The dual problem is also an LP and has the form

$$q^*(\mathbf{x}) = d(\mathbf{x}) + \inf\{\mathbf{p}^{\mathrm{T}}\mathbf{b}(\mathbf{x}) : \mathbf{p}^{\mathrm{T}}\mathbf{A}(\mathbf{x}) = (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}, \mathbf{p} \geq \mathbf{0}\}. \tag{8.6}$$

Theorem 8.4.2. *Let Assumption 8.4.1 hold. Consider the following finite NLP:*

$$\begin{aligned} \inf_{(\mathbf{x},\mathbf{p}) \in X \times \mathbb{R}^m} & f(\mathbf{x}) \\ \text{s.t. } & \mathbf{p}^{\mathrm{T}}\mathbf{b}(\mathbf{x}) + d(\mathbf{x}) \leq 0, \\ & \mathbf{p}^{\mathrm{T}}\mathbf{A}(\mathbf{x}) - (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}, \mathbf{p} \geq \mathbf{0}. \end{aligned} \tag{8.7}$$

Suppose that for each $\mathbf{x} \in X$, $\{\mathbf{p} \in \mathbb{R}^m : \mathbf{p}^{\mathrm{T}}\mathbf{A}(\mathbf{x}) = (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}, \mathbf{p} \geq \mathbf{0}\}$ *is nonempty. Then for all* $\mathbf{x}$ *feasible in* (GSIP), *there exists* $\mathbf{p} \in \mathbb{R}^m$ *such that* $(\mathbf{x}, \mathbf{p})$ *is feasible in NLP* (8.7), *and for all* $(\mathbf{x}, \mathbf{p})$ *feasible in NLP* (8.7), $\mathbf{x}$ *feasible in* (GSIP).

Proof. Choose $\mathbf{x}$ feasible in (GSIP); then $g^*(\mathbf{x}) \leq 0$. Consider first the case that $g^*(\mathbf{x}) = -\infty$. Since d is real (finite)-valued, it follows that the LLP (8.5) is infeasible. By assumption, the dual LP (8.6) is feasible; consequently it either has a solution or is unbounded. But by linear programming duality theory, see for instance Table 4.2 in [25], the dual LP (8.6) must be unbounded. In other words, for any finite R, there exists $\mathbf{p}$ satisfying $\mathbf{p}^{\mathrm{T}}\mathbf{A}(\mathbf{x}) = (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}$, $\mathbf{p} \geq \mathbf{0}$, $\mathbf{p}^{\mathrm{T}}\mathbf{b}(\mathbf{x}) < R$. Thus, choose a $\mathbf{p}$ such that $\mathbf{p}^{\mathrm{T}}\mathbf{b}(\mathbf{x}) < -d(\mathbf{x})$ (and $\mathbf{p}^{\mathrm{T}}\mathbf{A}(\mathbf{x}) = (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}$, $\mathbf{p} \geq \mathbf{0}$). It follows that $(\mathbf{x}, \mathbf{p})$ is feasible in NLP (8.7). Otherwise, if $-\infty < g^*(\mathbf{x}) \leq 0$, then LLP (8.5) has a finite optimum, and again linear programming duality asserts that

$g^*(\mathbf{x}) = q^*(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}\mathbf{b}(\mathbf{x}) + d(\mathbf{x})$ for some $\mathbf{p}$ satisfying $\mathbf{p}^{\mathrm{T}}\mathbf{A}(\mathbf{x}) = (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}$, $\mathbf{p} \geq \mathbf{0}$ (see for instance Theorem 4.4 in [25]). Again, it follows that $(\mathbf{x}, \mathbf{p})$ is feasible in NLP (8.7).

Conversely, choose $(\mathbf{x}, \mathbf{p})$ feasible in NLP (8.7). If the dual LP (8.6) is unbounded, then again we must have that LLP (8.5) is infeasible, and so $\mathbf{x}$ is feasible in (GSIP). Otherwise, the dual LP (8.6) has a finite optimum and $g^*(\mathbf{x}) = q^*(\mathbf{x})$, and by the definition of $q^*(\mathbf{x})$ as an infimum we must have $q^*(\mathbf{x}) \leq \mathbf{p}^{\mathrm{T}}\mathbf{b}(\mathbf{x}) + d(\mathbf{x}) \leq 0$. Thus, $g^*(\mathbf{x}) \leq 0$ and the feasibility of $\mathbf{x}$ in (GSIP) follows. □

When an LP has a solution, it is equivalent to its KKT conditions. Thus, when LLP (8.5) is feasible and bounded, it is easy to see that (GSIP) is equivalent to the NLP

$$\begin{aligned} \inf_{(\mathbf{x},\mathbf{y},\mathbf{p}) \in X \times \mathbb{R}^{n_y} \times \mathbb{R}^m} & f(\mathbf{x}) \\ \text{s.t. } & (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}\mathbf{y} + d(\mathbf{x}) \leq 0, \\ & \mathbf{A}(\mathbf{x})\mathbf{y} \leq \mathbf{b}(\mathbf{x}), \\ & \mathbf{p}^{\mathrm{T}}\mathbf{A}(\mathbf{x}) = (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}, \mathbf{p} \geq \mathbf{0}, \\ & p_i(\mathbf{a}_i^{\mathrm{T}}\mathbf{y} - b_i(\mathbf{x})) = 0, \forall i \in \{1, \dots, m\}, \end{aligned} \tag{8.8}$$

where $\mathbf{a}_i^{\mathrm{T}}$ is the i^{th} row of $\mathbf{A}$. As mentioned, this is the subject of [184, 185], among others. The most obvious differences between (8.7) and (8.8) are the inclusion in (8.8) of the lower-level primal variables $\mathbf{y}$ and the related constraints, and the complementary slackness conditions or complementarity constraints $p_i(\mathbf{a}_i^{\mathrm{T}}\mathbf{y} - b_i(\mathbf{x})) = 0$ for each i. As mentioned, this makes NLP (8.8) an MPCC.

However, a significant theoretical difference between the NLPs (8.7) and (8.8) is the fact that (8.7) is still equivalent to the (GSIP) when the lower-level program (8.5) is infeasible, as established in Theorem 8.4.2. Meanwhile, if for some $\mathbf{x} \in X$ the LLP (8.5) is infeasible, then there does not exist $(\mathbf{y}, \mathbf{p})$ satisfying the KKT conditions in the constraints of (8.8), and so $\mathbf{x}$ is infeasible in the NLP (8.8). These differences suggest that the duality-based reformulation (8.7) is superior to the reformulation based on the KKT conditions (8.8). Indeed, the examples in §8.6 establish both the theoretical and numerical benefits of the duality-based reformulation (8.7).

Note that the numerical tractability and expanded applicability of the duality-based reformulation stem from the same reason: the exclusion of a representation of the optimal solution set of the LLP (8.5) in the constraints of the NLP (8.7), unlike in the KKT-based reformulation. In fact, the constraints of the KKT-based reformulation also include a representation of the feasible set and optimal solution set of the dual LP (8.2). Results from parametric programming establish that these solution sets can have undesirable parametric properties, such as being non-singleton valued [54]. In contrast, the constraints of the duality-based reformulation (8.7) merely represent the dual feasible set.

8.4.3 Second-order cone program LLP

This section focuses on the case that the LLP is a second-order cone program (SOCP), which is a class of programs for which strong duality holds. One complication that prevents the application of Proposition 8.4.1 to this case is that fact that the constraints involve the two-norm, and thus **h** is not differentiable. This case is specified by the following assumption.

Assumption 8.4.2. *Assume that* $Y = D_y = \mathbb{R}^{n_y}$, $g : (\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}\mathbf{y} + s(\mathbf{x})$ *for some functions* $\mathbf{c} : X \to \mathbb{R}^{n_y}$ *and* $s : X \to \mathbb{R}$, *and for each* $i \in \{1, \dots, m\}$, $h_i : (\mathbf{x}, \mathbf{y}) \mapsto \|\mathbf{A}_i(\mathbf{x})\mathbf{y} + \mathbf{b}_i(\mathbf{x})\|_2 - (\mathbf{e}_i(\mathbf{x}))^{\mathrm{T}}\mathbf{y} - d_i(\mathbf{x})$ *for some functions* $\mathbf{A}_i : X \to \mathbb{R}^{n_i \times n_y}$, $\mathbf{b}_i : X \to \mathbb{R}^{n_i}$, $\mathbf{e}_i : X \to \mathbb{R}^{n_y}$, *and* $d_i : X \to \mathbb{R}$.

Under Assumption 8.4.2, the LLP is the SOCP

$$g^*(\mathbf{x}) = s(\mathbf{x}) + \sup_{\mathbf{y} \in \mathbb{R}^{n_y}} (\mathbf{c}(\mathbf{x}))^{\mathrm{T}}\mathbf{y} \qquad (8.9)$$
$$\text{s.t. } \|\mathbf{A}_i(\mathbf{x})\mathbf{y} + \mathbf{b}_i(\mathbf{x})\|_2 \leq (\mathbf{e}_i(\mathbf{x}))^{\mathrm{T}}\mathbf{y} + d_i(\mathbf{x}), \quad \forall i \in \{1, \dots, m\}.$$

The dual problem is also an SOCP (see for instance [109]):

$$q^*(\mathbf{x}) = s(\mathbf{x}) + \inf_{\mathbf{z}_1, w_1, \dots, \mathbf{z}_m, w_m} \sum_{i=1}^{m} \mathbf{z}_i^{\mathrm{T}}\mathbf{b}_i(\mathbf{x}) + w_i d_i(\mathbf{x}) \qquad (8.10)$$
$$\text{s.t. } \sum_{i=1}^{m} \mathbf{z}_i^{\mathrm{T}}\mathbf{A}_i(\mathbf{x}) + w_i(\mathbf{e}_i(\mathbf{x}))^{\mathrm{T}} = -(\mathbf{c}(\mathbf{x}))^{\mathrm{T}},$$
$$\|\mathbf{z}_i\|_2 \leq w_i, \quad (\mathbf{z}_i, w_i) \in \mathbb{R}^{n_i} \times \mathbb{R}, \quad \forall i \in \{1, \dots, m\}.$$

Linear programs, convex quadratically-constrained quadratic programs, and (convex) QP are special cases of SOCP. However, the duality property for SOCP used in the following result requires the existence of a Slater point for the SOCP LLP (8.9). In contrast, a Slater point is not required when the LLP is a QP or linear program (see Corollary 8.4.1 and Theorem 8.4.2).

Theorem 8.4.3. *Let Assumption 8.4.2 hold. Consider the following finite NLP:*

$$\begin{aligned} \inf_{\mathbf{x},\mathbf{z}_1,w_1,\ldots,\mathbf{z}_m,w_m} \quad & f(\mathbf{x}) \\ \text{s.t. } & s(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{z}_i^{\mathrm{T}}\mathbf{b}_i(\mathbf{x}) + w_i d_i(\mathbf{x}) \le 0, \\ & \sum_{i=1}^{m} \mathbf{z}_i^{\mathrm{T}}\mathbf{A}_i(\mathbf{x}) + w_i(\mathbf{e}_i(\mathbf{x}))^{\mathrm{T}} = -(\mathbf{c}(\mathbf{x}))^{\mathrm{T}}, \\ & \|\mathbf{z}_i\|_2 \le w_i, \quad (\mathbf{z}_i, w_i) \in \mathbb{R}^{n_i} \times \mathbb{R}, \quad \forall i \in \{1,\ldots,m\}, \\ & \mathbf{x} \in X. \end{aligned} \tag{8.11}$$

Suppose that for all $\mathbf{x} \in X$*, there exists* $\mathbf{y}_s(\mathbf{x})$ *such that*

$$\|\mathbf{A}_i(\mathbf{x})\mathbf{y}_s(\mathbf{x}) + \mathbf{b}_i(\mathbf{x})\|_2 < (\mathbf{e}_i(\mathbf{x}))^{\mathrm{T}}\mathbf{y}_s(\mathbf{x}) + d_i(\mathbf{x}), \quad \forall i.$$

Then for all $\mathbf{x}$ *feasible in* (GSIP) *there exists* $\mathbf{v} \equiv (\mathbf{z}_1, w_1, \ldots, \mathbf{z}_m, w_m)$ *such that* $(\mathbf{x}, \mathbf{v})$ *is feasible in NLP* (8.11)*, and conversely for all* $(\mathbf{x}, \mathbf{z}_1, w_1, \ldots, \mathbf{z}_m, w_m)$ *feasible in NLP* (8.11)*,* $\mathbf{x}$ *is feasible in* (GSIP)*.*

Proof. Under the assumption that (8.9) is bounded (has bounded optimal objective value) and has a feasible point that strictly satisfies the inequality constraints, then strong duality holds and a dual optimal solution exists; see Theorem 4.2.1 in [134]. Then for $\mathbf{x}$ such that $g^*(\mathbf{x}) \le 0$, (8.9) has bounded optimal objective value, and by assumption there exists a strictly feasible point, and so there exists $(\mathbf{z}_1, w_1, \ldots, \mathbf{z}_m, w_m)$ feasible in the dual SOCP (8.10) with $s(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{z}_i^{\mathrm{T}}\mathbf{b}_i(\mathbf{x}) + w_i d_i(\mathbf{x}) = q^*(\mathbf{x}) = g^*(\mathbf{x}) \le 0$. Conversely, for $(\mathbf{x}, \mathbf{z}_1, w_1, \ldots, \mathbf{z}_m, w_m)$ feasible in NLP (8.11), we have (by weak duality) $g^*(\mathbf{x}) \le q^*(\mathbf{x}) \le 0$, and so $\mathbf{x}$ is feasible in (GSIP). □

As a practical point, the constraints $\|\mathbf{z}_i\|_2 \le w_i$ appearing in NLP (8.11) are not differentiable. To overcome this, they could be replaced with the pair of constraints $\|\mathbf{z}_i\|_2^2 \le w_i^2$, $w_i \ge 0$. Note that such a manipulation applied directly to the SOCP LLP (8.9), in the hope of obtaining an LLP with smooth $\mathbf{h}(\mathbf{x}, \cdot)$ in order to apply Proposition 8.4.1, in general would not preserve the convexity of the new function $\mathbf{h}(\mathbf{x},\cdot)$. Meanwhile, as NLP (8.11) is already nonconvex in general, such a reformulation has no downside.

8.4.4 General cone program LLP

Another class of convex programs that satisfy a strong duality property is the class of cone programs; see §4.6.1 of [33] and Ch. 4 of [134]. This is a very broad class encompassing LP, convex QP, SOCP, and semi-definite programs (SDP). Under the proper assumptions, the dual problem of a cone program is also a cone program and strong duality holds.

Thus, reformulation of (GSIP) when (LLP) is a cone program is possible; the issue is that the reformulation, while a finite NLP, involves generalized inequalities induced by the defining cone. For instance, the constraints $\|\mathbf{z}_i\|_2 \le w_i$ in the dual SOCP (8.10) express the constraint that $(\mathbf{z}_,w_i) \in K_{SOC,i}$, where $K_{SOC,i}$ is the second-order cone in $\mathbb{R}^{n_i+1}$. It happens that this constraint is not too difficult to handle; similarly if the defining cone is polyhedral, as in LP, the reformulation leads to practical numerical solution methods.

In contrast, if (LLP) is an SDP, the defining cone is the cone of positive semi-definite matrices. The dual is also an SDP, and as a result, the reformulation of (GSIP) would involve matrix inequalities. In general, these would be nonlinear matrix inequality constraints as well. General-purpose deterministic global optimization software typically cannot handle these constraints. Without more structure, global solution of the reformulation is not numerically possible at present.

For these same reasons, the case when the LLP is a (possibly nonconvex) quadratic program with a single quadratic constraint is not considered. Although this class of problems, despite being nonconvex, satisfies a strong duality property (see Appendix B of [33]), the dual is an SDP and the same issues arise.

8.4.5 Connections to robust optimization

Consider the following "uncertain" optimization problem with a polyhedral feasible set:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \; & f(\mathbf{x}) \\ \text{s.t. } & \mathbf{M}\mathbf{x} \le \mathbf{d}, \end{aligned} \tag{8.12}$$

where the data defining the feasible set are uncertain; i.e. all that is known is that $(\mathbf{M}, \mathbf{d}) \in U$ for some $U \subset \mathbb{R}^{p \times n} \times \mathbb{R}^{p}$. The "robust" counterpart of problem (8.12) is

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \; & f(\mathbf{x}) \\ \text{s.t.}\, & \mathbf{M}\mathbf{x} \leq \mathbf{d}, \quad \forall (\mathbf{M}, \mathbf{d}) \in U. \end{aligned} \tag{8.13}$$

For illustrative purposes, assume that there is only one constraint and so $\mathbf{M} = \mathbf{m}^{\mathrm{T}}$. In the case that U is an ellipsoidal set, the robust counterpart (8.13) is an SIP with an SOCP LLP. To see this, assume that $U = \{\mathbf{y} \in \mathbb{R}^{n+1} : \|\mathbf{A}(\mathbf{y} - \mathbf{y}_0)\|_2 \leq 1\}$ for some invertible $\mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)}$ and $\mathbf{y}_0 \in \mathbb{R}^{n+1}$ (in this case $U = \{\mathbf{y}_0 + \mathbf{A}^{-1}\mathbf{y}' : \|\mathbf{y}'\|_2 \leq 1\}$, i.e. the image of the unit ball under an affine mapping). Let $\mathbf{c} : \mathbf{x} \mapsto (\mathbf{x}, -1)$. Then the robust counterpart becomes

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \; & f(\mathbf{x}) \\ \text{s.t.}\, & \mathbf{m}^{\mathrm{T}}\mathbf{x} - d = \mathbf{c}(\mathbf{x})^{\mathrm{T}}\mathbf{y} \leq 0, \quad \forall (\mathbf{m}, d) \equiv \mathbf{y} : \|\mathbf{A}(\mathbf{y} - \mathbf{y}_0)\|_2 \leq 1, \end{aligned}$$

which is indeed an SIP with an SOCP LLP. We can apply Theorem 8.4.3 to obtain

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}, w} \; & f(\mathbf{x}) \\ \text{s.t.}\, & -\mathbf{z}^{\mathrm{T}}\mathbf{A}\mathbf{y}_0 + w \leq 0, \\ & \mathbf{z}^{\mathrm{T}}\mathbf{A} = (-\mathbf{x}, 1)^{\mathrm{T}}, \\ & \|\mathbf{z}\|_2 \leq w, \\ & (\mathbf{x}, \mathbf{z}, w) \in \mathbb{R}^n \times \mathbb{R}^{n+1} \times \mathbb{R}. \end{aligned}$$

If f is affine, this program is also an SOCP.

Uncertain convex programs and their robust counterpart have been considered in [18, 19]. These papers focus on the cases when the robust counterpart has a convex reformulation, as in the example above. When f is nonconvex, or $\mathbf{c}$ is nonaffine (allowing us to potentially construct uncertain programs with non-polyhedral feasible sets), Theorem 8.4.3 provides a way to reformulate the robust counterpart as a finite program. Similarly, when the uncertainty U is polyhedral (for instance, an interval), Theorem 8.4.2 potentially provides a way to reformulate the robust counterparts.

8.5 Numerical method

The goal of this section is to apply the SIP solution method of [122] and discuss the assumptions necessary for the method to converge to the global solution of the reformulation (SIP), and thus to the global solution of the original (GSIP).

The method in [122] proceeds by iteratively solving NLPs which furnish lower and upper bounds on the global optimal value of an SIP. The upper bound is always evaluated at some SIP feasible point, and so upon finite termination one has a feasible point which yields an objective value within some tolerance of the global optimum. Considering (SIP), subproblems at a specific iteration are the lower bounding problem (LBP) for a finite subset $Y^{LBP} \subset Y$

$$\begin{aligned} f^*_{LBP} = \inf_{(\mathbf{x},\boldsymbol{\mu}) \in X \times M} & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x},\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in Y^{LBP}, \\ & \boldsymbol{\mu} \geq \mathbf{0}, \end{aligned} \tag{LBP}$$

the upper bounding problem (UBP) for a finite subset $Y^{UBP} \subset Y$ and $\epsilon_R > 0$

$$\begin{aligned} f^*_{UBP} = \inf_{(\mathbf{x},\boldsymbol{\mu}) \in X \times M} & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x},\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) \leq -\epsilon_R, \quad \forall \mathbf{y} \in Y^{UBP}, \\ & \boldsymbol{\mu} \geq \mathbf{0}, \end{aligned} \tag{UBP}$$

and evaluation of (8.1), the dual function q at given $(\mathbf{x},\boldsymbol{\mu})$, which coincides with the lower-level program of (SIP). Note that the constraints of the subproblems (LBP) and (UBP), although nonlinear, are inequality constraints, and no "irregularity" has been introduced by the reformulation as an SIP. Similarly, evaluation of the dual function q is a convex program under the assumptions of Lemma 8.3.2 and when $\boldsymbol{\mu}$ is nonnegative. Thus, these subproblems are amenable to solution with many available solvers. With these definitions, the algorithm for solution of (GSIP) is given in Algorithm 7, which is in essence Algorithm 2.1 in [122] adapted for the current problem and notation. However, Algorithm 7 is more explicit in its use of approximate solutions to the various subproblems and pays closer attention to the interplay of tolerances.

Algorithm 7 Solution method for (GSIP)

```
Require: $\delta_a > 0$, $\delta_r \in (0, 0.2]$, $\epsilon_{atol} \geq 5\delta_a$, $\epsilon_{rtol} \geq 5\delta_r$, $\epsilon_{R,0} > 0$, $r > 1$, $Y^{LBP,0} \subset Y$, $Y^{UBP,0} \subset Y$
  Set $\epsilon_{tol} = \epsilon_{atol}$, $f^{LBD} = -\infty$, $f^{UBD} = +\infty$, $Y^{LBP} = Y^{LBP,0}$, $Y^{UBP} = Y^{UBP,0}$, $\epsilon_R = \epsilon_{R,0}$.

  while $f^{UBD} - f^{LBD} > \epsilon_{tol}$ do
    Calculate $f^{LBD}$ and $(\bar{\mathbf{x}}, \bar{\boldsymbol{\mu}})$ feasible in (LBP) such that $f^{LBD} \leq f^*_{LBP} \leq f(\bar{\mathbf{x}})$ and $f(\bar{\mathbf{x}}) - f^{LBD} \leq \max\{\delta_a, \delta_r\, |f(\bar{\mathbf{x}})|\}$.
    Determine: that $q(\bar{\mathbf{x}}, \bar{\boldsymbol{\mu}}) \leq 0$, or calculate $\bar{\mathbf{y}} \in Y$ such that $g(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \bar{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) > 0$.
    if $q(\bar{\mathbf{x}}, \bar{\boldsymbol{\mu}}) \leq 0$ then
      $f^{UBD} \leftarrow f(\bar{\mathbf{x}})$, $\mathbf{x}^* \leftarrow \bar{\mathbf{x}}$
      return $\mathbf{x}^*$.
    else
      $Y^{LBP} \leftarrow \{\bar{\mathbf{y}}\} \cup Y^{LBP}$
    end if
    Determine feasibility of (UBP).
    if (UBP) is infeasible then
      $\epsilon_R \leftarrow \epsilon_R / r$
    else
      Calculate $\underline{f}_{UBP}$ and $(\bar{\mathbf{x}}, \bar{\boldsymbol{\mu}})$ feasible in (UBP) such that $\underline{f}_{UBP} \leq f^*_{UBP} \leq f(\bar{\mathbf{x}})$ and $f(\bar{\mathbf{x}}) - \underline{f}_{UBP} \leq \max\{\delta_a, \delta_r\, |f(\bar{\mathbf{x}})|\}$.
      Determine: that $q(\bar{\mathbf{x}}, \bar{\boldsymbol{\mu}}) \leq 0$, or calculate $\bar{\mathbf{y}} \in Y$ such that $g(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \bar{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) > 0$.
      if $q(\bar{\mathbf{x}}, \bar{\boldsymbol{\mu}}) \leq 0$ then
        if $f(\bar{\mathbf{x}}) \leq f^{UBD}$ then
          $f^{UBD} \leftarrow f(\bar{\mathbf{x}})$, $\mathbf{x}^* \leftarrow \bar{\mathbf{x}}$
        end if
        $\epsilon_R \leftarrow \epsilon_R / r$
      else
        $Y^{UBP} \leftarrow \{\bar{\mathbf{y}}\} \cup Y^{UBP}$
      end if
    end if
    if $f^{UBD} < +\infty$ then
      $\epsilon_{tol} \leftarrow \max\{\epsilon_{atol}, \epsilon_{rtol}|f^{UBD}|\}$
    end if
  end while
  return $\mathbf{x}^*$.
```

The rest of this section is devoted to establishing conditions on the original GSIP that ensure that Algorithm 7 converges to the global solution of (GSIP). A few assumptions that will be critical are stated first.

Assumption 8.5.1. *Assume that X and $Y = D_y$ are compact, and that f, g, and $\mathbf{h}$ are continuous.*

Assumption 8.5.2. *For given $\epsilon_f > 0$, assume that there exists a point $\mathbf{x}_S \in X$ such that $\mathbf{x}_S$ is an ϵ_f-optimal GSIP Slater point for* (GSIP)*; i.e.*

$$f(\mathbf{x}_S) \leq f^* + \epsilon_f \quad \text{and} \quad g(\mathbf{x}_S, \mathbf{y}) < 0, \quad \forall \mathbf{y} \in \widehat{Y}(\mathbf{x}_S).$$

First, it is easy to establish that a GSIP Slater point implies the existence of an SIP Slater point.

Proposition 8.5.1. *For some $M \subset \mathbb{R}^m$, let Assumption 8.3.1 hold. For some $\epsilon_f > 0$, let Assumption 8.5.2 hold. Assume $Y = D_y$ is compact and that $g(\mathbf{x}^S, \cdot)$ and $\mathbf{h}(\mathbf{x}^S, \cdot)$ are continuous. Then there exists $\boldsymbol{\mu}_S \in M$ and $\epsilon_S > 0$ such that*

$$f(\mathbf{x}_S) \leq f^* + \epsilon_f, \quad \boldsymbol{\mu}_S \geq \mathbf{0}, \quad \text{and} \quad g(\mathbf{x}_S, \mathbf{y}) - \boldsymbol{\mu}_S^{\mathrm{T}} \mathbf{h}(\mathbf{x}_S, \mathbf{y}) \leq -\epsilon_S, \quad \forall \mathbf{y} \in Y$$

(i.e. $(\mathbf{x}_S, \boldsymbol{\mu}_S)$ is an SIP Slater point for (SIP)*).*

Proof. Since $\mathbf{h}(\mathbf{x}_S, \cdot)$ is continuous and Y is compact, $\widehat{Y}(\mathbf{x}_S)$ is compact. Thus, since $g(\mathbf{x}_S, \cdot)$ is continuous and negative on $\widehat{Y}(\mathbf{x}_S)$, $g^*(\mathbf{x}_S) = -\epsilon_S$ for some $\epsilon_S > 0$. Then, by Assumption 8.3.1, there exists a $\boldsymbol{\mu}_S \in M$, $\boldsymbol{\mu}_S \geq \mathbf{0}$ such that

$$-\epsilon_S = q(\mathbf{x}_S, \boldsymbol{\mu}_S) = \sup\{g(\mathbf{x}_S, \mathbf{y}) - \boldsymbol{\mu}_S^{\mathrm{T}} \mathbf{h}(\mathbf{x}_S, \mathbf{y}) : \mathbf{y} \in Y\}.$$

Consequently,

$$g(\mathbf{x}_S, \mathbf{y}) - \boldsymbol{\mu}_S^{\mathrm{T}} \mathbf{h}(\mathbf{x}_S, \mathbf{y}) \leq -\epsilon_S, \quad \forall \mathbf{y} \in Y.$$

□

We can now establish the finite convergence of Algorithm 7 to a feasible, epsilon-optimal solution of the original (GSIP).

Theorem 8.5.1. *For some bounded $M \subset \mathbb{R}^m$, let Assumption 8.3.1 hold. Let Assumption 8.5.1 hold, and for $\epsilon_f > 0$ let Assumption 8.5.2 hold. Then for inputs $\delta_a \geq \epsilon_f$, $\delta_r \in (0, 0.2]$, $\epsilon_{atol} \geq 5\delta_a$, $\epsilon_{rtol} \geq 5\delta_r$, $\epsilon_{R,0} > 0$, $r > 1$, $Y^{LBP,0} \subset Y$, $Y^{UBP,0} \subset Y$, Algorithm 7 terminates finitely with a point $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f^* + \epsilon_{tol}$, where $\mathbf{x}^*$ is feasible in* (GSIP).

Proof. Since M is bounded, $X \times \overline{M}$ is compact. Further, $L : X \times \overline{M} \times Y \to \mathbb{R}$ defined by $L : (\mathbf{x}, \boldsymbol{\mu}, \mathbf{y}) \mapsto g(\mathbf{x}, \mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y})$ is continuous. By Proposition 8.5.1, an SIP Slater point exists, and by Theorem 8.3.1, this SIP Slater point is also ϵ_f-optimal for (SIP). We will establish that in finite iterations Algorithm 7 produces a point $\mathbf{x}^*$ feasible and ϵ_{tol}-optimal in (SIP), and thus also feasible and ϵ_{tol}-optimal in (GSIP). (By a point "$\mathbf{x}^*$ feasible in (SIP)" it is meant that there exists a $\boldsymbol{\mu}^* \geq \mathbf{0}$, $\boldsymbol{\mu}^* \in M$, such that $(\mathbf{x}^*, \boldsymbol{\mu}^*)$ is feasible in (SIP).)

Algorithm 7 can terminate in one of two ways: the difference between the upper and lower bounds ($f^{UBD} - f^{LBD}$) is less than a certain tolerance or the solution of (LBP) is feasible in (SIP).

Consider the first case: the termination condition

$$f^{UBD} - f^{LBD} \leq \epsilon_{tol} = \max\{\epsilon_{atol}, \epsilon_{rtol} \left|f^{UBD}\right|\} \tag{8.14}$$

is satisfied. By the definition of the upper and lower bounds we have $f^{UBD} = f(\mathbf{x}^*)$ and $f^{LBD} \leq f^*_{SIP}$ (since f^{LBD} is a lower bound for f^*_{LBP} and (LBP) is always a relaxation of (SIP)). Thus, $f(\mathbf{x}^*) \leq f^*_{SIP} + \epsilon_{tol}$ (and the point $\mathbf{x}^*$ is always feasible in (SIP)). Consequently, it remains to show that the required difference between the upper and lower bounds can be achieved in finite iterations. By Lemma 2.4 in [122], in finite iterations we have $f^*_{UBP} \leq f^*_{SIP} + \epsilon_f$. Further, the corresponding approximate solution of (UBP), $\mathbf{x}^*$, is feasible in (SIP). By Lemma 2.2 in [122], in finite iterations we have $f^*_{SIP} - (1/3)\epsilon_f \leq f^*_{LBP}$. Thus, at some iteration, we have

$$f^*_{UBP} - f^*_{LBP} \leq \frac{4}{3}\epsilon_f.$$

Let the corresponding upper bound of (UBP) be f^{UBD} (which equals $f(\mathbf{x}^*)$ for some $\mathbf{x}^*$ feasible in (SIP)) and the corresponding lower bound and upper bound of (LBP) be f^{LBD} and $\overline{f}_{LBP}$, respectively. By the construction of these values (i.e. by the termination criteria for the subproblems) we have

$$f^{UBD} - f^*_{UBP} \leq \max\{\delta_a, \delta_r \left|f^{UBD}\right|\},$$
$$f^*_{LBP} - f^{LBD} \leq \max\{\delta_a, \delta_r \left|\overline{f}_{LBP}\right|\},$$

which subsequently yields

$$f^{UBD} - f^{LBD} \leq \max\{\delta_a, \delta_r \left|f^{UBD}\right|\} + \max\{\delta_a, \delta_r \left|\overline{f}_{LBP}\right|\} + \frac{4}{3}\epsilon_f. \tag{8.15}$$

The challenge now is to go through the various cases and show that Inequality (8.15) implies the termination criterion (8.14).

Case 1: $\delta_a \geq \delta_r \left|\overline{f}_{LBP}\right|$.

Using $\epsilon_f \leq \delta_a$ we have $f^{UBD} - f^{LBD} \leq \max\{\delta_a, \delta_r \left|f^{UBD}\right|\} + \frac{7}{3}\delta_a$.

Case 1a: $\delta_a \geq \delta_r \left|f^{UBD}\right|$.

We have $f^{UBD} - f^{LBD} \leq \frac{10}{3}\delta_a \leq 5\delta_a = \max\{5\delta_a, 5\delta_r \left|f^{UBD}\right|\}$ which implies (8.14).

Case 1b: $\delta_a < \delta_r \left|f^{UBD}\right|$.

We have $f^{UBD} - f^{LBD} \leq \frac{10}{3}\delta_r \left|f^{UBD}\right| \leq \max\{5\delta_a, 5\delta_r \left|f^{UBD}\right|\}$ which implies (8.14).

Case 2: $\delta_a < \delta_r \left|\overline{f}_{LBP}\right|$.

First, by the termination criterion of (LBP) and the reverse triangle inequality, we have $\left|\overline{f}_{LBP}\right| - \left|f^{LBD}\right| \leq \left|\overline{f}_{LBP} - f^{LBD}\right| \leq \max\{\delta_a, \delta_r \left|\overline{f}_{LBP}\right|\} = \delta_r \left|\overline{f}_{LBP}\right|$. By the assumptions on δ_r, we can rearrange to get $\left|\overline{f}_{LBP}\right| \leq 1/(1-\delta_r) \left|f^{LBD}\right|$. Thus

$$f^{UBD} - f^{LBD} \leq \max\{\delta_a, \delta_r \left|f^{UBD}\right|\} + \frac{\delta_r}{(1-\delta_r)} \left|f^{LBD}\right| + \frac{4}{3}\epsilon_f. \tag{8.16}$$

Case 2a: $\delta_a \geq \delta_r \left|f^{UBD}\right|$.

Using $\epsilon_f \leq \delta_a$ we have $f^{UBD} - f^{LBD} \leq (7/3)\delta_a + \frac{\delta_r}{(1-\delta_r)} \left|f^{LBD}\right|$. The reverse triangle inequality gives $\left|f^{LBD}\right| - \left|f^{UBD}\right| \leq \left|f^{UBD} - f^{LBD}\right| \leq (7/3)\delta_a + \frac{\delta_r}{(1-\delta_r)} \left|f^{LBD}\right|$. Again, by the assumptions on δ_r, we can rearrange this to get

$$\left|f^{LBD}\right| \leq \frac{1-\delta_r}{1-2\delta_r}(\left|f^{UBD}\right| + (7/3)\delta_a).$$

Using this in Inequality (8.16), we get $f^{UBD} - f^{LBD} \leq (7/3)\delta_a + \frac{\delta_r}{1-2\delta_r}(|f^{UBD}| + (7/3)\delta_a)$. Since $\delta_r \leq 0.2$, we have ${\delta_r}/(1-2\delta_r) \leq (5/3)\delta_r$. Then we have $f^{UBD} - f^{LBD} \leq (7/3)\delta_a + (5/3)\delta_r |f^{UBD}| + (5/3)(7/3)\delta_r\delta_a$. Using the fact that $\delta_r \leq 0.2$ and the assumption of this case ($\delta_a \geq \delta_r |f^{UBD}|$), we get $f^{UBD} - f^{LBD} \leq (4 + 7/9)\delta_a \leq 5\delta_a = \max\{5\delta_a, 5\delta_r |f^{UBD}|\}$ which implies (8.14).

Case 2b: $\delta_a < \delta_r |f^{UBD}|$.

Similarly to the previous case, we obtain from (8.16): $|f^{LBD}| \leq \frac{1-\delta_r}{1-2\delta_r}(|f^{UBD}| + (7/3)\delta_r |f^{UBD}|)$. Using this in Inequality (8.16), we get

$$f^{UBD} - f^{LBD} \leq (7/3)\delta_r |f^{UBD}| + \frac{\delta_r}{1-2\delta_r}(|f^{UBD}| + (7/3)\delta_r |f^{UBD}|).$$

As before, ${\delta_r}/(1-2\delta_r) \leq (5/3)\delta_r$, so

$$f^{UBD} - f^{LBD} \leq (7/3 + 5/3 + (5/3)(7/3)\delta_r)\delta_r |f^{UBD}|.$$

Using the fact that $\delta_r \leq 0.2$ we get $f^{UBD} - f^{LBD} \leq (4 + 7/9)\delta_r |f^{UBD}| \leq \max\{5\delta_a, 5\delta_r |f^{UBD}|\}$, which implies (8.14).

Now, consider the the other case for termination: (LBP) produces a point $\mathbf{x}^*$ feasible in (SIP). It follows that $f(\mathbf{x}^*)$ is a valid upper bound for f^*_{SIP}, and indeed we set $f^{UBD} = f(\mathbf{x}^*)$. But from the termination criterion for (LBP), we have $f^{UBD} - f^{LBD} = f(\mathbf{x}^*) - f^{LBD} \leq \max\{\delta_a, \delta_r |f(\mathbf{x}^*)|\}$. Letting $\overline{f}_{LBP} = f(\mathbf{x}^*)$, the analysis proceeding from Inequality (8.15) can be reused. □

As a final note, Assumptions 8.3.1 and 8.5.1 are fairly easy to verify for a specific problem. On the other hand, Assumption 8.5.2 is more difficult to verify. However, Assumption 8.5.2 is required only to *guarantee* finite termination of Algorithm 7. Similar to the discussion in §2.2 of [122], it is easy to see that if Algorithm 7 converges finitely, then the solution provided is global optimal to a certain tolerance. That is to say, without Assumption 8.5.2, we do not need to worry that finite termination of Algorithm 7 furnishes a suboptimal or infeasible solution.

8.6 Examples

Numerical experiments are considered in this section. The example in §8.6.2 considers (GSIP) when Assumption 8.4.1 holds (when the LLP is linear). This example demonstrates that the duality-based reformulation (8.7) can significantly reduce the computational effort compared to the KKT-based reformulation (8.8). The example in §8.6.3 also considers a GSIP with a linear LLP, but in this example the LLP is infeasible for certain values of $\mathbf{x}$. This example clearly establishes that the duality-based reformulation is qualitatively different from the KKT-based reformulation. Section 8.6.4 considers a portfolio optimization problem which yields a GSIP with nonlinear LLP and applies three applicable reformulations. Section 8.6.5 also considers a GSIP with nonlinear LLP; the flexibility of Algorithm 7 is demonstrated.

8.6.1 Methods

All numerical studies are performed on a 64-bit Linux virtual machine allocated a single core of a 3.07 GHz Intel Xeon processor and 1.28 GB RAM. The studies are performed in GAMS version 24.3.3 [56]. Deterministic global optimizers BARON version 14.0.3 [197, 159] and ANTIGONE version 1.1 [120] are employed. Examples that require it use an implementation of Algorithm 7 which is based on an implementation of Algorithm 2.1 in [122] by Alexander Mitsos. This implementation is coded in GAMS, employing BARON for the solution of (LBP), (UBP), and the dual function (8.1) unless otherwise noted. Unless otherwise noted the parameters are $\epsilon_{R,0} = 1$, $r = 2$, $Y^{LBP,0} = Y^{UBP,0} = \varnothing$, $\epsilon_{atol} = 5\delta_a$, and $\epsilon_{rtol} = 5\delta_r$.

8.6.2 Computation times for linear LLP

Under Assumption 8.4.1, consider an instance of (GSIP) where $n_y \in \mathbb{N}$, $n_x = 3n_y$, $\widetilde{X} = [-10, 10]^{2n_y} \times [-1, 1]^{n_y}$, $X = \{\mathbf{x} \in \widetilde{X} : x_{n_y+i} \geq x_i, \forall i \in \{1, \dots, n_y\}\}$,

$$\begin{aligned} d &: \mathbf{x} \mapsto 0, \\ \mathbf{c} &: \mathbf{x} \mapsto (x_{2n_y+1}, \dots, x_{3n_y}), \\ \mathbf{b} &: \mathbf{x} \mapsto (-x_1, \dots, -x_{n_y}, x_{n_y+1}, \dots, x_{2n_y}), \end{aligned} \qquad \mathbf{A} : \mathbf{x} \mapsto \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix},$$

and $f : \mathbf{x} \mapsto -\prod_{i=1}^{n_y}(x_{n_y+i} - x_i)$, where $\mathbf{I}$ is the $n_y \times n_y$ identity matrix. For a geometric interpretation, we can think of this problem as related to a design centering problem in

Table 8.1: Solution of GSIP with linear LLP from §8.6.2 for various sizes n_y; the "$*$" indicates that the desired optimality tolerances were not satisfied by the end of the time reported.

	$n_y = 4$	$n_y = 6$	$n_y = 8$	$n_y = 10$
Duality-based reformulation (8.7), CPU time (s)	0.06	0.20	0.26	0.08
KKT-based reformulation (8.8), CPU time (s)	0.62	11.5	373	1200 ($*$)

n_y dimensions, in which a maximum volume box ($\widehat{Y}(\mathbf{x}) = [x_1, x_{n_y+1}] \times \cdots \times [x_{n_y}, x_{2n_y}]$) is desired, except that the normal of the vector defining the linear infinite constraint is a function of the upper-level variables $\mathbf{x}$. When $n_y = 2$, one possible solution is $\mathbf{x} = (-10, -10, 0, 10, 1, 0)$. In general, the optimal objective value is $-10 \times (20)^{n_y - 1}$.

Since the lower-level program of this problem always has a solution (for all $\mathbf{x} \in X$, the feasible set is bounded and nonempty), so does its dual LP (8.6), and thus the duality-based reformulation as the NLP (8.7) given by Theorem 8.4.2 is applicable. Similarly, the KKT-based reformulation in (8.8) can also be applied. These NLP reformulations are solved with BARON, with relative and absolute optimality tolerances both equal to 10^{-5}. Results are summarized in Table 8.1 for various n_y.

The duality-based reformulation can be solved in less than a second for each problem size considered. Meanwhile, the solution time of the KKT-based reformulation grows rapidly with the size of the problem, and when $n_y = 10$, there is still optimality gap of approximately 100% at the end of the time allotted (20 minutes).

8.6.3 Infeasible linear LLP

Again, consider a GSIP with a linear LLP defined by $n_y = 2$, $n_x = 4$, $X = [-10, 10]^4$,

$$\begin{aligned} d &: \mathbf{x} \mapsto 0, \\ \mathbf{c} &: \mathbf{x} \mapsto (-1, -1), \\ \mathbf{b} &: \mathbf{x} \mapsto (-x_1, -x_2, x_3, x_4, 0), \end{aligned} \qquad \mathbf{A} : \mathbf{x} \mapsto \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

and $f : \mathbf{x} \mapsto -(x_4 - x_2)(x_3 - x_1)$. This is related to a two-dimensional design centering problem, in which a maximum volume box $[x_1, x_3] \times [x_2, x_4]$ is sought. However, there is an extra constraint in the definition of the set $\widehat{Y}(\mathbf{x})$ so that it is empty if $x_1 > 0$. Of course, this means that such $\mathbf{x}$ are feasible in the GSIP. It so happens that the infimum of this problem is not achieved. Note that the sequence $\mathbf{x}_k = (x_{1,k}, -10, 10, 10)$ with $x_{1,k}$ decreasing to 0 is feasible and $f(\mathbf{x}_k) \to -200$; however $\mathbf{x} = (0, -10, 10, 10)$ is infeasible since the LLP is feasible and its feasible set contains a point $\mathbf{y}$ (for instance, $\mathbf{y} = (0, -1)$) such that $\mathbf{c}(\mathbf{x})^{\mathrm{T}}\mathbf{y} > 0$.

Despite this, within some tolerance we can approximate this optimal objective value with a feasible point by solving the duality-based NLP reformulation. The LLP has a solution for some $\mathbf{x} \in X$, thus so does the dual LP (8.6), and so for this $\mathbf{x}$ the dual feasible set is nonempty. But since $\mathbf{A}$ and $\mathbf{c}$ are constant, the dual feasible set is always nonempty and so Theorem 8.4.2 applies. Solving the duality-based reformulation with BARON with relative and absolute optimality tolerances both equal to 10^{-5}, the solution found is $\mathbf{x}^* = (0.00003, -10, 10, 10)$ (which solves in less than a tenth of a second).

However, solving the KKT-based reformulation with BARON with the same tolerances yields the solution $\mathbf{x}^* = (0, 0, 10, 10)$. This is a completely different answer; the point is indeed feasible in the original GSIP, but it is clearly suboptimal. This demonstrates that one must be careful when applying the KKT-based reformulation, and that in general the duality-based reformulation is more flexible for defining an equivalent problem, and still amenable to approximate numerical solution.

8.6.4 Portfolio optimization

The following problem is Problem 7 from §5.2 in [184]. The problem is related to a portfolio optimization problem; a fixed amount of capital (for simplicity, taken to be one dollar) is to be invested among N shares. The i^{th} share at the end of some period has some return y_i. The objective is to maximize the portfolio value at the end the the period. Of course, there is some uncertainty in the return values, and in addition there is some aversion to straying too far from investing equally in all shares. For a more detailed description and related problems, see [184]. The mathematical program is the following GSIP:

$$\begin{aligned}
&\max_{\mathbf{x},r} r && (8.17)\\
&\text{s.t. } r - \mathbf{y}^{\mathrm{T}}\mathbf{x} \le 0, \quad \forall \mathbf{y} \in \widehat{Y}(\mathbf{x}),\\
&\quad \mathbf{1}^{\mathrm{T}}\mathbf{x} = 1, \quad \mathbf{x} \ge \mathbf{0},\\
&\quad (\mathbf{x}, r) \in [\mathbf{0}, \mathbf{1}] \times \mathbb{R},\\
&\quad \widehat{Y}(\mathbf{x}) = \left\{ \mathbf{y} \in \mathbb{R}^N : \sum_{i=1}^{N} (y_i - \bar{y}_i)^2 \le \Theta(\mathbf{x})^2 \right\},
\end{aligned}$$

where

$$\begin{aligned}
&\Theta : \mathbf{x} \mapsto 1.5 \left(1 + \sum_{i=1}^{N} \left(x_i - {}^{1}\!/_{N}\right)^2 \right),\\
&\bar{y}_i = 1.15 + i\left({}^{0.05}\!/_{N}\right).
\end{aligned}$$

Three solution approaches are discussed: Algorithm 7 is applied to the SIP reformulation from Theorem 8.3.1; BARON and ANTIGONE are applied to the NLP reformulation (8.3) from Proposition 8.4.1; and BARON and ANTIGONE are applied as well to the NLP reformulation (8.11) from Theorem 8.4.3. For Algorithm 7, let $\delta_a = \delta_r = 2 \times 10^{-5}$, giving overall relative and absolute optimality tolerances equal to 10^{-4}. Relative and absolute optimality tolerances for BARON and ANTIGONE applied to the NLP reformulations are both 10^{-4}.

SIP reformulation

First consider the SIP reformulation. Take $g : (\mathbf{x}, r, \mathbf{y}) \mapsto r - \mathbf{y}^{\mathrm{T}}\mathbf{x}$ and $h_1 : (\mathbf{x}, r, \mathbf{y}) \mapsto \sum_{i=1}^{N} (y_i - \bar{y}_i)^2 - \Theta(\mathbf{x})^2$. Then the LLP is a smooth convex program

$$g^*(\mathbf{x}, r) = \sup \left\{ r - \mathbf{y}^{\mathrm{T}}\mathbf{x} : \sum_{i=1}^{N} (y_i - \bar{y}_i)^2 \le \Theta(\mathbf{x})^2 \right\},$$

which achieves its supremum for all $(\mathbf{x}, r)$ and for which $\bar{\mathbf{y}}$ is a Slater point for all $\mathbf{x}$ (since Θ is bounded below by 1.5). Next, compact X, Y, and M are required. Since each $x_i \in [0, 1]$, $\Theta(\mathbf{x})$ is bounded above by $1.5(1 + N)$ for all $\mathbf{x} \in [\mathbf{0}, \mathbf{1}]$. Consequently, $\widehat{Y}(\mathbf{x}) \subset [\bar{\mathbf{y}} - (1.5(1 + N))\mathbf{1}, \bar{\mathbf{y}} + (1.5(1 + N))\mathbf{1}]$ for all $\mathbf{x} \in [\mathbf{0}, \mathbf{1}]$. Estimating further (noting that

$\bar{\mathbf{y}} \in [(1.15)\mathbf{1}, (1.2)\mathbf{1}]$), let $y^L(N) = 1.15 - 1.5(1+N)$ and $y^U(N) = 1.2 + 1.5(1+N)$ and take $Y = [(y^L(N))\mathbf{1}, (y^U(N))\mathbf{1}]$ which is a superset of $\widehat{Y}(\mathbf{x})$ for all $\mathbf{x}$ (thus we can intersect the feasible set of the LLP with this Y and not change the optimal value). Then by Lemma 8.3.2, strong duality holds for all $\mathbf{x}$.

For $(\mathbf{x}, r)$ feasible in problem (8.17), we have $r \leq \mathbf{y}^{\mathrm{T}}\mathbf{x}$ for all $\mathbf{y} \in \widehat{Y}(\mathbf{x})$, and so $r \leq \mathbf{y}^{\mathrm{T}}\mathbf{x}$ for all $\mathbf{y} \in Y$. Since $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{1}^{\mathrm{T}}\mathbf{x} = 1$, an upper bound for feasible r is $y^U(N)$. Similarly, a lower bound is $y^L(N)$ (specifically, for any $(\mathbf{x}, r)$ optimal for problem (8.17), $r > y^L(N)$). Thus, take $X = \{\mathbf{x} : \mathbf{x} \geq \mathbf{0}, \mathbf{1}^{\mathrm{T}}\mathbf{x} = 1\} \times [y^L(N), y^U(N)]$. Then $g(\mathbf{x}, r, \bar{\mathbf{y}}) \geq y^L(N) - \bar{\mathbf{y}}^{\mathrm{T}}\mathbf{x} \geq y^L(N) - 1.2(\mathbf{1}^{\mathrm{T}}\mathbf{x}) = y^L(N) - 1.2$ for all $(\mathbf{x}, r) \in X$. An upper bound for h_1 evaluated at a Slater point is $-(1.5)^2$. Then by Lemma 8.3.3 take $M = [0, {}^{(y^L(N) - 1.2)}/_{2.25}]$. It is clear that Assumption 8.5.1 holds.

General LLP reformulation

Next, the reformulation from Proposition 8.4.1 is applied. For the purposes of applying this reformulation, take $D_y = \mathbb{R}^N$ and let Y equal the interior of its previous definition (i.e. the interior of $\left[(y^L(N))\mathbf{1}, (y^U(N))\mathbf{1}\right]$). Then with the definitions of X and M as before, the hypotheses of Proposition 8.4.1 are satisfied and the reformulation of problem (8.17) is

$$\begin{aligned} \max_{\mathbf{x}, r, \mathbf{y}, \mu_1} \; & r \\ \text{s.t.}\; & r - \mathbf{y}^{\mathrm{T}}\mathbf{x} - \mu_1 \left(\sum_{i=1}^{N} (y_i - \bar{y}_i)^2 - \Theta(\mathbf{x})^2 \right) \leq 0, \\ & -\mathbf{x} - \mu_1 2(\mathbf{y} - \bar{\mathbf{y}}) = \mathbf{0}, \\ & (\mathbf{x}, r) \in X, \quad \mathbf{y} \in \overline{Y}, \quad \mu_1 \in M. \end{aligned} \tag{8.18}$$

SOCP LLP reformulation

Now consider the reformulation based on SOCP duality from Theorem 8.4.3. Since Θ is non-negative-valued, the LLP can be written as an SOCP:

$$g^*(\mathbf{x}, r) = \sup\{r - \mathbf{y}^{\mathrm{T}}\mathbf{x} : \|\mathbf{y} - \bar{\mathbf{y}}\|_2 \leq \Theta(\mathbf{x})\}.$$

To satisfy Assumption 8.4.2 explicitly, let $\mathbf{c} : (\mathbf{x}, r) \mapsto -\mathbf{x}$, $s : (\mathbf{x}, r) \mapsto r$, $\mathbf{A}_1 : (\mathbf{x}, r) \mapsto \mathbf{I}$ (the identity), $\mathbf{b}_1 : (\mathbf{x}, r) \mapsto -\bar{\mathbf{y}}$, $\mathbf{e}_1 : (\mathbf{x}, r) \mapsto \mathbf{0}$, and $d_1 : (\mathbf{x}, r) \mapsto \Theta(\mathbf{x})$. As before, $\bar{\mathbf{y}}$

Table 8.2: Solution of portfolio optimization problem (8.17) with $N = 10$ by various methods.

Method	Lower bound	Optimality gap	CPU time (s)
Algorithm 7	0.7033	7×10^{-5}	44.3
NLP Reformulation (8.3), BARON	0.7033	7.25	1200
NLP Reformulation (8.3), ANTIGONE	0.7033	0.13	1200
NLP Reformulation (8.11), BARON	0.7033	10^{-4}	0.06
NLP Reformulation (8.11), ANTIGONE	0.7033	10^{-4}	0.19

satisfies the Slater point assumption of Theorem 8.4.3. The reformulation becomes

$$\begin{aligned} \max_{\mathbf{x},r,\mathbf{z},w} \ & r \\ \text{s.t.}\ & r - \mathbf{z}^{\mathrm{T}}\bar{\mathbf{y}} + w\Theta(\mathbf{x}) \le 0, \\ & \mathbf{z}^{\mathrm{T}} = \mathbf{x}^{\mathrm{T}}, \\ & \|\mathbf{z}\|_2^2 \le w^2, \quad w \ge 0, \\ & (\mathbf{x}, r) \in X, \quad (\mathbf{z}, w) \in \mathbb{R}^N \times \mathbb{R}. \end{aligned} \tag{8.19}$$

Note that the smooth reformulation of the second-order cone constraint has been used. Further, $(\mathbf{z}, w)$ have not been restricted to a compact set. However, it is possible to do so; clearly $\mathbf{z}$ is in a compact set, and since $(\mathbf{z}, r) \mapsto r - \mathbf{z}^{\mathrm{T}}\bar{\mathbf{y}}$ is bounded on X and Θ is positive-valued, we could derive an upper bound for w. It is very likely that BARON and ANTIGONE identify such bounds as part of constraint propagation when they pre-process the problem.

Discussion

First, set $N = 10$. Table 8.2 lists solution statistics. The optimal objective value (or lower bound) agrees with the results of [184] (which uses a local method applied to a KKT-based reformulation). Algorithm 7 applied to the SIP reformulation from Theorem 8.3.1 is fairly successful; although it requires 48 iterations, over which almost 200 NLP subproblems are solved, these subproblems are fairly easy and the overall CPU time is less than a minute to achieve the desired optimality tolerances. The NLP reformulation (8.11) based on SOCP duality from Theorem 8.4.3 is solved very quickly by either BARON or ANTIGONE.

Table 8.3: Solution of SOCP duality-based reformulation (8.19) of portfolio optimization problem (8.17) for various N.

	Lower	BARON		ANTIGONE	
	bound	Optimality gap	CPU time (s)	Optimality gap	CPU time (s)
$N = 10$	0.7033	1.0×10^{-4}	0.06	1.0×10^{-4}	0.19
$N = 20$	0.8411	1.0×10^{-4}	0.46	1.0×10^{-4}	0.53
$N = 50$	0.9638	3.7×10^{-4}	1200	1.0×10^{-4}	3.0
$N = 100$	1.0259	1.3×10^{-2}	1200	1.0×10^{-4}	51.0
$N = 150$	1.0535	5.0×10^{-3}	1200	1.1×10^{-4}	185

Meanwhile, solution of the NLP reformulation (8.3) from Proposition 8.4.1 encounters some difficulty; neither BARON nor ANTIGONE achieve the desired optimality tolerances in the time allotted (20 minutes), although both identify an optimal feasible solution as a candidate. One possible reason for this comes from the observation that there are "worse" nonconvexities appearing in problem (8.18) compared to problem (8.19); for instance, there will be a term of the form $\mu_1 x_1^2 x_2^2$ in the constraints of problem (8.18), while only quadratic terms and trilinear terms of the form wx_i^2 appear in the constraints of problem (8.19).

Using the SOCP duality-based reformulation (8.19), larger instances (larger N) are attempted and reported in Table 8.3. Both BARON and ANTIGONE find an optimal solution as a candidate, although the time required by each to verify that this solution is optimal to the desired tolerance varies. Overall, the performance of ANTIGONE is quite good, and the optimal objective values agree with the results of [184].

8.6.5 Nonlinear LLP

A design centering problem with nonlinear LLP is considered and solved with Algorithm 7. Since the LLP of the following example involves trigonometric functions, BARON cannot be used to calculate the dual function value required by Algorithm 7. However, as mentioned earlier, under the assumptions of Lemma 8.3.2, the dual function (8.1) is defined in terms of a convex program (when $\boldsymbol{\mu} \geq \mathbf{0}$). In Algorithm 7 the dual function is always evaluated at nonnegative $\boldsymbol{\mu}$, and so SNOPT [59] is used instead to evaluate the dual function.

Consider the following design centering problem:

$$\begin{aligned} &\max_{\mathbf{x}\in X} \operatorname{vol}(\widehat{Y}(\mathbf{x})) \\ &\text{s.t. } g(\mathbf{y}) = -\cos(y_1)\sin(y_2) + \frac{y_1}{y_2^2+1} - \sum_i \alpha_i(-y_i)(1-y_i) \le 0, \quad \forall \mathbf{y} \in \widehat{Y}(\mathbf{x}), \end{aligned} \tag{8.20}$$

where $\alpha_1 = 1.841$, $\alpha_2 = 6.841$, $Y = D_y = [-1,1] \times [-1,1]$,

$$\mathbf{h} : (\mathbf{x},\mathbf{y}) \mapsto \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \\ -x_3 + y_1 \\ -x_4 + y_2 \end{bmatrix},$$

$\widehat{Y}(\mathbf{x}) = \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x},\mathbf{y}) \le \mathbf{0}\} = \{\mathbf{y} \in Y : y_1 \in [x_1,x_3], y_2 \in [x_2,x_4]\}$, $\operatorname{vol}(\widehat{Y}(\mathbf{x})) = (x_3 - x_1)(x_4 - x_2)$, and $X = \{\mathbf{x} \in Y \times Y \subset \mathbb{R}^4 : x_3 - x_1 \ge 0.002, x_4 - x_2 \ge 0.002\}$.

It is easy to see that g is twice continuously differentiable on Y. Using this fact we can verify that the Hessian matrix of g is negative semi-definite for all $\mathbf{y} \in Y$. Consequently g is concave on Y (the form of g is inspired by αBB relaxations- see §3.3 of [6]). The rest of the hypotheses of Lemma 8.3.2 also hold, establishing strong duality for the LLP: Y is convex. For each $\mathbf{x}$, $\mathbf{h}(\mathbf{x},\cdot)$ is convex. The midpoint $\mathbf{y}_s(\mathbf{x})$ of $\widehat{Y}(\mathbf{x})$ satisfies $h_i(\mathbf{x},\mathbf{y}_s(\mathbf{x})) \le 0.001$ for each $\mathbf{x}$. $\widehat{Y}(\mathbf{x})$ is nonempty and compact for each $\mathbf{x}$, thus $\sup\{g(\mathbf{y}) : \mathbf{y} \in \widehat{Y}(\mathbf{x})\}$ is finite for each $\mathbf{x}$.

Further, using interval arithmetic we can show that g is bounded below by -1.84148 on Y, thus by Lemma 8.3.3, Assumption 8.3.1 holds for $M = [\mathbf{0}, ({}^{1.84148}\!/\!{}_{0.001})\mathbf{1}]$. Finally, it is clear that Assumption 8.5.1 also holds. Using $\delta_a = \delta_r = 2 \times 10^{-5}$, Algorithm 7 converges in 23 iterations to $\mathbf{x}^* = (-1,-1,1,-0.15103)$. The corresponding CPU time is 4.8s.

For comparison, the NLP reformulation from Proposition 8.4.1 is also constructed and solved (although Y is not open in this example it is plausible that the reformulation still holds). Again, the presence of trigonometric functions precludes the use of the versions of BARON and ANTIGONE included in GAMS version 24.3.3. The solution obtained from solving the NLP reformulation (8.3) locally with SNOPT depends on the starting point. See Table 8.4. Although the local solver is fast in each case (less than a tenth of a second), the quality of the solution obtained varies.

Table 8.4: Starting point and solution obtained from solving NLP reformulation (8.3) of GSIP (8.20) locally with SNOPT. The other components of the starting point are $(\mathbf{y}, \boldsymbol{\mu}) = (\mathbf{0}, \mathbf{0})$.

Starting $\mathbf{x}$	Solution $\mathbf{x}^*$	Relation to GSIP (8.20)
$(-1, -1, 1, 1)$	$(-1, -1, 1, 1)$	infeasible
$(-1, -1, 1, 0.5)$	$(-1, -1, 1, 1)$	infeasible
$(-0.5, -0.5, 0.5, 0.5)$	$(-1, -1, -0.382, 1)$	suboptimal
$(-0.5, -0.5, 0.5, 0)$	$(-1, -1, 1, -0.15098)$	optimal

8.7 Conclusions

This chapter has considered duality-based reformulations of (GSIP) and the practicality of the global solution of these reformulations. A reformulation to a finite NLP was discussed, which avoids the inclusion of complementarity constraints, in contrast with previous results based on the KKT conditions. More specific reformulations are possible, such as when the LLP is an LP or SOCP. In the case of a linear LLP, the reformulation can hold even when the lower-level program is infeasible. Under more general assumptions, it was established that (GSIP) is equivalent to an SIP. These assumptions are easily satisfied when strong duality holds for the lower-level program. A global feasible point method for the solution of SIP was adapted for the solution of (GSIP). The merits of this method were discussed, which include the fact that it involves the iterative solution of simple, tractable, and easily constructed finite NLPs. The computational benefits of the reformulations and solution methods were demonstrated with numerical examples.

Chapter 9

Design centering and robust design in engineering applications

9.1 Introduction

This chapter discusses the theoretical and practical issues involved with solving design centering problems. The problems considered will be in the general form

$$\begin{aligned} \max_{\mathbf{x}} \ & \mathrm{vol}(D(\mathbf{x})) \\ \text{s.t. } & D(\mathbf{x}) \subset G, \\ & \mathbf{x} \in X, \end{aligned} \tag{DC}$$

where $(n_x, n_y, m) \in \mathbb{N}^3$, $Y \subset \mathbb{R}^{n_y}$, $\mathbf{g} : Y \to \mathbb{R}^m$, $G = \{\mathbf{y} \in Y : \mathbf{g}(\mathbf{y}) \leq \mathbf{0}\}$, $X \subset \mathbb{R}^{n_x}$, D is a set-valued mapping $X \rightrightarrows \mathbb{R}^{n_y}$, and vol$(\cdot)$ denotes the "volume" of a set (or some suitable proxy- in this work D will either be ball- or interval-valued, and the choice of volume/proxy will be clear). $D(\mathbf{x})$ is called a "candidate" design space, which is feasible if $D(\mathbf{x})$ is a subset of G, and optimal if it is the "largest" such feasible design space.

Ensuring feasibility of the solution is typically of paramount importance in any method for the solution of (DC). An application of (DC) is to *robust design* problems. In this case, $\mathbf{g}$ represents constraints on a system or process. Given some input parameters $\mathbf{y}$, one desires, for instance, on-specification product, or perhaps more importantly, safe system behavior, indicated by $\mathbf{g}(\mathbf{y}) \leq \mathbf{0}$. In robust design, one seeks a nominal set point $\mathbf{y}_c$ at which to operate the system, and further determine the amount one can deviate from this set point (with respect to some norm) and still have safe process behavior. Then the result is that one seeks a set $D(\mathbf{y}_c, \delta) = \{\mathbf{y} : \|\mathbf{y} - \mathbf{y}_c\| \le \delta\} \subset G$. One goal might be to maximize operational flexibility, in which case the largest $D(\mathbf{y}_c, \delta)$ is sought, i.e. $(\mathbf{y}_c, \delta)$ with the largest δ. This example provides some basic motivation for the focus of this work: The focus on the case that D is ball- or interval-valued comes from the fact that a solution should yield an explicit bound on the maximum acceptable deviation from some nominal set point. The focus on solution methods that are feasible point methods comes from the fact that a solution which violates $D(\mathbf{x}) \subset G$ is not acceptable, especially when safety is concerned.

Under some subtle assumptions (discussed in §9.2), problem (DC) is equivalent to a generalized semi-infinite program (GSIP) expressed as

$$\begin{aligned} &\max_{\mathbf{x}} \operatorname{vol}(D(\mathbf{x})) \\ &\text{s.t. } g_i(\mathbf{y}) \le 0, \quad \forall \mathbf{y} \in D(\mathbf{x}), \quad \forall i \in \{1, \dots, m\}, \\ &\quad \mathbf{x} \in X. \end{aligned} \tag{GSIP}$$

Because design centering problems are a particular instance of GSIP, this work approaches design centering problems from the perspective of and with tools from the GSIP literature. This approach is hardly original [181, 185, 210], however, bringing together these ideas in one work is useful. Further, this work compares different numerical approaches from the GSIP literature. In particular, global optimization methods are considered; as a consequence, challenges and advantages appear that are not present when applying local optimization methods.

The end goal of this work is the case when the constraints of the system $\mathbf{g}$ are implicitly defined by the solution of systems of algebraic or differential equations, as is often the case for robust design in engineering applications. In this case, explicit expressions for $\mathbf{g}$ and its derivatives are, in general, difficult to obtain, and many methods for GSIP require this information in a numerical implementation. Consequently, the focus turns toward approximate solution methods inspired by global, feasible point methods. This discussion, and in particular the challenges in implementing the numerical methods, is original. Some of these approximations come from restrictions of (DC) which are apparent when considering the GSIP reformulation. Other approximations come from terminating a feasible point method early.

Connections to previous work in the literature are pointed out throughout this chapter, which is organized as follows. Section 9.2 discusses some important concepts and the relationship between (DC) and (GSIP), and assumptions that will hold for the rest of this work. Some interesting cases of (DC) and connections to other problems including "flexibility indices" are also discussed. Section 9.3 discusses the case when $\mathbf{g}$ is an affine function. Reformulations as smooth, convex programs with polyhedral feasible sets are possible in this case. The main purpose of this section is to point out this special and tractable case. These reformulations are not necessarily apparent when approaching (DC) from the more general perspective taken in the GSIP literature, which is to use duality results for the lower level programs (see §9.2) to reformulate the infinite constraints. This approach is the subject of Section 9.5; thus in this section the lower level programs are convex programs, or more generally, strong duality holds for the lower-level programs. A number of reformulations of (DC) to simpler problems (finite nonlinear programs (NLPs) or standard semi-infinite programs (SIPs)) are possible. The application of global optimization methods to these reformulations reveals some interesting behavior that is not apparent when applying local methods, as in previous work. Section 9.6 discusses the most general case, when the lower-level programs are not necessarily convex, and the subsequent need for approximate methods. An example of robust design from an engineering application is considered. Section 9.7 concludes with some final thoughts.

9.2 Preliminaries

9.2.1 Notation

Denote the set of symmetric matrices in $\mathbb{R}^{n \times n}$ by $\mathbb{S}^{n \times n}$. For a symmetric matrix $\mathbf{M}$, the notation $\mathbf{M} \succeq \mathbf{0}$ ($\mathbf{M} \preceq \mathbf{0}$) means that $\mathbf{M}$ is positive (negative) semidefinite. Similarly, $\mathbf{M} \succ \mathbf{0}$ ($\mathbf{M} \prec \mathbf{0}$) means $\mathbf{M}$ is positive (negative) definite. Denote a square diagonal matrix with diagonal given by the vector $\mathbf{m}$ by $\operatorname{diag}(\mathbf{m})$.

9.2.2 Equivalence of (DC) and (GSIP)

Throughout this chapter, the terms "equivalent" and "equivalence" are used to relate two mathematical programs in the standard way.

Definition 9.2.1. (Equivalence) Two mathematical programs

$$\max_{\mathbf{x}} f(\mathbf{x}) \qquad\qquad \max_{\mathbf{x},\mathbf{z}} f(\mathbf{x})$$
$$\text{s.t.}\, \mathbf{x} \in X, \qquad\qquad \text{s.t.}\, (\mathbf{x},\mathbf{z}) \in S,$$

are said to be equivalent if for each $\mathbf{x} \in X$, there exists $\mathbf{z}$ such that $(\mathbf{x},\mathbf{z}) \in S$, and for each $(\mathbf{x},\mathbf{z}) \in S$, $\mathbf{x} \in X$.

It is clear that if two programs are equivalent, then the solution sets (if nonempty) have the same "$\mathbf{x}$" components since the objective functions are the same.

Next, the following assumption is made; this assumption holds for the remainder of this work, however most results still include hypotheses which imply this assumption. Under this assumption it is shown that in fact (DC) and (GSIP) are equivalent.

Assumption 9.2.1. *Assume that in* (DC), $D(\mathbf{x})$ *is nonempty and a subset of* Y *for all* $\mathbf{x} \in X$.

Consider the *lower level programs* (LLPs) of (GSIP), for $\mathbf{x} \in X$ and $i \in \{1,\dots,m\}$:

$$g_i^*(\mathbf{x}) = \sup\{g_i(\mathbf{y}) : \mathbf{y} \in D(\mathbf{x})\}. \tag{LLP i}$$

In the context of robust design, $\mathbf{y}$ represents parameters or inputs to a system. In the context of GSIP, $\mathbf{y}$ are called the lower (level) variables, while $\mathbf{x}$ are called the upper variables. For $\mathbf{x} \in X$, it is clear that if $D(\mathbf{x})$ is nonempty, then $\mathbf{x}$ is feasible in (GSIP) if and only if $g_i^*(\mathbf{x}) \le 0$ for each i. On the other hand, if $D(\mathbf{x})$ is empty, then no constraints are required to hold in (GSIP), and so $\mathbf{x}$ is feasible (alternatively we could define the supremum of a real function on the empty set as $-\infty$).

However, this is hardly acceptable behavior for a design centering problem. A method should not return an empty design space as an "optimal" solution, although a useful feature of a method would be the ability to identify whether any non-empty feasible design space exists. For the most part, we will need to assume, for instance, that G is nonempty in order to apply a method.

Another complicating fact is that G is defined as a subset of Y. If Assumption 9.2.1 did not hold, the constraints in (GSIP) would need to be modified to read

$$g_i(\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in D(\mathbf{x}) \cap Y.$$

However, this leads to similar complications as before. If, for instance, $D(\mathbf{x})$ is nonempty, but $D(\mathbf{x}) \cap Y$ is empty, then neither $D(\mathbf{x})$ nor $D(\mathbf{x}) \cap Y$ are acceptable solutions to (DC).

Understanding this, the equivalence of (DC) and (GSIP), established in the following result, is intuitive.

Proposition 9.2.1. *Under Assumption 9.2.1, problems* (DC) *and* (GSIP) *are equivalent.*

Proof. Since the objective functions in (DC) and (GSIP) are the same, we just need to establish that their feasible sets are the same. So consider $\mathbf{x}$ feasible in (DC). Then $\mathbf{x} \in X$ and $D(\mathbf{x}) \subset G$. This means that for all $\mathbf{y} \in D(\mathbf{x})$, $\mathbf{y} \in Y$ and $\mathbf{g}(\mathbf{y}) \leq \mathbf{0}$. We immediately have that $\mathbf{x}$ is feasible in (GSIP).

Conversely, choose $\mathbf{x}$ feasible in (GSIP). Again, $\mathbf{x} \in X$ and for all $\mathbf{y} \in D(\mathbf{x})$, $\mathbf{g}(\mathbf{y}) \leq \mathbf{0}$. Under Assumption 9.2.1, $D(\mathbf{x})$ is nonempty and a subset of Y, and so for all $\mathbf{y} \in D(\mathbf{x})$, $\mathbf{y} \in G$. Thus $D(\mathbf{x}) \subset G$ and $\mathbf{x}$ is feasible in (DC). □

9.2.3 Related problems

Design under uncertainty

It is often of interest in robust design to take into account unknown, uncertain, or varying parameters/inputs. This takes into account noise or model error. In this case, if we partition the lower level variables into controls inputs and uncertain parameter inputs $(\mathbf{u}, \mathbf{p})$, respectively, then an interesting problem is to determine whether a control set point exists that yields desirable system behavior for all possible realizations of the uncertainty, and if so, determine such a set point that allows the most flexibility to deviate from it. Given the set of possible uncertain inputs P, this problem is

$$\begin{aligned} &\max_{\mathbf{x}} \operatorname{vol}(\widetilde{D}(\mathbf{x})) \\ &\quad \text{s.t. } \mathbf{g}(\mathbf{u}, \mathbf{p}) \leq \mathbf{0}, \quad \forall (\mathbf{u}, \mathbf{p}) \in \widetilde{D}(\mathbf{x}) \times P, \\ &\qquad \mathbf{x} \in X. \end{aligned} \tag{9.1}$$

For instance, if the system of interest is a process near steady-state, then a feasible point $\mathbf{x}$ of (9.1) would guarantee that the process constraints $\mathbf{g}(\mathbf{u}(t), \mathbf{p}(t)) \leq \mathbf{0}$ are satisfied, for all t such that $(\mathbf{u}(t), \mathbf{p}(t)) \in \widetilde{D}(\mathbf{x}) \times P$. In other words, no matter what the values of the uncertain, potentially time-varying inputs $\mathbf{p}$ are, as long as the controlled inputs $\mathbf{u}$ take values in $\widetilde{D}(\mathbf{x})$, the system will behave as desired. We note that (9.1) is the same form as problem (GSIP), it is just that part of the form of the design space has been fixed to be the uncertainty P.

A related problem is that of calculating a "feasibility index" for process design under uncertainty. This idea goes back to [68, 195, 196], and has been addressed more recently in [52, 192, 193]. This problem can be written equivalently as an SIP or a min-max problem in the forms

$$\begin{aligned}
&\min_{\mathbf{x}} f(\mathbf{x}) && \iff && \min_{\mathbf{x}} f(\mathbf{x}) \\
&\text{s.t. } \tilde{g}(\mathbf{x}, \mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in \widetilde{Y}, && && \text{s.t. } 0 \geq \sup\{\tilde{g}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in \widetilde{Y}\}.
\end{aligned} \tag{9.2}$$

One interpretation of this problem is that $\mathbf{x}$ represents some process design decisions, while $\mathbf{y}$ is a vector of uncertain model parameters. The goal is to minimize some economic objective f of the design variables which guarantees safe process design for any realization of the uncertain parameter $\mathbf{y}$ (indicated by $\tilde{g}(\mathbf{x}, \mathbf{y}) \leq 0$, for any $\mathbf{y} \in \widetilde{Y}$).

The definition of the "flexibility index" in Equation 8 of [195] is a kind of GSIP. The rest of that work focuses on conditions that allow this definition to be reformulated as an SIP of the form (9.2). The results in [195] focus on the case when D is interval-valued. A similar argument is repeated below, which depends on having a design space which is the image of the unit ball under an affine mapping. In this case the proxy for volume is taken to be the determinant of the matrix in the affine transformation.

Proposition 9.2.2. *Suppose* $X \subset Y \times \mathbb{R}^{n_y \times n_y}$, $D : (\mathbf{y}_c, \mathbf{P}) \mapsto \{\mathbf{y}_c + \mathbf{P}\mathbf{y}_d : \|\mathbf{y}_d\| \leq 1\}$ *for some norm* $\|\cdot\|$ *on* $\mathbb{R}^{n_y}$, $D(\mathbf{y}_c, \mathbf{P}) \subset Y$ *for all* $(\mathbf{y}_c, \mathbf{P}) \in X$, *and* $\operatorname{vol}(D(\mathbf{y}_c, \mathbf{P})) = \det(\mathbf{P})$.

Then (GSIP) *is equivalent to the SIP*

$$\begin{aligned} &\max_{\mathbf{y}_c,\mathbf{P}} \det(\mathbf{P}) \\ &\quad \text{s.t. } \mathbf{g}(\mathbf{y}_c + \mathbf{P}\mathbf{y}_d) \leq \mathbf{0}, \quad \forall \mathbf{y}_d \in B_1 \equiv \{\mathbf{y} : \|\mathbf{y}\| \leq 1\}, \\ &\quad (\mathbf{y}_c, \mathbf{P}) \in X. \end{aligned} \tag{9.3}$$

Proof. The reformulation is immediate given the form of D and the fact that Assumption 9.2.1 holds. But to be explicit, consider the problem for given $(\mathbf{y}_c, \mathbf{P}) \in X$

$$g_i^{**}(\mathbf{y}_c, \mathbf{P}) = \sup\{g_i(\mathbf{y}_c + \mathbf{P}\mathbf{y}_d) : \mathbf{y}_d \in B_1\}.$$

For $\mathbf{y}$ feasible in (LLP i), by definition there exists $\mathbf{y}_d \in B_1$ such that $\mathbf{y} = \mathbf{y}_c + \mathbf{P}\mathbf{y}_d$. Thus $g_i^*(\mathbf{y}_c, \mathbf{P}) \leq g_i^{**}(\mathbf{y}_c, \mathbf{P})$. Conversely, for $\mathbf{y}_d \in B_1$, there exists $\mathbf{y} \in D(\mathbf{y}_c, \mathbf{P})$ with $\mathbf{y} = \mathbf{y}_c + \mathbf{P}\mathbf{y}_d$. Thus $g_i^{**}(\mathbf{y}_c, \mathbf{P}) \leq g_i^*(\mathbf{y}_c, \mathbf{P})$, and together the inequalities imply that the feasible sets of (GSIP) and (9.3) are the same, and so equivalence follows. □

In robust design applications, an extra step is required to make use of this form of D. To check that an operating condition or process parameters $\mathbf{y}$ are in the calculated design space requires checking that the norm of the solution $\mathbf{y}_d$ of $\mathbf{y} - \mathbf{y}_c = \mathbf{P}\mathbf{y}_d$ is less than one. Consequently, the LU factorization of the optimal $\mathbf{P}$ should be computed to minimize this computation, especially if it is to be performed online. Further, SIP (9.3) is still a somewhat abstract problem and assuming more structure leads to more tractable restrictions such as

$$\begin{aligned} &\max_{\mathbf{y}_c,\mathbf{d}} \sum_{i=1}^{n_y} \ln(d_i) \\ &\quad \text{s.t. } \mathbf{g}(\mathbf{y}_c + \text{diag}(\mathbf{d})\mathbf{y}_d) \leq \mathbf{0}, \quad \forall \mathbf{y}_d : \|\mathbf{y}_d\| \leq 1, \\ &\quad (\mathbf{y}_c, \mathbf{d}) \in X \subset \{(\mathbf{y}_c, \mathbf{d}) \in Y \times \mathbb{R}^{n_y} : \mathbf{d} > \mathbf{0}\}, \end{aligned} \tag{9.4}$$

where in effect the variable $\mathbf{P}$ in SIP (9.3) has been restricted to (a subset of) the space of diagonal positive-definite matrices (see also the proof of Corollary 9.3.3 for justification of the use of the logarithm of the objective). Of course (9.4) is still a semi-infinite problem, but under further assumptions on $\mathbf{g}$ and the norm used, finite convex reformulations are possible (see §9.3.2).

Dynamic problems and backward reachability

When the system of interest is dynamic in nature, $\mathbf{g}$ may be defined in terms of the solution of an initial value problem in ordinary differential equations. In this case the design centering problem has some connections to the problem of computing the "backward" reachable set of a dynamic system. For instance, consider that the lower-level variables $\mathbf{y}$ represent the initial conditions (concentrations) of a batch chemical reaction, and G is the set of initial conditions which yield a desired output purity specification. Then G can be characterized as the backward reachable set of the set of (final) states which satisfy the purity specification (see also the example in §9.6.3). The calculation or approximation of reachable sets has a rich literature [84, 152, 168], and see in particular [31, 103, 121] for the backwards problem.

In the present setting, calculating the exact backward reachable set (i.e. G) is of little use for the same reasons discussed in §9.1; in this example, a nominal initial condition and the maximum deviation from it such that the purity specification is still met are desired. An "inner" ellipsoidal estimate could be calculated, but this is typically only possible when the dynamics are linear [38, 142]. As mentioned earlier, this work aims to develop methods that are applicable even when $\mathbf{g}$ may be implicitly defined by the solutions of a nonlinear dynamic system.

9.3 Affine constraints

This section deals with the case that $\mathbf{g}$ is an affine function and $Y = \mathbb{R}^{n_y}$. Specifically, assume that for each $i \in \{1, \dots, m\}$,

$$g_i(\mathbf{y}) = \mathbf{c}_i^{\mathrm{T}}\mathbf{y} - b_i,$$

for some $\mathbf{c}_i \in \mathbb{R}^{n_y}$ and $b_i \in \mathbb{R}$. Consequently, G is a (convex) polyhedron. Reformulations for different forms of D are given; in each case the reformulation is a convex program.

In §9.5, reformulations of (GSIP) are presented which rely on strong duality holding for each (LLP i). Consequently, the reformulations in §9.5 will be applicable to the current situation with $\mathbf{g}$ affine and D convex-valued. However, in the best case those reformulations involve nonconvex NLPs, which do not reduce to convex programs under the assumptions of the present section. Thus, it is worthwhile to be aware of the special reformulations in the present section.

9.3.1 Ball-valued design space

Let the upper variables $\mathbf{x}$ of (DC) be $(\mathbf{y}_c, \delta) \in \mathbb{R}^{n_y} \times \mathbb{R}$, and let $D(\mathbf{x})$ be the closed δ-ball around $\mathbf{y}_c$ for some norm $\|\cdot\|$: $D : (\mathbf{y}_c, \delta) \mapsto \{\mathbf{y} : \|\mathbf{y} - \mathbf{y}_c\| \leq \delta\}$. Then problem (DC) becomes

$$\begin{aligned} &\max_{\mathbf{y}_c, \delta} \delta \\ &\text{s.t. } \mathbf{c}_i^{\mathrm{T}} \mathbf{y} - b_i \leq 0, \quad \forall \mathbf{y} : \|\mathbf{y} - \mathbf{y}_c\| \leq \delta, \quad \forall i \in \{1, \dots, m\}, \\ &\quad \mathbf{y}_c \in \mathbb{R}^{n_y}, \delta \geq 0. \end{aligned} \tag{9.5}$$

To check that Assumption 9.2.1 holds, note that $D(\mathbf{y}_c, \delta)$ is nonempty for $\delta \geq 0$ (it at least contains $\mathbf{y}_c$), and it must trivially be a subset of $Y = \mathbb{R}^{n_y}$.

Problem (9.5) can be reformulated as a linear program (LP), following the ideas in §8.5 of [33]. Problem (9.5) is related to the problem of Chebyshev centering. For specific norms, this reformulation also appears in [76]. To establish this, consider the LLPs:

$$g_i^*(\mathbf{y}_c, \delta) = \sup\{\mathbf{c}_i^{\mathrm{T}} \mathbf{y} : \|\mathbf{y} - \mathbf{y}_c\| \leq \delta\} - b_i. \tag{9.6}$$

The following lemma establishes an explicit expression for g_i^*.

Lemma 9.3.1. *For all $\mathbf{y}_c \in \mathbb{R}^{n_y}$ and all $\delta \geq 0$, g_i^* defined in Equation* (9.6) *satisfies*

$$g_i^*(\mathbf{y}_c, \delta) = \delta \|\mathbf{c}_i\|_* + \mathbf{c}_i^{\mathrm{T}} \mathbf{y}_c - b_i.$$

Proof. If $\delta = 0$, then the supremum defining g_i^* is over the singleton set $\{\mathbf{y}_c\}$, and so $g_i^*(\mathbf{y}_c, \delta) = \mathbf{c}_i^{\mathrm{T}} \mathbf{y}_c - b_i$ which satisfies the conclusion of the lemma. Otherwise, note that

$$g_i^*(\mathbf{y}_c, \delta) = \sup\{\mathbf{c}_i^{\mathrm{T}} (\mathbf{y} - \mathbf{y}_c) : \|\mathbf{y} - \mathbf{y}_c\| \leq \delta\} + \mathbf{c}_i^{\mathrm{T}} \mathbf{y}_c - b_i,$$

and further, if $\delta > 0$, one can set $\mathbf{z} = {}^{(\mathbf{y} - \mathbf{y}_c)}/_{\delta}$ and then

$$g_i^*(\mathbf{y}_c, \delta) = \sup\left\{(\delta \mathbf{c}_i)^{\mathrm{T}} \mathbf{z} : \|\mathbf{z}\| \leq 1\right\} + \mathbf{c}_i^{\mathrm{T}} \mathbf{y}_c - b_i.$$

Applying the definition of the dual norm $\|\cdot\|_*$, we obtain

$$g_i^*(\mathbf{y}_c, \delta) = \delta \|\mathbf{c}_i\|_* + \mathbf{c}_i^{\mathrm{T}} \mathbf{y}_c - b_i.$$

□

We can now establish that problem (9.5) is equivalent to an LP, which consequently provides an efficient numerical solution for problem (9.5).

Theorem 9.3.1. *Problem* (9.5) *is equivalent to the LP*

$$\begin{aligned} \max_{\mathbf{y}_c, \delta}\ & \delta \\ \text{s.t. } & \mathbf{c}_i^{\mathrm{T}} \mathbf{y}_c + (\|\mathbf{c}_i\|_*)\delta \le b_i \quad \forall i \in \{1, \dots, m\}, \\ & \mathbf{y}_c \in \mathbb{R}^{n_y}, \delta \ge 0. \end{aligned} \tag{9.7}$$

Proof. As in Proposition 9.2.1, equivalence follows once we show that the feasible sets are equal. So, consider $(\mathbf{y}_c, \delta)$ feasible in problem (9.5). Considering problem (9.5) as a GSIP, it follows that $g_i^*(\mathbf{y}_c, \delta) \le 0$ for all i. By Lemma 9.3.1, it immediately follows that $(\mathbf{y}_c, \delta)$ are feasible in (9.7). Conversely, for $(\mathbf{y}_c, \delta)$ feasible in (9.7), Lemma 9.3.1 again establishes that $(\mathbf{y}_c, \delta)$ are feasible in (9.5). □

9.3.2 General ellipsoidal design space

A convex reformulation is also possible when the design space is an ellipsoid and its "shape" is a decision variable. This is a special case of the reformulation in Proposition 9.2.2. In this case, let $X \subset \{(\mathbf{y}_c, \mathbf{P}) \in \mathbb{R}^{n_y} \times \mathbb{S}^{n_y \times n_y} : \mathbf{P} \succ \mathbf{0}\}$ and $D : (\mathbf{y}_c, \mathbf{P}) \mapsto \{\mathbf{y}_c + \mathbf{P}\mathbf{y}_d : \|\mathbf{y}_d\|_2 \le 1\}$, i.e., the design space is the image of the unit two-norm ball under an affine transformation, and thus an ellipsoid. Let $\mathrm{vol}(D(\cdot)) : (\mathbf{y}_c, \mathbf{P}) \mapsto \det(\mathbf{P})$ and $Y = \mathbb{R}^{n_y}$. Problem (DC) becomes

$$\begin{aligned} \max_{\mathbf{y}_c, \mathbf{P}}\ & \det(\mathbf{P}) \\ \text{s.t. } & \mathbf{c}_i^{\mathrm{T}}(\mathbf{y}_c + \mathbf{P}\mathbf{y}_d) - b_i \le 0, \quad \forall \mathbf{y}_d : \|\mathbf{y}_d\|_2 \le 1, \quad \forall i \in \{1, \dots, m\}, \\ & (\mathbf{y}_c, \mathbf{P}) \in X. \end{aligned}$$

Similarly to the argument in Theorem 9.3.1 (and taking the logarithm of the objective), this becomes

$$\begin{aligned} \max_{\mathbf{y}_c, \mathbf{P}} \; & \ln(\det(\mathbf{P})) \\ \text{s.t. } & \mathbf{c}_i^{\mathrm{T}} \mathbf{y}_c + \left\| \mathbf{c}_i^{\mathrm{T}} \mathbf{P} \right\|_2 - b_i \le 0, \quad \forall i \in \{1, \dots, m\}, \\ & (\mathbf{y}_c, \mathbf{P}) \in X. \end{aligned} \tag{9.8}$$

Problem 9.8 is in fact a convex program and enjoys a rich history of analysis; see [33, §8.4.2], [134, Sections 6.4.4 and 6.5], [95]. However, further reformulation to a "standard" form (such as a semidefinite program) is necessary in order to apply general-purpose software for cone programs such as YALMIP [3, 110] and CVX [62, 61] (both of which provide front-ends for the solvers SeDuMi [194, 2] and MOSEK [1]). This is possible by the arguments in [134, §6.4.4] or [20, §4.2].

9.3.3 Interval-valued design space

In the case of the infinity-norm, we can generalize the form of the design space a little more, and still obtain a fairly tractable formulation. In this case, let the upper variables $\mathbf{x}$ of (DC) be $(\mathbf{y}^L, \mathbf{y}^U) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}$, and let $D(\mathbf{y}^L, \mathbf{y}^U)$ be a nonempty interval: $D : (\mathbf{y}^L, \mathbf{y}^U) \mapsto [\mathbf{y}^L, \mathbf{y}^U]$. Again with affine $\mathbf{g}$, problem (DC) becomes

$$\begin{aligned} \max_{\mathbf{y}^L, \mathbf{y}^U} \; & \prod_j (y_j^U - y_j^L) \\ \text{s.t. } & \mathbf{c}_i^{\mathrm{T}} \mathbf{y} - b_i \le 0, \quad \forall \mathbf{y} \in [\mathbf{y}^L, \mathbf{y}^U], \quad \forall i \in \{1, \dots, m\}, \\ & \mathbf{y}^L \le \mathbf{y}^U, \\ & (\mathbf{y}^L, \mathbf{y}^U) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}. \end{aligned} \tag{9.9}$$

The constraints $\mathbf{y}^L \le \mathbf{y}^U$ ensure that $D(\mathbf{y}^L, \mathbf{y}^U)$ is nonempty, and thus that Assumption 9.2.1 holds.

The reformulation of this problem has been considered in [15, 171]. An alternative derivation follows. Begin by analyzing the lower-level programs $g_i^*(\mathbf{y}^L, \mathbf{y}^U) = \sup\{\mathbf{c}_i^{\mathrm{T}} \mathbf{y} : \mathbf{y} \in [\mathbf{y}^L, \mathbf{y}^U]\} - b_i$. Again, if $(\mathbf{y}^L, \mathbf{y}^U)$ is feasible in (9.9), then $g_i^*(\mathbf{y}^L, \mathbf{y}^U) \le 0$. Further, the lower-level programs are linear programs with box constraints, and consequently can be solved by inspection: an optimal solution $\mathbf{y}^i$ of the i^{th} lower-level program can be constructed by letting $y_j^i = y_j^U$ if $c_{i,j} \geq 0$ and $y_j^i = y_j^L$ otherwise (where $c_{i,j}$ denotes the j^{th} component of $\mathbf{c}_i$). In fact, we can construct matrices $\mathbf{M}_i^L$, $\mathbf{M}_i^U \in \mathbb{R}^{n_y \times n_y}$ by initializing them to zero matrices, and then setting the j^{th} element of the diagonal of $\mathbf{M}_i^U$ to 1 (one) if $c_{i,j} \geq 0$, and otherwise setting the j^{th} element of the diagonal of $\mathbf{M}_i^L$ to 1. The result is that $\mathbf{M}_i^L \mathbf{y}^L + \mathbf{M}_i^U \mathbf{y}^U = \mathbf{y}^i$ as constructed earlier. Since each $\mathbf{c}_i$ is constant, this holds no matter what the value of $(\mathbf{y}^L, \mathbf{y}^U)$ is. This leads to the following result.

Theorem 9.3.2. *Consider the linearly-constrained NLP*

$$\begin{aligned}
\max_{\mathbf{y}^L, \mathbf{y}^U} \; & \prod_j (y_j^U - y_j^L) \\
\text{s.t. } & \mathbf{c}_i^{\mathrm{T}} \mathbf{M}_i^L \mathbf{y}^L + \mathbf{c}_i^{\mathrm{T}} \mathbf{M}_i^U \mathbf{y}^U \leq b_i, \quad \forall i, \\
& \mathbf{y}^L \leq \mathbf{y}^U, \\
& (\mathbf{y}^L, \mathbf{y}^U) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_y},
\end{aligned} \tag{9.10}$$

where $\mathbf{M}_i^L$ and $\mathbf{M}_j^U$ are diagonal n_y by n_y matrices where the j^{th} element of the diagonals, $m_{i,j}^L$ and $m_{i,j}^U$, respectively, are given by

$$m_{i,j}^L = \begin{cases} 1 & c_{i,j} < 0, \\ 0 & c_{i,j} \geq 0, \end{cases} \quad \textit{and} \quad m_{i,j}^U = \begin{cases} 0 & c_{i,j} < 0, \\ 1 & c_{i,j} \geq 0. \end{cases}$$

Problem (9.10) *is equivalent to problem* (9.9).

A more numerically favorable restriction of (9.10) is possible, at the expense of the restriction potentially being infeasible if G is "thin."

Corollary 9.3.3. *Consider the convex program*

$$\begin{aligned}
\max_{\mathbf{y}^L, \mathbf{y}^U} \; & \sum_j \ln(y_j^U - y_j^L) \\
\text{s.t. } & \mathbf{c}_i^{\mathrm{T}} \mathbf{M}_i^L \mathbf{y}^L + \mathbf{c}_i^{\mathrm{T}} \mathbf{M}_i^U \mathbf{y}^U \leq b_i, \quad \forall i, \\
& \mathbf{y}^U - \mathbf{y}^L \geq \epsilon \mathbf{1}, \\
& (\mathbf{y}^L, \mathbf{y}^U) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_y},
\end{aligned} \tag{9.11}$$

where $\epsilon > 0$, and $\mathbf{M}_i^L$ and $\mathbf{M}_j^U$ are defined as in Theorem 9.3.2. If an optimal solution $(\mathbf{y}^{L,*}, \mathbf{y}^{U,*})$ of problem (9.9) *satisfies $\mathbf{y}^{U,*} - \mathbf{y}^{L,*} \geq \epsilon \mathbf{1}$, then this is also a solution of problem* (9.11).

Proof. Since $\ln(\cdot)$ is a nondecreasing concave function and for each j, $(y_j^L, y_j^U) \mapsto (y_j^U - y_j^L)$ is a concave function, then the objective function $\sum_j \ln(y_j^U - y_j^L)$ is concave on the convex feasible set, and so this maximization problem is indeed a convex program.

Denote the feasible set of problem (9.11) by X_R. Denote the objective function of (9.9) by $f(\mathbf{y}^L, \mathbf{y}^U) = \prod_j (y_j^U - y_j^L)$. Note that f is positive on X_R. If an optimal solution $(\mathbf{y}^{L,*}, \mathbf{y}^{U,*})$ of problem (9.9) satisfies $\mathbf{y}^{U,*} - \mathbf{y}^{L,*} \geq \epsilon \mathbf{1}$, then clearly $(\mathbf{y}^{L,*}, \mathbf{y}^{U,*}) \in X_R$. Since $\ln(\cdot)$ is increasing, $\arg\max\{f(\mathbf{y}^L, \mathbf{y}^U) : (\mathbf{y}^L, \mathbf{y}^U) \in X_R\} = \arg\max\{\ln(f(\mathbf{y}^L, \mathbf{y}^U)) : (\mathbf{y}^L, \mathbf{y}^U) \in X_R\}$, and so $(\mathbf{y}^{L,*}, \mathbf{y}^{U,*})$ is a solution of problem (9.11). □

9.4 Convex constraints

When Y is a convex set and $\mathbf{g}$ is a convex function, G is a convex set. In this case, results typically used in the analysis of interior point methods for convex programming inspire approaches for design centering. A few connections are mentioned here.

First, an analytic center of G is defined as an optimal solution of the convex problem

$$\begin{aligned} \min_{\mathbf{y}} \; & -\sum_{i=1}^{m} \ln(-g_i(\mathbf{y})) \\ \text{s.t. } & \mathbf{g}(\mathbf{y}) < \mathbf{0}. \end{aligned} \tag{9.12}$$

The objective is the logarithmic barrier function associated with the set of inequalities representing G, which tends to positive infinity at the boundary of G. Conceptually, an element of the solution set of (9.12) is maximizing the distance to the boundary of G. Consequently, an analytic center may provide a good estimate of the center of a ball or interval-valued design space. Unfortunately, as discussed in §9.1, a measure of "operational flexibility" is also desired, and in general this information is not available for the analytic center.

However, when ϕ is a self-concordant barrier function for G (see Definition 2.3.1 in [134]), one can make a statement about when an ellipsoid defined by the Hessian of ϕ (the Dikin ellipsoid) is a subset of G; see Proposition 2.3.2 of [134]. When $\mathbf{g}$ is affine (as in §9.3) the objective of (9.12) is in fact a self-concordant barrier function for G. Of course, when $\mathbf{g}$ is affine, the case of inscribing an ellipsoid has already been considered in §9.3.2. For further discussion and references see §8.5 of [33].

9.5 Convex LLP

For the most part, in this section it is assumed that the LLPs are convex programs (and so $\mathbf{g}$ is a *concave* function), but to be more accurate, the main focus of this section is when duality results hold for each LLP. When this is the case, a number of reformulations provide a way to solve (GSIP) via methods for NLPs or SIPs. In §9.5.1, the LLPs have a specific form and the concavity of $\mathbf{g}$ is not necessary. Reformulation results from the literature are reviewed and numerical examples considered.

9.5.1 Reformulation

KKT conditions

In the literature, one of the first reformulations of (GSIP) when the LLPs are convex programs comes from replacing the LLPs with algebraic constraints which are necessary and sufficient for a maximum; i.e., their KKT conditions. This approach can be found in [184, 185], for instance. The following result establishes the equivalence of (GSIP) with a mathematical program with complementarity constraints (MPCC), a type of NLP. Refer to MPCC (9.13) as the "KKT reformulation."

Proposition 9.5.1. *Suppose Y is a nonempty, open, convex set. Suppose $D(\mathbf{x})$ is compact for each $\mathbf{x} \in X$ and $D(\mathbf{x}) = \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}\}$ for some $\mathbf{h} : X \times Y \to \mathbb{R}^{n_h}$ where $\mathbf{h}(\mathbf{x}, \cdot)$ is convex and differentiable for each $\mathbf{x} \in X$. Suppose that for each $i \in \{1, \dots, m\}$ and each $\mathbf{x} \in X$ the Slater condition holds for* (LLP *i*)*: there exists a $\mathbf{y}_s \in Y$ such that $\mathbf{h}(\mathbf{x}, \mathbf{y}_s) < \mathbf{0}$.*

Suppose that $\mathbf{g}$ *is concave and differentiable. Then* (GSIP) *is equivalent to the MPCC*

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y}^1,\boldsymbol{\mu}^1,\dots,\mathbf{y}^m,\boldsymbol{\mu}^m} \quad & \mathrm{vol}(D(\mathbf{x})) && (9.13)\\ \text{s.t. } & g_i(\mathbf{y}^i) \le 0, \quad \forall i, \\ & \mathbf{h}(\mathbf{x},\mathbf{y}^i) \le \mathbf{0}, \quad \forall i, \\ & \nabla g_i(\mathbf{y}^i) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x},\mathbf{y}^i)\boldsymbol{\mu}^i = \mathbf{0}, \quad \forall i, \\ & \boldsymbol{\mu}^i \ge \mathbf{0}, \quad \mathbf{y}^i \in Y, \quad \forall i, \\ & \mu^i_j h_j(\mathbf{x},\mathbf{y}^i) = 0, \quad \forall (i,j), \\ & \mathbf{x} \in X. \end{aligned}$$

Note that the assumptions imply that $D(\mathbf{x})$ is nonempty and a subset of Y for all $\mathbf{x} \in X$; thus Assumption 9.2.1 is still satisfied. For each $i \in \{1,\dots,m\}$, the hypotheses imply that (LLP i) is a differentiable convex program (satisfying a constraint qualification) which achieves its maximum; thus there exists a $\boldsymbol{\mu}^i \in \mathbb{R}^{n_h}$ such that $(\mathbf{y}^i, \boldsymbol{\mu}^i)$ is a KKT point if and only if $\mathbf{y}^i$ is a global optimum of (LLP i). It can be shown that the result in Proposition 9.5.1 holds under weaker conditions than the Slater condition for the LLPs, as in [184]. However, the Slater condition has a natural interpretation in design centering problems; a design space must have some minimum size or afford some minimum amount of operational flexibility. The Slater condition is common to many reformulations in this work. Indeed, a Slater-like condition has already been used in Corollary 9.3.3 and will be used throughout the rest of §9.5.

The constraints $\mu^i_j h_j(\mathbf{x},\mathbf{y}^i) = 0$, $\mu^i_j \ge 0$, and $h_j(\mathbf{x},\mathbf{y}^i) \le 0$ in NLP (9.13) are the complementarity constraints which give the class of MPCC its name. Unfortunately, there are numerical difficulties involved in solving MPCCs. This relates to the fact that the Mangasarian-Fromovitz Constraint Qualification is violated everywhere in its feasible set [42]. This motivates the reformulation of Proposition 9.5.5.

Lagrangian dual

It is helpful to analyze a reformulation of (GSIP) based on Lagrangian duality at this point. Assume $D(\mathbf{x}) = \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x},\mathbf{y}) \le \mathbf{0}\}$ for some $\mathbf{h} : X \times Y \to \mathbb{R}^{n_h}$. Define the dual function

$q_i(\mathbf{x}, \cdot)$ and its effective domain by

$$q_i(\mathbf{x}, \boldsymbol{\mu}) = \sup\{g_i(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\},$$
$$\mathrm{dom}(q_i(\mathbf{x}, \cdot)) = \{\boldsymbol{\mu} \in \mathbb{R}^{n_h} : q_i(\mathbf{x}, \boldsymbol{\mu}) < +\infty\}.$$

Then define the (Lagrangian) dual problem of (LLP i) by

$$q_i^*(\mathbf{x}) = \inf\left\{q_i(\mathbf{x}, \boldsymbol{\mu}) : \boldsymbol{\mu} \geq \mathbf{0}, \boldsymbol{\mu} \in \mathrm{dom}(q_i(\mathbf{x}, \cdot))\right\}. \tag{9.14}$$

Under appropriate assumptions, one can establish that $g_i^*(\mathbf{x}) = q_i^*(\mathbf{x})$ (known as strong duality) for each $\mathbf{x}$. This forms the basis for the following results. The first result establishes that one can always obtain an SIP restriction of (GSIP).

Proposition 9.5.2. *Suppose $D(\mathbf{x}) = \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}\}$ for some $\mathbf{h} : X \times Y \to \mathbb{R}^{n_h}$. For any $M \subset \mathbb{R}^{n_h}$ and for any $(\mathbf{x}, \boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^m)$ feasible in the SIP*

$$\begin{aligned} \max_{\mathbf{x}, \boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^m} \quad & \mathrm{vol}(D(\mathbf{x})) \\ \text{s.t.}\ & g_i(\mathbf{y}) - (\boldsymbol{\mu}^i)^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in Y, \quad \forall i, \\ & \boldsymbol{\mu}^i \geq \mathbf{0}, \quad \boldsymbol{\mu}^i \in M, \quad \forall i, \\ & \mathbf{x} \in X, \end{aligned} \tag{9.15}$$

$\mathbf{x}$ *is feasible in* (GSIP).

Proof. Follows from Proposition 8.3.1. □

The next result establishes that SIP (9.15) is equivalent to (GSIP) under hypotheses similar to those in Proposition 9.5.1.

Proposition 9.5.3. *Suppose Y is convex. Suppose $D(\mathbf{x}) = \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}\}$ for some $\mathbf{h} : X \times Y \to \mathbb{R}^{n_h}$. For all $\mathbf{x} \in X$, suppose $\mathbf{g}$ is concave, $\mathbf{h}(\mathbf{x}, \cdot)$ is convex, $g_i^*(\mathbf{x})$ (defined by* (LLP i)*) is finite for all i, and there exists a $\mathbf{y}_s(\mathbf{x}) \in Y$ such that $\mathbf{g}(\mathbf{y}_s(\mathbf{x})) > -\mathbf{g}_b$ for some $\mathbf{g}_b > \mathbf{0}$ and $\mathbf{h}(\mathbf{x}, \mathbf{y}_s(\mathbf{x})) \leq -\mathbf{h}_b$ for some $\mathbf{h}_b > \mathbf{0}$. Then for compact $M = [\mathbf{0}, \mathbf{b}^*] \subset \mathbb{R}^{n_h}$ (where $b_j^* = \max_i\{g_{b,i}\}/h_{b,j}$),* (GSIP) *is equivalent to SIP* (9.15).

Proof. Follows from Lemmata 8.3.2 and 8.3.3 and Theorem 8.3.1. □

Wolfe dual

To obtain a more numerically tractable reformulation than the KKT reformulation (9.13), we follow the ideas in [42] to obtain a reformulation of (GSIP) which does not have complementarity constraints. This follows by looking at the dual function $q_i(\mathbf{x}, \cdot)$ and noting that if Y is a nonempty open convex set and g_i and $-\mathbf{h}(\mathbf{x}, \cdot)$ are concave and differentiable, then for $\boldsymbol{\mu} \geq \mathbf{0}$, the supremum defining the dual function is achieved at $\mathbf{y}$ if and only if $\nabla g_i(\mathbf{y}) - \nabla_{\mathbf{y}} \mathbf{h}(\mathbf{x}, \mathbf{y})\boldsymbol{\mu} = \mathbf{0}$. Consequently, we obtain the Wolfe dual problem of (LLP i):

$$q_i^W(\mathbf{x}) = \inf\left\{g_i(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y, \boldsymbol{\mu} \geq \mathbf{0}, \nabla g_i(\mathbf{y}) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y})\boldsymbol{\mu} = \mathbf{0}\right\}. \tag{9.16}$$

Under suitable assumptions (namely, that (LLP i) achieves its supremum and a Slater condition), strong duality holds. An alternate proof follows, based on (much better established) Lagrangian duality results. See also [58, §6.3].

Lemma 9.5.4. *Suppose Y is a nonempty open convex set. For a given $\mathbf{x} \in X$ and $i \in \{1, \dots, m\}$, suppose the following: $D(\mathbf{x})$ is compact and $D(\mathbf{x}) = \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}\}$ for some $\mathbf{h}(\mathbf{x}, \cdot) : Y \to \mathbb{R}^{n_h}$ which is convex and differentiable. Suppose that the Slater condition holds for (LLP i) (there exists a $\mathbf{y}_s \in Y$ such that $\mathbf{h}(\mathbf{x}, \mathbf{y}_s) < \mathbf{0}$). Suppose g_i is concave and differentiable. Then there exists $(\mathbf{y}^i, \boldsymbol{\mu}^i)$ satisfying $\boldsymbol{\mu}^i \geq \mathbf{0}$, $\mathbf{y}^i \in Y$, $\nabla g_i(\mathbf{y}^i) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y}^i)\boldsymbol{\mu}^i = \mathbf{0}$, and $g_i^*(\mathbf{x}) = g_i(\mathbf{y}^i) - (\boldsymbol{\mu}^i)^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}^i)$. Further, $q_i^W(\mathbf{x}) = g_i^*(\mathbf{x})$.*

Proof. First, it is established that the Wolfe dual is weaker than the Lagrangian dual (9.14) (i.e. $q_i^W(\mathbf{x}) \geq q_i^*(\mathbf{x})$, thus establishing weak duality between (LLP i) and the Wolfe dual). As before, since Y is a nonempty open convex set and g_i and $-\mathbf{h}$ are concave and differentiable, then for $\widetilde{\boldsymbol{\mu}} \geq \mathbf{0}$, $\sup\{g_i(\mathbf{y}) - \widetilde{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\}$ is achieved at $\widetilde{\mathbf{y}} \in Y$ if and only if $\nabla g_i(\widetilde{\mathbf{y}}) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \widetilde{\mathbf{y}})\widetilde{\boldsymbol{\mu}} = \mathbf{0}$. Let

$$F_W = \{(\widetilde{\mathbf{y}}, \widetilde{\boldsymbol{\mu}}) : \widetilde{\boldsymbol{\mu}} \geq \mathbf{0}, \widetilde{\mathbf{y}} \in Y, \nabla g_i(\widetilde{\mathbf{y}}) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \widetilde{\mathbf{y}})\widetilde{\boldsymbol{\mu}} = \mathbf{0}\}.$$

Thus, for all $(\widetilde{\mathbf{y}}, \widetilde{\boldsymbol{\mu}}) \in F_W$, we have

$$\sup\{g_i(\mathbf{y}) - \widetilde{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\} = g_i(\widetilde{\mathbf{y}}) - \widetilde{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \widetilde{\mathbf{y}}),$$

which also implies that $\widetilde{\boldsymbol{\mu}} \in \operatorname{dom}(q_i(\mathbf{x}, \cdot))$. It follows that for all $(\widetilde{\mathbf{y}}, \widetilde{\boldsymbol{\mu}}) \in F_W$, we have

$\widetilde{\boldsymbol{\mu}} \geq \mathbf{0}$, $\widetilde{\boldsymbol{\mu}} \in \operatorname{dom}(q_i(\mathbf{x}, \cdot))$, and $q_i(\mathbf{x}, \widetilde{\boldsymbol{\mu}}) = g_i(\widetilde{\mathbf{y}}) - \widetilde{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \widetilde{\mathbf{y}})$. Therefore, by definition of the dual problem (9.14), for all $(\widetilde{\mathbf{y}}, \widetilde{\boldsymbol{\mu}}) \in F_W$, we have

$$q_i^*(\mathbf{x}) \leq g_i(\widetilde{\mathbf{y}}) - \widetilde{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \widetilde{\mathbf{y}}).$$

Consequently, taking the infimum over all $(\widetilde{\mathbf{y}}, \widetilde{\boldsymbol{\mu}}) \in F_W$ yields $q_i^*(\mathbf{x}) \leq q_i^W(\mathbf{x})$, by the definition of the Wolfe dual. Note that F_W may be empty, in which case the infimum in the definition of $q_i^W(\mathbf{x})$ is over an empty set, and the inequality $q_i^*(\mathbf{x}) \leq q_i^W(\mathbf{x})$ holds somewhat trivially.

Next we establish $q_i^W(\mathbf{x}) \leq g_i^*(\mathbf{x})$, using strong duality for the Lagrangian dual. Since g_i is differentiable on Y, it is continuous on Y and since $D(\mathbf{x})$ is compact, (LLP i) achieves its supremum (since $D(\mathbf{x})$ is nonempty under the Slater condition). Under the convexity assumptions and Slater conditions, strong duality holds for (LLP i); i.e. $q_i^*(\mathbf{x}) = g_i^*(\mathbf{x})$ (see for instance Proposition 5.3.1 in [23]). Further, a duality multiplier exists; that is, there exists $\boldsymbol{\mu}^i \geq \mathbf{0}$ such that $g_i^*(\mathbf{x}) = \sup\{g_i(\mathbf{y}) - (\boldsymbol{\mu}^i)^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\}$. Since (LLP i) achieves its supremum, there exists a maximizer $\mathbf{y}^i$ of (LLP i). Because a duality multiplier exists, by Proposition 5.1.1 in [21], we have $\mathbf{y}^i \in \arg\max\{g_i(\mathbf{y}) - (\boldsymbol{\mu}^i)^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in Y\}$, and thus

$$g_i^*(\mathbf{x}) = g_i(\mathbf{y}^i) - (\boldsymbol{\mu}^i)^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}^i).$$

Again, since Y is a nonempty open set we have $\nabla g_i(\mathbf{y}^i) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x}, \mathbf{y}^i)\boldsymbol{\mu}^i = \mathbf{0}$. In other words, $(\mathbf{y}^i, \boldsymbol{\mu}^i) \in F_W$ defined before, which establishes the first claim. Finally, applying the definition of $q_i^W(\mathbf{x})$ as an infimum over F_W, we get that $g_i^*(\mathbf{x}) \geq q_i^W(\mathbf{x})$. But since $g_i^*(\mathbf{x}) = q_i^*(\mathbf{x})$ and $q_i^*(\mathbf{x}) \leq q_i^W(\mathbf{x})$ (established above), we have $g_i^*(\mathbf{x}) = q_i^W(\mathbf{x})$. □

With this, one can establish an NLP reformulation of (GSIP) which does not have complementarity constraints. Refer to NLP (9.17) as the "Wolfe reformulation."

Proposition 9.5.5. *Suppose Y is a nonempty open convex set. Suppose $D(\mathbf{x})$ is compact for each $\mathbf{x} \in X$ and $D(\mathbf{x}) = \{\mathbf{y} \in Y : \mathbf{h}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}\}$ for some $\mathbf{h} : X \times Y \to \mathbb{R}^{n_h}$ where $\mathbf{h}(\mathbf{x}, \cdot)$ is convex and differentiable for each $\mathbf{x} \in X$. Suppose that for each $i \in \{1, \ldots, m\}$ and each $\mathbf{x} \in X$ the Slater condition holds for* (LLP i)*: there exists a $\mathbf{y}_s \in Y$ such that $\mathbf{h}(\mathbf{x}, \mathbf{y}_s) < \mathbf{0}$.*

Suppose $\mathbf{g}$ *is concave and differentiable. Then* (GSIP) *is equivalent to the NLP*

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{y}^1,\boldsymbol{\mu}^1,\dots,\mathbf{y}^m,\boldsymbol{\mu}^m} \quad & \mathrm{vol}(D(\mathbf{x})) \\
\text{s.t.}\ & g_i(\mathbf{y}^i) - (\boldsymbol{\mu}^i)^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}^i) \le 0, \quad \forall i, \\
& \nabla g_i(\mathbf{y}^i) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x},\mathbf{y}^i)\boldsymbol{\mu}^i = \mathbf{0}, \quad \forall i, \\
& \boldsymbol{\mu}^i \ge \mathbf{0}, \quad \mathbf{y}^i \in Y, \quad \forall i, \\
& \mathbf{x} \in X.
\end{aligned} \tag{9.17}$$

Proof. Choose $\mathbf{x} \in X$ feasible in (GSIP), then $g_i^*(\mathbf{x}) \le 0$ for each i. By Lemma 9.5.4, there exists $(\mathbf{y}^i, \boldsymbol{\mu}^i)$ such that $\boldsymbol{\mu}^i \ge \mathbf{0}$, $\mathbf{y}^i \in Y$, $\nabla g_i(\mathbf{y}^i) - \nabla_{\mathbf{y}}\mathbf{h}(\mathbf{x},\mathbf{y}^i)\boldsymbol{\mu}^i = \mathbf{0}$, and $g_i^*(\mathbf{x}) = g_i(\mathbf{y}^i) - (\boldsymbol{\mu}^i)^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}^i)$. In other words, $(\mathbf{x},\mathbf{y}^1,\boldsymbol{\mu}^1,\dots,\mathbf{y}^m,\boldsymbol{\mu}^m)$ is feasible in (9.17).

Conversely, choose $(\mathbf{x},\mathbf{y}^1,\boldsymbol{\mu}^1,\dots,\mathbf{y}^m,\boldsymbol{\mu}^m)$ feasible in (9.17). Again, by Lemma 9.5.4 (in fact, weak duality between (LLP i) and the Wolfe dual (9.16) suffices), we must have $g_i^*(\mathbf{x}) \le 0$ for each i, which establishes that $\mathbf{x}$ is feasible in (GSIP). Equivalence follows. □

One notes that indeed NLP (9.17) does not have the complementarity constraints that make MPCC (9.13) numerically unfavorable. Proposition 9.5.5 is similar to Corollary 2.4 in [42]. The difference is that the latter result assumes that $-\mathbf{g}$ and $\mathbf{h}(\mathbf{x},\cdot)$ are convex on all of $Y = \mathbb{R}^{n_y}$. As this is a rather strong assumption, this motivates the authors of [42] to weaken this, and merely assume that $\mathbf{g}$ is concave on $D(\mathbf{x})$ for each $\mathbf{x}$. They then obtain an NLP which adds the constraints $\mathbf{h}(\mathbf{x},\mathbf{y}^i) \le \mathbf{0}$ to NLP (9.17). The justification is cursory, although the result is plausible. In design centering applications, assuming that $\mathbf{g}$ is concave on Y versus assuming $\mathbf{g}$ is concave on $D(\mathbf{x})$ for all $\mathbf{x}$ is typically not much stronger anyway.

From inspection of their constraints, NLP (9.17) is a relaxation of MPCC (9.13). However, what is interesting is that (GSIP) is equivalent to MPCC (9.13) and NLP (9.17) under the same conditions (the identical assumptions of Propositions 9.5.1 and 9.5.5). Thus, MPCC (9.13) and NLP (9.17) are in fact equivalent under these conditions.

When (LLP i) is a linear program or second-order cone program, more specific duality results hold and thus stronger reformulations are possible (see §8.4). However, this would imply that g_i is affine, and this case has been covered in §9.3.

General quadratically constrained quadratic LLP

When $Y = \mathbb{R}^{n_y}$, $D(\mathbf{x})$ is defined in terms of a ball given by a weighted 2-norm, and each g_i is quadratic, a specific duality result can be used to reformulate (GSIP). It should be stressed that this does not require that the LLPs are convex programs, despite the fact that this is part of a section titled "Convex LLP." This duality result applies to the general case of a quadratic program with a single quadratic constraint; given $\mathbf{A}_0$, $\mathbf{A} \in \mathbb{S}^{n_y \times n_y}$, $\mathbf{b}_0$, $\mathbf{b} \in \mathbb{R}^{n_y}$, and c_0, $c \in \mathbb{R}$, define

$$p^* = \sup\{\mathbf{y}^{\mathrm{T}}\mathbf{A}_0\mathbf{y} + 2\mathbf{b}_0^{\mathrm{T}}\mathbf{y} + c_0 : \mathbf{y} \in \mathbb{R}^{n_y}, \mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{y} + 2\mathbf{b}^{\mathrm{T}}\mathbf{y} + c \leq 0\}. \tag{9.18}$$

The (Lagrangian) dual of this problem is

$$\begin{aligned} &q^Q : \mu \mapsto \sup\{\mathbf{y}^{\mathrm{T}}(\mathbf{A}_0 - \mu\mathbf{A})\mathbf{y} + 2(\mathbf{b}_0 - \mu\mathbf{b})^{\mathrm{T}}\mathbf{y} + c_0 - \mu c : \mathbf{y} \in \mathbb{R}^{n_y}\}, \\ &d^* = \inf\{q^Q(\mu) : \mu \in \mathrm{dom}(q^Q), \mu \geq 0\}. \end{aligned} \tag{9.19}$$

Noting that the Lagrangian of (9.18) is a quadratic function, for given $\mu \geq 0$ the supremum defining the dual function q^Q is achieved at $\mathbf{y}^*$ if and only if the second-order conditions $\mathbf{A}_0 - \mu\mathbf{A} \preceq \mathbf{0}$, $(\mathbf{A}_0 - \mu\mathbf{A})\mathbf{y}^* = -(\mathbf{b}_0 - \mu\mathbf{b})$, are satisfied; otherwise the supremum is $+\infty$. This leads to

$$\begin{aligned} d^* = \inf_{\mu,\mathbf{y}}\ & \mathbf{y}^{\mathrm{T}}(\mathbf{A}_0 - \mu\mathbf{A})\mathbf{y} + 2(\mathbf{b}_0 - \mu\mathbf{b})^{\mathrm{T}}\mathbf{y} + c_0 - \mu c \\ \text{s.t. } & (\mathbf{A}_0 - \mu\mathbf{A})\mathbf{y} = -(\mathbf{b}_0 - \mu\mathbf{b}), \\ & \mathbf{A}_0 - \mu\mathbf{A} \preceq \mathbf{0}, \\ & \mu \geq 0, \quad \mathbf{y} \in \mathbb{R}^{n_y} \end{aligned} \tag{9.20}$$

(note the similarity to the Wolfe dual (9.16)). Whether or not program (9.18) is convex, strong duality holds assuming (9.18) has a Slater point. That is to say, $p^* = d^*$ (and the dual solution set is nonempty) assuming there exists $\mathbf{y}_s$ such that $\mathbf{y}_s^{\mathrm{T}}\mathbf{A}\mathbf{y}_s + 2\mathbf{b}^{\mathrm{T}}\mathbf{y}_s + c < 0$. A proof of this can be found in Appendix B of [33]. The proof depends on the somewhat cryptically named "S-procedure," which is actually a theorem of the alternative. A review of results related to the S-procedure or S-lemma can be found in [147]. The required results are stated formally in the following.

Lemma 9.5.6. *Consider the quadratically constrained quadratic program* (9.18) *and its dual* (9.19). *Suppose there exists* $\mathbf{y}_s$ *such that* $\mathbf{y}_s^{\mathrm{T}}\mathbf{A}\mathbf{y}_s + 2\mathbf{b}^{\mathrm{T}}\mathbf{y}_s + c < 0$. *Then* $p^* = d^*$. *Further, if* p^* *is finite, then the solution set of the dual problem* (9.19) *is nonempty.*

Lemma 9.5.7. *Consider problem* (9.18) *and its dual* (9.19). *If* $p^* = d^*$ *(strong duality holds), and there exists* $(\mu^*, \mathbf{y}^*)$ *with* μ^* *in the solution set of the dual* (9.19) *and* $\mathbf{y}^*$ *in the solution set of problem* (9.18), *then* $(\mu^*, \mathbf{y}^*)$ *is optimal in problem* (9.20).

Proof. Since there is no duality gap, μ^* is a duality multiplier (see Proposition 5.1.4 in [21]). Thus, any optimal solution of the primal problem (9.18) maximizes the Lagrangian for this fixed μ^*, i.e. $q^Q(\mu^*) = (\mathbf{y}^*)^{\mathrm{T}}(\mathbf{A}_0 - \mu^*\mathbf{A})\mathbf{y}^* + 2(\mathbf{b}_0 - \mu^*\mathbf{b})^{\mathrm{T}}\mathbf{y}^* + c_0 - \mu^* c$ (see Proposition 5.1.1 in [21]). Thus the second-order conditions $\mathbf{A}_0 - \mu^*\mathbf{A} \preceq \mathbf{0}$, $(\mathbf{A}_0 - \mu^*\mathbf{A})\mathbf{y}^* = -(\mathbf{b}_0 - \mu^*\mathbf{b})$, are satisfied. Since $d^* = q^Q(\mu^*)$, $(\mu^*, \mathbf{y}^*)$ must be optimal in (9.20). □

A reformulation of (GSIP) when the LLPs are quadratically constrained quadratic programs follows.

Proposition 9.5.8. *Suppose* $Y = \mathbb{R}^{n_y}$, *and that for* $i \in \{1, \dots, m\}$ *there exist* $(\mathbf{A}_i, \mathbf{b}_i, c_i) \in \mathbb{S}^{n_y \times n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}$ *such that* $g_i : \mathbf{y} \mapsto \mathbf{y}^{\mathrm{T}}\mathbf{A}_i\mathbf{y} + 2\mathbf{b}_i^{\mathrm{T}}\mathbf{y} + c_i$. *Suppose* $X \subset \{(\mathbf{P}, \mathbf{y}_c) \in \mathbb{S}^{n_y \times n_y} \times \mathbb{R}^{n_y} : \mathbf{P} \succ \mathbf{0}\}$. *Suppose that* $D(\mathbf{P}, \mathbf{y}_c) = \{\mathbf{y} : (\mathbf{y} - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}(\mathbf{y} - \mathbf{y}_c) \le 1\}$ *and* $\mathrm{vol}(D(\mathbf{P}, \mathbf{y}_c)) = \det(\mathbf{P})^{-1}$. *Then* (GSIP) *is equivalent to the program*

$$\begin{aligned}
\max_{\mathbf{P}, \mathbf{y}_c, \boldsymbol{\mu}, \mathbf{y}^1, \dots, \mathbf{y}^m} \quad & \det(\mathbf{P})^{-1} && (9.21)\\
\text{s.t. } & (\mathbf{y}^i)^{\mathrm{T}}\mathbf{A}_i\mathbf{y}^i + 2\mathbf{b}_i^{\mathrm{T}}\mathbf{y}^i + c_i - \mu_i\left((\mathbf{y}^i - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}(\mathbf{y}^i - \mathbf{y}_c) - 1\right) \le 0, \quad \forall i,\\
& (\mathbf{A}_i - \mu_i\mathbf{P})\mathbf{y}^i = -(\mathbf{b}_i + \mu_i\mathbf{P}\mathbf{y}_c), \quad \forall i,\\
& \mathbf{A}_i - \mu_i\mathbf{P} \preceq \mathbf{0}, \quad \forall i,\\
& \mu_i \ge 0, \quad \mathbf{y}^i \in \mathbb{R}^{n_y}, \quad \forall i,\\
& (\mathbf{P}, \mathbf{y}_c) \in X.
\end{aligned}$$

Proof. Note that $D(\mathbf{P}, \mathbf{y}_c) = \{\mathbf{y} : \mathbf{y}^{\mathrm{T}}\mathbf{P}\mathbf{y} - 2\mathbf{y}_c^{\mathrm{T}}\mathbf{P}\mathbf{y} + \mathbf{y}_c^{\mathrm{T}}\mathbf{P}\mathbf{y}_c - 1 \le 0\}$. Also, for all $(\mathbf{P}, \mathbf{y}_c) \in X$, a solution exists for (LLP i) for each i since $\mathbf{P}$ is constrained to be positive definite and so $D(\mathbf{P}, \mathbf{y}_c)$ is compact. Further, by assumption on X, $\mathbf{y}_c$ is a Slater point for each LLP for all $(\mathbf{P}, \mathbf{y}_c) \in X$. Consequently, by Lemma 9.5.6, strong duality holds for each LLP and an optimal dual solution exists.

Choose $(\mathbf{P}, \mathbf{y}_c)$ feasible in (GSIP). Then for all i, $g_i^*(\mathbf{P}, \mathbf{y}_c) \leq 0$. Then by Lemma 9.5.7, there exists $(\mu_i, \mathbf{y}^i)$ optimal in the dual of (LLP i) written in the form (9.20), and combined with strong duality $(\mathbf{P}, \mathbf{y}_c, \boldsymbol{\mu}, \mathbf{y}^1, \dots, \mathbf{y}^m)$ is feasible in (9.21). Conversely, choose $(\mathbf{P}, \mathbf{y}_c, \boldsymbol{\mu}, \mathbf{y}^1, \dots, \mathbf{y}^m)$ feasible in (9.21). Weak duality establishes that $g_i^*(\mathbf{P}, \mathbf{y}_c) \leq 0$ for each i, and so $(\mathbf{P}, \mathbf{y}_c)$ is feasible in (GSIP). Equivalence follows. □

Note that program (9.21) contains nonlinear matrix inequalities. Consequently, many general-purpose software for the solution of NLP cannot handle this problem. Choosing $\mathbf{P}$ by some heuristic leads to a more practical reformulation. Further, Y can be restricted to a subset of $\mathbb{R}^{n_y}$ by taking advantage of strong duality. Refer to NLP (9.22) as the "Quadratic reformulation."

Corollary 9.5.1. *Suppose that for $i \in \{1, \dots, m\}$ there exist $(\mathbf{A}_i, \mathbf{b}_i, c_i) \in \mathbb{S}^{n_y \times n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}$ such that $g_i : \mathbf{y} \mapsto \mathbf{y}^{\mathrm{T}}\mathbf{A}_i\mathbf{y} + 2\mathbf{b}_i^{\mathrm{T}}\mathbf{y} + c_i$. Suppose $\mathbf{P} \in \mathbb{S}^{n_y \times n_y}$ is given and $\mathbf{P} \succ \mathbf{0}$, and further $X \subset \{(\mathbf{y}_c, \delta) : \mathbf{y}_c \in \mathbb{R}^{n_y}, \delta \geq \epsilon\}$ for some $\epsilon > 0$. Suppose that $D(\mathbf{y}_c, \delta) = \{\mathbf{y} : (\mathbf{y} - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}(\mathbf{y} - \mathbf{y}_c) \leq \delta^2\}$, $\mathrm{vol}(D(\mathbf{y}_c, \delta)) = \delta$, and that $Y \subset \mathbb{R}^{n_y}$ satisfies $D(\mathbf{y}_c, \delta) \subset Y$ for all $(\mathbf{y}_c, \delta) \in X$. Then* (GSIP) *is equivalent to the program*

$$\begin{aligned}
\max_{\mathbf{y}_c, \delta, \boldsymbol{\mu}, \mathbf{y}^1, \dots, \mathbf{y}^m} \quad & \delta \\
\text{s.t. } & g_i(\mathbf{y}^i) - \mu_i \left((\mathbf{y}^i - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}(\mathbf{y}^i - \mathbf{y}_c) - \delta^2 \right) \leq 0, \quad \forall i, \\
& (\mathbf{A}_i - \mu_i \mathbf{P})\mathbf{y}^i = -(\mathbf{b}_i + \mu_i \mathbf{P}\mathbf{y}_c), \quad \forall i, \\
& \mathbf{A}_i - \mu_i \mathbf{P} \preceq \mathbf{0}, \quad \forall i, \\
& \mu_i \geq 0, \quad \mathbf{y}^i \in Y, \quad \forall i, \\
& (\mathbf{y}_c, \delta) \in X.
\end{aligned} \tag{9.22}$$

Proof. The proof is similar to that of Proposition 9.5.8. The added constraint that the $\mathbf{y}^i$ components of the solutions of (9.22) are in Y does not change the fact that (9.22) is a "restriction" of (GSIP) (i.e. for $(\mathbf{y}_c, \delta, \boldsymbol{\mu}, \mathbf{y}^1, \dots, \mathbf{y}^m)$ feasible in (9.22), $(\mathbf{y}_c, \delta)$ is feasible in (GSIP)).

We only need to check that for $(\mathbf{y}_c, \delta)$ feasible in (GSIP), that there exist $(\mu_i, \mathbf{y}^i)$ such that $(\mathbf{y}_c, \delta, \boldsymbol{\mu}, \mathbf{y}^1, \dots, \mathbf{y}^m)$ is feasible in (9.22) (specifically that $\mathbf{y}^i \in Y$ for each i). But this must hold since we can take $\mathbf{y}^i$ and μ_i to be optimal solutions of (LLP i) and its dual, respectively, for each i. Thus $\mathbf{y}^i \in D(\mathbf{y}_c, \delta) \subset Y$ for each i, and so by strong duality and

Lemma 9.5.7, $(\mathbf{y}_c, \delta, \boldsymbol{\mu}, \mathbf{y}^1, \dots, \mathbf{y}^m)$ is feasible in (9.22). □

The Quadratic reformulation (9.22) is still nonlinear and nonconvex, but the matrix inequality constraints are linear, and as demonstrated by an example in §9.5.2, these can sometimes be reformulated as explicit constraints on $\boldsymbol{\mu}$.

Convex quadratic constraints

This section discusses a special case, when $D(\mathbf{x})$ is defined in terms of a ball given by a weighted 2-norm, and each g_i is *convex* and quadratic. In this case, a convex reformulation is possible. This case has the geometric interpretation of inscribing the maximum volume ellipsoid in the intersection of ellipsoids, and has been considered in [20, §4.9.1], [32, §3.7.3], [33, §8.5]. The representation of the design space is a little different from what has been considered so far; it depends on the inverse of the symmetric square root of a positive semidefinite matrix.

Proposition 9.5.9. *Suppose* $Y = \mathbb{R}^{n_y}$, *and that for* $i \in \{1, \dots, m\}$ *there exist* $(\mathbf{A}_i, \mathbf{b}_i, c_i) \in \mathbb{S}^{n_y \times n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}$, *with* $\mathbf{A}_i \succ \mathbf{0}$, *such that* $g_i : \mathbf{y} \mapsto \mathbf{y}^{\mathrm{T}}\mathbf{A}_i\mathbf{y} + 2\mathbf{b}_i^{\mathrm{T}}\mathbf{y} + c_i$. *Suppose* $X \subset \{(\mathbf{P}, \mathbf{y}_c) \in \mathbb{S}^{n_y \times n_y} \times \mathbb{R}^{n_y} : \mathbf{P} \succ \mathbf{0}\}$. *Suppose that* $D(\mathbf{P}, \mathbf{y}_c) = \{\mathbf{y} : (\mathbf{y} - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}^{-2}(\mathbf{y} - \mathbf{y}_c) \le 1\}$ *and* $\mathrm{vol}(D(\mathbf{P}, \mathbf{y}_c)) = \det(\mathbf{P})$. *Then* (GSIP) *is equivalent to the program*

$$\max_{\mathbf{P}, \mathbf{y}_c, \boldsymbol{\mu}} \det(\mathbf{P}) \tag{9.23}$$

$$\text{s.t. } \boldsymbol{\mu} \ge 0, \quad (\mathbf{P}, \mathbf{y}_c) \in X,$$

$$\begin{bmatrix} -\mathbf{A}_i^{-1} & -\mathbf{A}_i^{-1}\mathbf{b}_i - \mathbf{y}_c & \mathbf{P} \\ (-\mathbf{A}_i^{-1}\mathbf{b}_i - \mathbf{y}_c)^{\mathrm{T}} & \mu_i - (\mathbf{b}_i^{\mathrm{T}}\mathbf{A}_i^{-1}\mathbf{b}_i - c_i) & \mathbf{0} \\ \mathbf{P} & \mathbf{0} & -\mu_i\mathbf{I} \end{bmatrix} \preceq \mathbf{0}, \quad \forall i.$$

Proof. The proof depends on the following characterization of (in essence) the dual function of (LLP i) in this case: Assuming $\mathbf{P}$, $\mathbf{A}_i$ are positive definite symmetric matrices and $\mu_i \ge 0$, we have

$$\sup\left\{\mathbf{y}^{\mathrm{T}}\mathbf{A}_i\mathbf{y} + 2\mathbf{b}_i^{\mathrm{T}}\mathbf{y} + c_i - \mu_i\left((\mathbf{y} - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}^{-2}(\mathbf{y} - \mathbf{y}_c) - 1\right) : \mathbf{y} \in \mathbb{R}^{n_y}\right\} \le 0 \tag{9.24}$$

if and only if

$$\begin{bmatrix} -\mathbf{A}_i^{-1} & -\mathbf{A}_i^{-1}\mathbf{b}_i - \mathbf{y}_c & \mathbf{P} \\ (-\mathbf{A}_i^{-1}\mathbf{b}_i - \mathbf{y}_c)^{\mathrm{T}} & \mu_i - (\mathbf{b}_i^{\mathrm{T}}\mathbf{A}_i^{-1}\mathbf{b}_i - c_i) & \mathbf{0} \\ \mathbf{P} & \mathbf{0} & -\mu_i\mathbf{I} \end{bmatrix} \preceq \mathbf{0}.$$

A proof of this equivalence can be found in §3.7.3 of [32].

Choosing $(\mathbf{P}, \mathbf{y}_c)$ feasible in (GSIP), by strong duality (Lemma 9.5.6) we have that for each i there exists $\mu_i \geq 0$ such that the dual function is nonpositive (i.e. Inequality (9.24) holds). Thus $(\mathbf{P}, \mathbf{y}_c, \boldsymbol{\mu})$ is feasible in problem (9.23). Conversely, for $(\mathbf{P}, \mathbf{y}_c, \boldsymbol{\mu})$ feasible in problem (9.23), by the above equivalence and weak duality $(\mathbf{P}, \mathbf{y}_c)$ is feasible in (GSIP). □

The matrix constraints in problem (9.23) are linear inequalities, and reformulating the objective along the lines of the discussion in §9.3.2 yields an SDP. As another practical note, the explicit matrix inverses in problem (9.23) could be removed, for instance, by introducing new variables $(\mathbf{E}_i, \mathbf{d}_i, f_i)$ and adding the linear constraints $\mathbf{A}_i\mathbf{E}_i = \mathbf{I}$, $\mathbf{A}_i\mathbf{d}_i = \mathbf{b}_i$, $\mathbf{b}_i^{\mathrm{T}}\mathbf{d}_i = f_i$, for each i.

9.5.2 Numerical examples

In this section global NLP solvers are applied to the reformulations of design centering problems discussed in the previous sections. The studies are performed in GAMS version 24.3.3 [56]. Deterministic global optimizers BARON version 14.0.3 [197, 159] and ANTIGONE version 1.1 [120] are employed. Algorithm 2.1 in [122] is applied to the SIP reformulation. More specifically, an implementation of Algorithm 7 from §8.5 is employed. This implementation is coded in GAMS, employing BARON for the solution of the subproblems. Unless otherwise noted the parameters are $\epsilon_{R,0} = 1$, $r = 2$, and $Y^{LBP,0} = Y^{UBP,0} = \varnothing$. These examples have a single infinite constraint (single LLP), and so the subscripts on g and solution components are dropped. All numerical studies were performed on a 64-bit Linux virtual machine allocated a single core of a 3.07 GHz Intel Xeon processor and 1.28 GB RAM.

Table 9.1: Solution times and solutions for problem (9.25) by various reformulations.

Method	Solution Time (s)		Solution			
	BARON	ANTIGONE	$\mathbf{y}^L$	$\mathbf{y}^U$	$\boldsymbol{\mu}$	$\mathbf{y}$
KKT reformulation (9.13)	0.07	0.05	$(0,-1)$	$(1,1)$	$(2,0,0,0)$	$(0,1)$
Wolfe reformulation (9.17)	0.65	0.29	$(0,-1)$	$(1,1)$	$(2,0,0,0)$	$(0,1)$
SIP reformulation (9.15)	10.5		$(-1,-1)$	$(1,0)$	$(0,0,0,2)$	

Convex LLP with interval design space

The following design centering problem is considered:

$$\max_{\mathbf{y}^L,\mathbf{y}^U} \operatorname{vol}([\mathbf{y}^L,\mathbf{y}^U]) \tag{9.25}$$
$$\text{s.t. } g(\mathbf{y}) = -(y_1+1)^2 - (y_2-1)^2 + 1 \le 0, \quad \forall \mathbf{y} \in [\mathbf{y}^L,\mathbf{y}^U],$$

where $\operatorname{vol}([\mathbf{y}^L,\mathbf{y}^U]) = (y_1^U - y_1^L)(y_2^U - y_2^L)$.

For the KKT and Wolfe reformulations, letting $Y = (-2,2) \times (-2,2)$, $X = \{(\mathbf{y}^L,\mathbf{y}^U) \in [-1,1]^2 \times [-1,1]^2 : y_1^U - y_1^L \ge 0.002, y_2^U - y_2^L \ge 0.002\}$, and

$$\mathbf{h} : X \times Y \ni (\mathbf{y}^L,\mathbf{y}^U,\mathbf{y}) \mapsto \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{y}^L \\ -\mathbf{y}^U \end{bmatrix}$$

it is clear that the hypotheses of Propositions 9.5.1 and 9.5.5 hold. The corresponding reformulations are solved to global optimality with BARON and ANTIGONE. The relative and absolute optimality tolerances are both 10^{-4}. The solution obtained in each case is $(\mathbf{y}^L,\mathbf{y}^U) = (0,-1,1,1)$. The other components of the solution and the solution times are in Table 9.1.

Meanwhile, the SIP reformulation from Proposition 9.5.3 holds for $M = [\mathbf{0},\mathbf{b}^*]$, where $\mathbf{b}^* = (18 \times 10^3)\mathbf{1}$ (for instance, by noting that $g(\mathbf{y}) \ge -18$ for all $\mathbf{y} \in Y$ and taking $h_{b,i} = 0.001$). Let $Y = [-2,2] \times [-2,2]$ for the purposes of this reformulation. For Algorithm 7, let the subproblem relative and absolute optimality tolerances δ_r and δ_a equal 10^{-5} and the overall relative and absolute optimality tolerances equal 10^{-4}. The method terminates in 28 iterations and the solution obtained is $(\mathbf{y}^L,\mathbf{y}^U) = (-1,-1,1,0)$. Although different from what what obtained with the NLP reformulations, the optimal objective value is the same and it is still optimal. The solution time is included in Table 9.1.

As expected, the NLP reformulations are quicker to solve than the SIP reformulation. What is somewhat surprising is that the KKT reformulation, which is an MPCC, solves more quickly than the Wolfe reformulation, which omits the complementarity constraints. This is perhaps due to the nature of the global solvers BARON and ANTIGONE, which can recognize and efficiently handle the complementarity constraints [159], and overall make use of the extra constraints to improve domain reduction through constraint propagation.

However, note that the KKT and Wolfe reformulations involve the derivatives of $\mathbf{g}$ and $\mathbf{h}$. Subsequently, solving these reformulations with implementations of global methods such as BARON and ANTIGONE requires explicit expressions for these derivatives. In a general-purpose modeling language such as GAMS, supplying these derivative expressions typically must be done by hand which is tedious and error prone. In contrast, the various NLP subproblems required by Algorithm 7 are defined in terms of the original functions $\mathbf{g}$ and $\mathbf{h}$.

Nonconvex quadratic LLP

The following design centering problem is considered:

$$\begin{aligned} &\max_{\mathbf{y}_c, \delta} \delta \\ &\text{s.t. } g(\mathbf{y}) = y_1^2 - y_2^2 \le 0, \quad \forall \mathbf{y} : \|\mathbf{y} - \mathbf{y}_c\|_2 \le \delta. \end{aligned} \tag{9.26}$$

Note that g is a nonconvex quadratic function, and that the lower-level program is a quadratically-constrained quadratic program. Each of the reformulations of §9.5 are considered. This is to demonstrate what happens when strong duality holds for the lower-level program, but an inappropriate reformulation is used. It will be seen that the KKT and Wolfe reformulations fail to give a correct answer, while the SIP reformulation succeeds.

Let $X = \{(\mathbf{y}_c, \delta) : \|\mathbf{y}_c\|_2 \le 2, 0.1 \le \delta \le 2\}$. Let

$$\mathbf{A} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] \quad \text{and} \quad \mathbf{P} = \mathbf{I},$$

where $\mathbf{I}$ is the identity matrix, so that $g(\mathbf{y}) = \mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{y}$ and $D(\mathbf{y}_c, \delta) = \{\mathbf{y} : (\mathbf{y} - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}(\mathbf{y} - \mathbf{y}_c) \le \delta^2\}$. With $Y = [-4, 4] \times [-4, 4]$, the Quadratic reformulation (9.22) is applicable by Corollary 9.5.1. Further, since $\mathbf{A}$ and $\mathbf{P}$ are diagonal, the matrix inequality constraint $\mathbf{A} - \mu\mathbf{P} \preceq \mathbf{0}$ in that problem can be reformulated as nonpositivity of the diagonal elements

Table 9.2: Solution times and solutions for problem (9.26) by various reformulations.

Method	Solution Time (s)		Solution			
	BARON	ANTIGONE	$\mathbf{y}_c$	δ	μ	$\mathbf{y}$
Quadratic reformulation (9.22)	2.07	17.08	$(0,-2)$	1.414	1	$(4,-1)$
KKT reformulation (9.13)	0.01	0.01	$(0,0)$	2	0	$(0,0)$
Wolfe reformulation (9.17)	0.01	0.01	$(0,0)$	2	0	$(0,0)$
SIP reformulation (9.15)	22.76		$(0,2)$	1.414	1	

of $\mathbf{A} - \mu\mathbf{P}$, which reduces to $\mu \geq 1$ (and $\mu \geq -1$, but this is redundant). General-purpose solvers can handle this form of the constraint more easily.

The Quadratic reformulation (9.22) is solved to global optimality with BARON and ANTIGONE. The relative and absolute optimality tolerances are both 10^{-4}. The solution obtained in each case is $\mathbf{y}_c = (0,-2)$, $\delta = 1.414$. The other components of the solution and solution statistics are in Table 9.2.

The SIP reformulation is also applicable in this case. The center of the design space $\mathbf{y}_c$ is a Slater point for the lower-level program for all $(\mathbf{y}_c, \delta) \in X$, and so $g(\mathbf{y}_c) \geq -4$ for all $(\mathbf{y}_c, \delta) \in X$. Further, let $h(\mathbf{y}_c, \delta, \mathbf{y}) = (\mathbf{y} - \mathbf{y}_c)^{\mathrm{T}}\mathbf{P}(\mathbf{y} - \mathbf{y}_c) - \delta^2$, so that $h(\mathbf{y}_c, \delta, \mathbf{y}_c) = -\delta^2 \leq -0.01$ for all $(\mathbf{y}_c, \delta) \in X$. With Lemma 8.3.3 and the strong duality result from Lemma 9.5.6, the conclusion of Proposition 9.5.3 holds with $M = [0, 400]$. For Algorithm 7, let the subproblem relative and absolute optimality tolerances δ_r and δ_a equal 2×10^{-5} and the overall relative and absolute optimality tolerances equal 10^{-4}. The method terminates in 33 iterations and the solution obtained is $(\mathbf{y}_c, \delta) = (0, 2, 1.414)$, a different optimal solution for (9.26). What is interesting to note is that the SIP solution method is competitive with the solution of the Quadratic reformulation for this example; see Table 9.2.

Not surprisingly, the KKT and Wolfe reformulations fail to provide even a feasible solution. Quite simply, this is due to the omission of the constraint $\mu \geq 1$, which is the only difference between the Wolfe reformulation (9.17) and the Quadratic reformulation (9.22). Consequently, even when strong duality holds, one must be careful if attempting to apply the KKT and Wolfe reformulations. In [42], the authors apply reformulations similar to (9.13) and (9.17) to a problem with quadratic LLPs, but do not include the constraint on the duality multiplier for a nonconvex LLP. However, since the numerical results were for a local solution method, if the starting point for the local solver was sufficiently close to a local minimum, convergence to an infeasible point would not be observed.

9.6 Nonconvex LLP: Approximation methods

In this section, approaches for finding a feasible solution of (GSIP) are considered, with optimality being a secondary concern. To this end, the focus is on constructing restrictions of (GSIP) which can be solved to global optimality practically. This has the benefit that these methods do not rely on initial guesses that typically must be supplied to a local optimization method.

Furthermore, the motivation of this section are those instances of (GSIP) when $\mathbf{g}$ might not be explicitly defined. For instance, in many engineering applications, the design constraints $\mathbf{g}$ may be defined implicitly by the solution of a system of algebraic or differential equations; the example in §9.6.3 provides an instance. In this case, many solution methods are impractical if not impossible to apply. For example, in the previous section, global solution of the NLP reformulations (9.13) and (9.17) typically requires explicit expressions for the derivative of $\mathbf{g}$. Thus applying these reformulations does not lead to a practical method. In the context of SIP, [193] provides a good discussion of why many methods (for SIP) cannot be applied practically to infinitely-constrained problems in engineering applications.

Methods for the general case of GSIP, and related problems such as bi-level programs, with nonconvex lower-level programs have been presented in [125, 127, 201]. The method in [127] is an extension of the SIP method from [122], upon which the developments of §9.6.2 are based. As that section demonstrates, the method from [122] does provide a practical approximate method for (GSIP). Similar arguments seem to apply to the method for nonconvex GSIP in [127], and thus this method merits further investigation for its applicability to robust design in engineering applications. However, the method is based on a reformulation of (GSIP) which introduces nonsmoothness. The implementation of the method requires reformulation of this nonsmoothness by introducing integer variables. The result is that the method requires the iterative solution of mixed-integer nonlinear programs, whereas the method in §9.6.2 only needs to solve NLPs, for which no additional nonsmoothness has been introduced.

9.6.1 Interval restriction with branch and bound

In this section, a method for solving a restriction of (GSIP) is described. The restriction is constructed by noting that the constraint $g_i(\mathbf{y}) \leq 0$ for all $\mathbf{y} \in D(\mathbf{x})$ is equivalent to

$g_i^*(\mathbf{x}) \leq 0$, where g_i^* is defined by (LLP i). To describe the restriction and solution method, make the following assumption; as in §9.3.3, it is assumed that a candidate design space is an interval parameterized by its endpoints.

Assumption 9.6.1. *Let $\mathbb{I}Y$ denote the set of all nonempty interval subsets of Y ($\mathbb{I}Y = \{[\mathbf{v}, \mathbf{w}] \subset Y : [\mathbf{v}, \mathbf{w}] \neq \varnothing\}$). Suppose that $X \subset \{(\mathbf{v}, \mathbf{w}) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_y} : \mathbf{v} \leq \mathbf{w}\}$ and denote the upper variables of* (GSIP) *as $\mathbf{x} = (\mathbf{y}^L, \mathbf{y}^U)$ and let $D(\mathbf{y}^L, \mathbf{y}^U) = [\mathbf{y}^L, \mathbf{y}^U]$. Assume that for interval A,* vol(A) *equals the volume of A. Assume that for each i there is a function $g_i^U : \mathbb{I}Y \to \mathbb{R}$ such that $g_i^U(D(\mathbf{x})) \geq g_i^*(\mathbf{x})$ for all $\mathbf{x} \in X$. Assume that each g_i^U is monotonic in the sense that $g_i^U(A) \leq g_i^U(B)$ for all A, $B \in \mathbb{I}Y$ with $A \subset B$.*

Under Assumption 9.6.1, the following program is a restriction of (GSIP):

$$\begin{aligned}
\max_{\mathbf{x}}\ & \mathrm{vol}(D(\mathbf{x})) \\
\text{s.t. } & g_i^U(D(\mathbf{x})) \leq 0, \quad \forall i, \\
& \mathbf{x} \in X.
\end{aligned} \tag{9.27}$$

One could take $g_i^U(A) = \max\{g_i(\mathbf{y}) : \mathbf{y} \in A\}$ (so that $g_i^U(D(\mathbf{x})) = g_i^*(\mathbf{x})$ trivially). There exist a number of results dealing with such mappings; [11, 12, 100, 151] are among a few dealing with the continuity and differentiability properties of such maps. Then one approach to solve (GSIP) might be to analyze (9.27) as an NLP using some of these parametric optimization results. This characterizes the "local reduction" approaches in [77, 91, 157, 187], as well as a local method for SIP which takes into account the potential nonsmoothness of g_i^* in [146].

The subject of this section is a different approach, where the restriction (9.27) is solved globally with branch and bound. In this case, one can take advantage of g_i^U which are cheap to evaluate. For instance, one choice for g_i^U would be the upper bound of an interval-valued inclusion monotonic inclusion function of g_i (see [130] for an introduction to interval arithmetic and inclusion functions). The benefit is that the interval-valued inclusion functions are typically cheaper to evaluate than the global optimization problems defining g_i^*. The idea of using interval arithmetic to construct a restriction is related to the method in [155], except that the optimization approach in that work is based on an "evolutionary" optimization algorithm. Conceptually similar are the methods for the global solution of SIP in [28, 29].

To solve problem (9.27) via branch and bound, we need to be able to obtain lower bounds and upper bounds on the optimal objective values of the subproblems

$$f_k^* = \max\{\mathrm{vol}(D(\mathbf{x})) : \mathbf{x} \in X_k, g_i^U(\mathbf{x}) \leq 0, \forall i\},$$

where $X_k \subset X$. For this discussion assume X_k is a nonempty interval subset of X. This means X_k will have the form $[\mathbf{v}_k^L, \mathbf{v}_k^U] \times [\mathbf{w}_k^L, \mathbf{w}_k^U]$ for $\mathbf{v}_k^L$, $\mathbf{v}_k^U$, $\mathbf{w}_k^L$, $\mathbf{w}_k^U \in \mathbb{R}^{n_y}$. Under Assumption 9.2.1, we always have $\mathbf{y}^L \leq \mathbf{y}^U$ for $(\mathbf{y}^L, \mathbf{y}^U) = \mathbf{x} \in X$. Thus, if X_k is a subset of X, we have $\mathbf{v}_k^U \leq \mathbf{w}_k^L$. Furthermore,

$$[\mathbf{v}^U, \mathbf{w}^L] \subset [\mathbf{y}^L, \mathbf{y}^U] \subset [\mathbf{v}^L, \mathbf{w}^U], \quad \forall (\mathbf{y}^L, \mathbf{y}^U) \in [\mathbf{v}^L, \mathbf{v}^U] \times [\mathbf{w}^L, \mathbf{w}^U] : \mathbf{v}^U \leq \mathbf{w}^L.$$

Consequently, $UB_k = \mathrm{vol}([\mathbf{v}_k^L, \mathbf{w}_k^U])$ is an upper bound for the optimal subproblem objective f_k^*. Meanwhile, any feasible point provides a lower bound. In the context of the current problem, there are two natural choices:

1. The point $(\mathbf{v}_k^L, \mathbf{w}_k^U)$ represents the "largest" candidate design space possible in X_k. Consequently, if feasible, it gives the best lower bound for this node.
2. The point $(\mathbf{v}_k^U, \mathbf{w}_k^L)$ represents the "smallest" candidate design space possible in X_k, and thus is more likely to be feasible, and thus to provide a nontrivial lower bound. However, if it is infeasible, i.e. if $g_i^U(\mathbf{v}_k^U, \mathbf{w}_k^L) > 0$ for some i, then $g_i^U(\mathbf{y}^L, \mathbf{y}^U) > 0$ for all $(\mathbf{y}^L, \mathbf{y}^U) = \mathbf{x} \in X_k$ (by the monotonicity property in Assumption 9.6.1), and thus X_k can be fathomed by infeasibility.

As noted, with the definition $g_i^U(D(\mathbf{x})) = g_i^*(\mathbf{x})$, determining whether either of these points is feasible requires evaluating g_i^*, which is still a global optimization problem. In contrast, if the g_i^U are cheap to evaluate, these lower and upper bounds are also cheap to obtain.

Under mild assumptions, if (GSIP) has a solution, so does the restriction (9.27), although it may be a somewhat trivial solution. Assume $g_i^U([\mathbf{y}, \mathbf{y}]) = g_i(\mathbf{y})$ for all $\mathbf{y}$ (as is the case when g_i^U is the upper bound of an interval extension of g_i). Let $D(\mathbf{x}^*) = [\mathbf{y}^{L,*}, \mathbf{y}^{U,*}]$ be a solution of (GSIP); then for any $\widehat{\mathbf{y}} \in D(\mathbf{x}^*)$, we have $g_i(\widehat{\mathbf{y}}) \leq 0$ for all i, and so $[\widehat{\mathbf{y}}, \widehat{\mathbf{y}}]$ is feasible in the restriction (9.27), assuming $(\widehat{\mathbf{y}}, \widehat{\mathbf{y}}) \in X$. Although $[\widehat{\mathbf{y}}, \widehat{\mathbf{y}}]$ would violate the Slater condition that has been present in many results, it is unnecessary in this approach.

Numerical experiments (see §9.6.3) show that the branch and bound algorithm applied to this problem can be slow. To try to explain why this might be the case, consider the lower and upper bounds described above. For a nonempty interval subset $X_k = [\mathbf{v}_k^L, \mathbf{v}_k^U] \times [\mathbf{w}_k^L, \mathbf{w}_k^U]$ of X, in the worst case $f_k^* = \mathrm{vol}(D(\mathbf{v}_k^U, \mathbf{w}_k^L))$, while the upper bound is $UB_k = \mathrm{vol}(D(\mathbf{v}_k^L, \mathbf{w}_k^U))$. In one dimension ($n_y = 1$), for example, $\mathrm{vol}(D(y^L, y^U)) = y^U - y^L$, so $f_k^* = w_k^L - v_k^U$ and $UB_k = w_k^U - v_k^L$. Meanwhile, the width (or diameter) of X_k is $\mathrm{diam}(X_k) = \max\{(w_k^U - w_k^L), (v_k^U - v_k^L)\}$. Thus one has

$$UB_k - f_k^* = (w_k^U - w_k^L) + (v_k^U - v_k^L) \le 2\,\mathrm{diam}(X_k).$$

Thus, the bounding procedure described is at least first-order convergent (see Definition 2.1 in [206], noting that the present problem is a maximization). See §9.8 for the general case. However, also note that for all $\alpha > 0$, there exists nonempty $\widetilde{X} = [\widetilde{v}^L, \widetilde{v}^U] \times [\widetilde{w}^L, \widetilde{w}^U]$ sufficiently small so that $\widetilde{w}^U - \widetilde{w}^L > \alpha(\widetilde{w}^U - \widetilde{w}^L)^2$ and $\widetilde{v}^U - \widetilde{v}^L > \alpha(\widetilde{v}^U - \widetilde{v}^L)^2$ which implies

$$\begin{aligned}(\widetilde{w}^U - \widetilde{w}^L) + (\widetilde{v}^U - \widetilde{v}^L) &> \alpha\big((\widetilde{w}^U - \widetilde{w}^L)^2 + (\widetilde{v}^U - \widetilde{v}^L)^2\big)\\ &\ge \alpha \max\{(\widetilde{w}^U - \widetilde{w}^L)^2, (\widetilde{v}^U - \widetilde{v}^L)^2\}\\ &= \alpha(\max\{(\widetilde{w}^U - \widetilde{w}^L), (\widetilde{v}^U - \widetilde{v}^L)\})^2 = \alpha\,\mathrm{diam}(\widetilde{X})^2.\end{aligned}$$

This establishes that the method is not, in general, second-order convergent. When the solution is unconstrained, a convergence order of two or greater is required to avoid the "cluster problem" when applying the branch and bound method; this refers to a phenomenon that hinders the efficiency of the branch and bound method (see [45, 207]). A deeper understanding of these issues might be wise if attempting to develop the method in this section further.

9.6.2 SIP restriction

Proposition 9.5.2 provides the inspiration for another restriction-based method; for $M \subset \mathbb{R}^{n_h}$ with nonempty intersection with the nonnegative orthant, SIP (9.15) is a restriction of (GSIP). Subsequently solving this SIP restriction with a feasible-point method yields a feasible solution of (GSIP).

In contrast, the other duality-based reformulations of (GSIP), such as those in Propositions 9.5.1 and 9.5.5, do not provide restrictions if the assumption of convexity of the LLPs is dropped. Furthermore, as mentioned in the discussion in §9.5.2, solving these reformulations globally would require explicit expressions for the derivatives of $\mathbf{g}$. Since the overall goal of this section is to be able to handle robust design problems where $\mathbf{g}$ might be defined implicitly by the solution of systems of algebraic or differential equations, such information can be difficult to obtain.

The discussion in §9.5.2 also demonstrates that the SIP reformulation can take advantage of strong duality even in the cases that the specific hypotheses of Proposition 9.5.3 fail. In other words, if strong duality happens to hold for the LLPs of a specific problem, there is hope that the global solution of SIP (9.15) will also be the global solution of the original problem (GSIP).

For simplicity assume $m = 1$ (so there is a single LLP) and drop the corresponding index. Global solution of SIP (9.15) by the feasible-point method from [122] (at a specific iteration of the method) requires the solution of the subproblems (see also §8.5)

$$\begin{aligned} \max_{\mathbf{x},\boldsymbol{\mu}} \ & \mathrm{vol}(D(\mathbf{x})) \\ \text{s.t. } & g(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in Y^{UBP}, \\ & \boldsymbol{\mu} \geq \mathbf{0}, \quad \boldsymbol{\mu} \in M \\ & \mathbf{x} \in X, \end{aligned} \tag{UBP}$$

$$\begin{aligned} \max_{\mathbf{x},\boldsymbol{\mu}} \ & \mathrm{vol}(D(\mathbf{x})) \\ \text{s.t. } & g(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) \leq -\epsilon_R, \quad \forall \mathbf{y} \in Y^{LBP}, \\ & \boldsymbol{\mu} \geq \mathbf{0}, \quad \boldsymbol{\mu} \in M, \\ & \mathbf{x} \in X, \end{aligned} \tag{LBP}$$

and for given $(\mathbf{x}, \boldsymbol{\mu})$,

$$q(\mathbf{x}, \boldsymbol{\mu}) = \sup\{g(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) : \mathbf{y} \in Y\}, \tag{SIP LLP}$$

for finite subsets $Y^{LBP} \subset Y$, $Y^{UBP} \subset Y$, and $\epsilon_R > 0$. Note that each subproblem is a finite NLP that is defined in terms of the original functions g and $\mathbf{h}$, and not their derivatives. As their names suggest, (UBP) and (LBP) aim to furnish upper and lower bounds, respectively, on SIP (9.15) that converge as the algorithm iterates.

A source of numerical difficulty that can arise in applying this method follows. A part of the algorithm is determining the feasibility (in SIP (9.15)) of the optimal solution $(\mathbf{x}, \boldsymbol{\mu})$ of either (UBP) or (LBP). This requires solving (SIP LLP) and checking that $q(\mathbf{x}, \boldsymbol{\mu}) \leq 0$. One must either guarantee that $q(\mathbf{x}, \boldsymbol{\mu}) \leq 0$, or else find $\mathbf{y} \in Y$ such that $g(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) > 0$. In the latter case, $\mathbf{y}$ is added to the discretization set Y^{UBP} (or Y^{LBP}). Typically global optimization methods find such guaranteed bounds and feasible points, but in practice we can often have the situation on finite termination that the approximate solution $\mathbf{y}$ of (SIP LLP) and its guaranteed upper bound UB^{llp} satisfy $g(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0 < UB^{llp}$. In this case, we cannot guarantee that the point $(\mathbf{x}, \boldsymbol{\mu})$ is feasible in SIP (9.15). Meanwhile, adding $\mathbf{y}$ to the discretization set Y^{UBP} does nothing to further restrict (UBP) since $g(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0$; that is, the upper-bounding problem remains unchanged for the next iteration. Its solution in the next iteration is the same as in the last, and the same ambiguity arises when solving (SIP LLP). The cycle repeats, and the upper bound that the method provides fails to improve. A similar effect can occur with the lower-bounding problem (LBP).

This effect can be overcome by redefining g by adding a constant tolerance to its value. Consider that the pathological case $\widetilde{g} = g(\mathbf{y}) - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0 < UB^{llp}$ occurs, where $\mathbf{y}$ is the approximate solution found for (SIP LLP). Assume that the relative and absolute optimality tolerances for the global optimization method used are $\varepsilon_{rtol} \leq 1$ and ε_{atol}, respectively. In this case $UB^{llp} - \widetilde{g} > \varepsilon_{rtol} |\widetilde{g}|$. So assuming that the termination criterion of the global optimization procedure is $UB^{llp} - \widetilde{g} \leq \max\{\varepsilon_{atol}, \varepsilon_{rtol} |\widetilde{g}|\}$, it is easy to see that we must have $UB^{llp} - \widetilde{g} \leq \varepsilon_{atol}$ and thus $UB^{llp} \leq \widetilde{g} + \varepsilon_{atol}$.

To determine if $(\mathbf{x}, \boldsymbol{\mu})$ is feasible, solve (SIP LLP) and let the solution be $\mathbf{y}$. Then, if $g(\mathbf{y}) + \varepsilon_{atol} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) \leq 0$, by the preceding discussion we can guarantee that $(\mathbf{x}, \boldsymbol{\mu})$ is feasible in the SIP (9.15). However, if $g(\mathbf{y}) + \varepsilon_{atol} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x}, \mathbf{y}) > 0$, adding the point $\mathbf{y}$ to the discretization set Y^{UBP} actually does restrict the upper-bounding problem (UBP), where g is redefined as $g \equiv g + \varepsilon_{atol}$. In effect, the following restriction of SIP (9.15)

$$\begin{aligned} &\max_{\mathbf{x},\boldsymbol{\mu}} \operatorname{vol}(D(\mathbf{x})) \\ &\quad \text{s.t. } g(\mathbf{y}) + \varepsilon_{atol} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{h}(\mathbf{x},\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in Y, \\ &\quad\quad \boldsymbol{\mu} \geq \mathbf{0}, \quad \boldsymbol{\mu} \in M, \\ &\quad\quad \mathbf{x} \in X, \end{aligned} \tag{9.28}$$

is being solved with a method which does not quite guarantee feasibility (in (9.28)) of the solutions found. However, the solutions found *are* feasible in the original SIP (9.15).

For concreteness, further aspects of this approach are discussed in the context of the example considered in §9.6.3.

9.6.3 Practical aspects and an example

Consider the following robust design problem. In a batch chemical reactor, two chemical species A and B react according to mass-action kinetics with an Arrhenius dependence on temperature to form chemical species C. However, A and C also react according to mass-action kinetics with a dependence on temperature to form chemical species D. The initial concentrations of A and B vary from batch to batch, although it can be assumed they are never outside the range [0.5, 2] (M). Although temperature can be controlled by a cooling element, it too might vary from batch to batch; it can be assumed it never leaves the range [300, 800] (K). What are the largest acceptable ranges for the initial concentrations of A and B and the operating temperature to ensure that the mole fraction of the undesired side product D is below 0.05 at the end of any batch operation?

As a mathematical program this problem is written as (since there is one LLP the subscript on g is dropped)

$$\begin{aligned} &\max_{\mathbf{y}^L,\mathbf{y}^U} \prod_j (y_j^U - y_j^L) \\ &\quad \text{s.t. } g(\mathbf{y}) = \frac{z_{\mathrm{D}}(t_f,\mathbf{y})}{\mathbf{1}^{\mathrm{T}}\mathbf{z}(t_f,\mathbf{y})} - 0.05 \leq 0, \quad \forall \mathbf{y} \in [\mathbf{y}^L, \mathbf{y}^U], \\ &\quad\quad \mathbf{y}^L \leq \mathbf{y}^U, \\ &\quad\quad \mathbf{y}^L, \mathbf{y}^U \in [0.5, 2] \times [0.5, 2] \times [3, 8], \end{aligned} \tag{9.29}$$

where $\mathbf{z}$ is a solution of the initial value problem in parametric ordinary differential equations

$$\begin{aligned}
\dot{z}_{\mathrm{A}}(t,\mathbf{y}) &= -k_1(100y_3)z_{\mathrm{A}}(t,\mathbf{y})z_{\mathrm{B}}(t,\mathbf{y}) - k_2(100y_3)z_{\mathrm{A}}(t,\mathbf{y})z_{\mathrm{C}}(t,\mathbf{y}),\\
\dot{z}_{\mathrm{B}}(t,\mathbf{y}) &= -k_1(100y_3)z_{\mathrm{A}}(t,\mathbf{y})z_{\mathrm{B}}(t,\mathbf{y}),\\
\dot{z}_{\mathrm{C}}(t,\mathbf{y}) &= k_1(100y_3)z_{\mathrm{A}}(t,\mathbf{y})z_{\mathrm{B}}(t,\mathbf{y}) - k_2(100y_3)z_{\mathrm{A}}(t,\mathbf{y})z_{\mathrm{C}}(t,\mathbf{y}),\\
\dot{z}_{\mathrm{D}}(t,\mathbf{y}) &= k_2(100y_3)z_{\mathrm{A}}(t,\mathbf{y})z_{\mathrm{C}}(t,\mathbf{y}),
\end{aligned}$$

on the time interval $[t_0, t_f]$, with initial conditions $\mathbf{z}(t_0,\mathbf{y}) = (y_1, y_2, 0, 0)$. See Table 9.3 for a summary of the parameter values used and the expressions for the kinetic parameters k_1 and k_2. Note that the variables y_1 and y_2 correspond to the initial concentrations of A and B, respectively, while y_3 is a scaled temperature. This scaling helps overcome some numerical issues. These numerical studies were performed on a 64-bit Linux virtual machine allocated a single core of a 3.07 GHz Intel Xeon processor.

Table 9.3: Parameter values for problem (9.29).

Symbol	Value/Expression
$[t_0, t_f]$	$[0, 0.1]$ (h)
A_1	150 ($\mathrm{M}^{-1}\mathrm{h}$)
A_2	80 ($\mathrm{M}^{-1}\mathrm{h}$)
E_1	4×10^3 (J/mol)
E_2	15×10^3 (J/mol)
R	8.3145 ($\mathrm{J/K \cdot mol}$)
k_1	$k_1 : T \mapsto A_1 \exp(-E_1/(RT))$
k_2	$k_2 : T \mapsto A_2 \exp(-E_2/(RT))$

SIP restriction

First consider applying the SIP restriction method discussed in §9.6.2. Let

$$\begin{aligned}
Y &= [0.5, 2] \times [0.5, 2] \times [3, 8],\\
X &= \{(\mathbf{y}^L, \mathbf{y}^U) \in Y \times Y : \mathbf{y}^U - \mathbf{y}^L \geq (0.01)\mathbf{1}\},\\
M &= [\mathbf{0}, (10)\mathbf{1}],
\end{aligned}$$

and

$$\mathbf{h} : X \times Y \ni (\mathbf{y}^L, \mathbf{y}^U, \mathbf{y}) \mapsto \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{y}^L \\ -\mathbf{y}^U \end{bmatrix}.$$

Thus it is clear that X, Y, and M are compact, and that the objective and constraints of the SIP restriction (9.15) are continuous; the latter (specifically, the continuity of g) follows from standard parametric analysis of initial value problems from, for instance, Chapter II of [116]. The value of M is somewhat arbitrary; the SIP restriction is still valid for any M. If M is too large, there can be numerical issues and solving the subproblems can be slow, but as M becomes smaller, the restriction can become more conservative.

Consider the subproblems (LBP), (UBP), and (SIP LLP) that must be solved. The lower-level program of the SIP restriction for this example is a global dynamic optimization problem. This problem must be solved repeatedly in the course of the SIP solution algorithm as a test for feasibility. Although this may seem daunting, the computational time is reasonable. Here, the "direct method" approach of sequential parameterization (sometimes called single-shooting) is taken; see §1.1 of [163]. A C++ code implementing the dynamic affine relaxation method from §7.6.2 and branch-and-bound is used to solve (SIP LLP) at the required values of $(\mathbf{x}, \boldsymbol{\mu})$.

Next consider (UBP) and (LBP). Although the function g appears in the constraints of these problems, this does not make them dynamically-constrained problems for this specific example. For this example, only the finite set of values $\{g(\mathbf{y}) : \mathbf{y} \in Y^{LBP} \cup Y^{UBP}\}$ is required at any iteration. These values can be obtained without too much extra effort in the course of solving the (SIP LLP) (since Y^{UBP} and Y^{LBP} are populated with the maximizers of the lower-level program). Meanwhile, $\mathbf{h}$ is a known, explicit function of $\mathbf{x}$ and $\mathbf{y}$, as are the objectives of (UBP) and (LBP), and overall their solution is no more arduous than in the cases considered in §9.5. The (UBP) and (LBP) are solved in GAMS with BARON. The user-defined extrinsic function capability of GAMS is used to communicate with the C++ code solving (SIP LLP).

Meanwhile, suppose that a ball-valued design space was being used in this example, and contrast this with the situation of trying to solve the SIP reformulation from Proposition 9.2.2. Since the SIP method from [122] holds for general SIP, one could attempt to solve an exact SIP reformulation similar to the form (9.3). However, the upper bounding problem for that reformulation would be

$$\begin{aligned} &\max_{\mathbf{y}_c,\delta} \delta \\ &\quad \text{s.t. } g(\mathbf{y}_c + \delta\mathbf{y}_d) \le 0, \quad \forall \mathbf{y}_d \in \widetilde{B}_1^{UBP}, \\ &\qquad (\mathbf{y}_c, \delta) \in X. \end{aligned}$$

The complication here is that the upper-level variables $(\mathbf{y}_c, \delta)$ are required to evaluate g; that is, in contrast to above, a finite set of values does not suffice, and in fact the upper and lower bounding problems become dynamic optimization problems in addition to the lower-level program. However, applying the solution method from [193] to the SIP reformulation from Proposition 9.2.2 might lead to a successful method for design centering problems when **g** is defined by the solution of a system of algebraic equations.

For the results in §9.6.3, the initial discretizations are $Y^{LBP,0} = Y^{UBP,0} = \varnothing$, the initial right-hand side restriction parameter is $\varepsilon_{R,0} = 1$, the right-hand side restriction parameter reduction factor is $r = 1.4$, and the overall relative and absolute optimality tolerances of 0.01, while the subproblems (UBP), (LBP), and (SIP LLP) are solved with relative and absolute tolerances of 10^{-5} (thus $\varepsilon_{atol} = 10^{-5}$ in (9.28) and the preceding discussion).

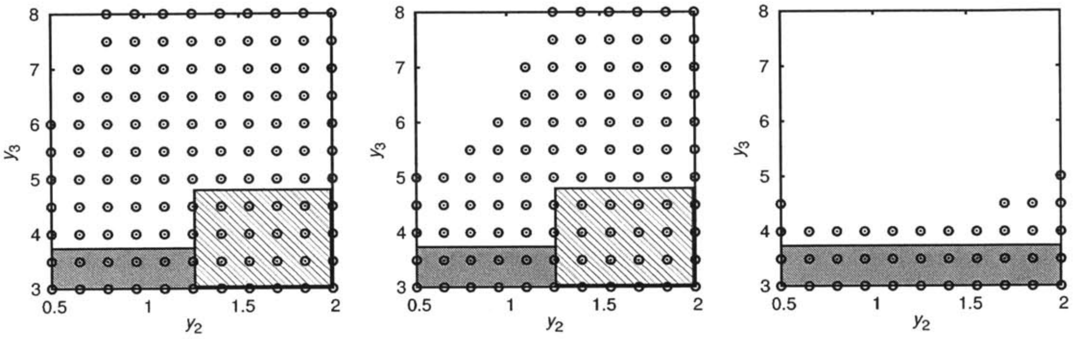
Interval restriction with branch and bound

To apply the interval restriction method from §9.6.1, Assumption 9.6.1 must be satisfied. The main challenge is defining the upper bound function g^U. In the robust design problem (9.29), g is defined in terms of the solution of an initial value problem in ordinary differential equations, and so the dynamic bounding method from [168] is used. This method provides interval bounds on each component of $\mathbf{z}(t_f, \cdot)$, the solution of the embedded differential equations. Combined with interval arithmetic, one obtains an inclusion monotonic interval extension (and thus an inclusion function) of g. Subsequently, the upper bound of this inclusion function (denote it g^U) satisfies the relevant parts of Assumption 9.6.1. A C++ implementation of the dynamic bounding method from [168] is used in conjunction with the interval arithmetic capabilities of MC++ [36, 124] to calculate the value of g^U.

The same C++ code implementing the branch-and-bound framework is used as in the previous section for solving (SIP LLP). The bounding scheme discussed in §9.6.1 is implemented with this branch-and-bound framework. For the results in §9.6.3, the relative and

Table 9.4: Results of the Interval and SIP restriction methods applied to (9.29).

Method	Solution		Objective Value	Solution Time (s)
	$\mathbf{y}^L$	$\mathbf{y}^U$		
Interval restriction (§9.6.1)	$(0.51, 1.26, 3.04)$	$(0.92, 1.99, 4.80)$	0.52	430
SIP restriction (§9.6.2)	$(0.50, 0.50, 3.00)$	$(2.0, 2.0, 3.74)$	1.66	84.6



(a) $y_1 = 0.6$ (b) $y_1 = 0.9$ (c) $y_1 = 2$

Figure 9-1: Sampling of $G = \{\mathbf{y} \in Y : g(\mathbf{y}) \leq 0\}$ for various fixed values of y_1 for problem (9.29); points in G are marked with a circle. The interval restriction solution (§9.6.1) is the striped gray box; the SIP restriction solution (§9.6.2) is the solid gray box.

absolute optimality tolerances for branch-and-bound are 0.15 and 10^{-6}, respectively.

Results and discussion

The results of the two methods applied to the robust design problem (9.29) are summarized in Table 9.4. Fig. 9-1 visualizes the results. As hinted in §9.6.1, the convergence of the branch-and-bound approach to solving the interval restriction is slow. Even with the fairly loose relative optimality tolerance of 0.15, the method takes 430 seconds to finish, requiring the solution of 218,541 lower-bounding problems. In contrast, the solution of the SIP restriction finishes in 85 seconds. This represents 7 iterations of the method from [122] (or more specifically, Algorithm 7). The dynamic optimization problem (SIP LLP) is solved 13 times. The total time required to solve the lower-level programs is 75.5 seconds, representing a majority of the effort in solving the SIP restriction.

The SIP restriction method attains a better optimal objective value than the interval restriction method by a factor of three, and in a fifth of the time. However, careful inspection of the dynamic bounding method used to construct g^U reveals that the interval restriction method is in fact a restriction for the problem

$$\begin{aligned}
\max_{\mathbf{y}^L,\mathbf{y}^U} \; & \prod_j (y_j^U - y_j^L) && (9.30)\\
\text{s.t. } & \frac{z_{\mathrm{D}}(t_f,\mathbf{y})}{\mathbf{1}^{\mathrm{T}}\mathbf{z}(t_f,\mathbf{y})} - 0.05 \le 0, \quad \forall (y_1,y_2,y_3) \in [y_1^L, y_1^U] \times [y_2^L, y_2^U] \times \mathcal{U}(y_3^L, y_3^U),\\
& \mathbf{y}^L \le \mathbf{y}^U,\\
& \mathbf{y}^L, \mathbf{y}^U \in [0.5, 2] \times [0.5, 2] \times [3, 8],\\
& \mathcal{U}(y_3^L, y_3^U) = \left\{ u \in L^1([t_0,t_f],\mathbb{R}) : u(t) \in [y_3^L, y_3^U], a.e.\ t \in [t_0, t_f] \right\}.
\end{aligned}$$

In words, one is looking for the largest acceptable ranges for the initial concentrations of A and B and range for the temperature *control profile* to ensure that the mole fraction of D is below the threshold. This is due to the fact that the bounding method from [168] indeed produces an interval $[\mathbf{z}^L(\mathbf{y}^L,\mathbf{y}^U), \mathbf{z}^U(\mathbf{y}^L,\mathbf{y}^U)]$ satisfying

$$\mathbf{z}(t_f,\mathbf{y}) \in [\mathbf{z}^L(\mathbf{y}^L,\mathbf{y}^U), \mathbf{z}^U(\mathbf{y}^L,\mathbf{y}^U)], \quad \forall \mathbf{y} \in [y_1^L, y_1^U] \times [y_2^L, y_2^U] \times \mathcal{U}(y_3^L, y_3^U).$$

Continuing the evaluation of g in interval arithmetic then guarantees that $g(\mathbf{y}) \le g^U([\mathbf{y}^L, \mathbf{y}^U])$ for all $\mathbf{y} \in [y_1^L, y_1^U] \times [y_2^L, y_2^U] \times \mathcal{U}(y_3^L, y_3^U)$. Thus the restriction (9.27) still holds.

One could argue that problem (9.30) is a somewhat harder problem than (9.29), and thus the increased computational time required and more conservative solution produced by the interval restriction method is acceptable in this case.

9.7 Conclusions and future work

This work has discussed a number of approaches to solving design centering problems, motivated by the specific instance of robust design in engineering applications. Reformulations to simpler problems were reviewed; many of these are inspired by duality-based reformulations from the GSIP literature. Two approaches for determining a feasible solution of a design centering problem were discussed and applied to an engineering application of robust design. The two methods are successful, with the SIP restriction-based approach (§9.6.2) performing better for this example.

One aspect of the restriction-based approaches in §9.6 that was not considered was the case of multiple infinite constraints (i.e. multiple LLPs). The interval restriction (§9.6.1) can likely handle multiple constraints in practice without much numerical difficulty. Meanwhile, [122] mentions a potential way to extend the basic SIP algorithm; this simply depends on using separate discretization sets and restriction parameters for each LLP. Unfortunately, numerical experiments show that such an extension to the method performs poorly for design centering problems. The lower-bounding problem (LBP) has trouble finding a feasible point, and so the algorithm is slow to converge. Extending the SIP method from [122] and the GSIP method [127] merits further investigation.

9.8 Convergence of bounding method from §9.6.1

Let $[\mathbf{v}^U, \mathbf{w}^L] \subset [\mathbf{v}^L, \mathbf{w}^U] \subset [\bar{\mathbf{y}}^L, \bar{\mathbf{y}}^U] \subset \mathbb{R}^{n_y}$. Let

$$\begin{aligned}
Y_j^v &= [v_1^L, w_1^U] \times \cdots \times [v_{j-1}^L, w_{j-1}^U] \times [v_j^L, v_j^U] \times [v_{j+1}^L, w_{j+1}^U] \times \cdots \times [v_{n_y}^L, w_{n_y}^U],\\
Y_j^w &= [v_1^L, w_1^U] \times \cdots \times [v_{j-1}^L, w_{j-1}^U] \times [w_j^L, w_j^U] \times [v_{j+1}^L, w_{j+1}^U] \times \cdots \times [v_{n_y}^L, w_{n_y}^U],
\end{aligned}$$

for each $j \in \{1, \dots, n_y\}$. Then it is easy to see that the "outer" interval $[\mathbf{v}^L, \mathbf{w}^U]$ is a subset of $[\mathbf{v}^U, \mathbf{w}^L] \cup \left(\bigcup_j (Y_j^v \cup Y_j^w) \right)$; for any $\mathbf{y} \in [\mathbf{v}^L, \mathbf{w}^U]$, each component y_j lies in one of $[v_j^L, v_j^U]$, $[v_j^U, w_j^L]$, or $[w_j^L, w_j^U]$, which is included in the definition of one of Y_j^v, $[\mathbf{v}^U, \mathbf{w}^L]$, or Y_j^w.

Thus

$$\begin{aligned}
\mathrm{vol}([\mathbf{v}^L, \mathbf{w}^U]) &\le \mathrm{vol}([\mathbf{v}^U, \mathbf{w}^L]) + \sum_j \mathrm{vol}(Y_j^v) + \mathrm{vol}(Y_j^w)\\
&= \mathrm{vol}([\mathbf{v}^U, \mathbf{w}^L]) + \sum_j ((v_j^U - v_j^L) + (w_j^U - w_j^L)) \prod_{k \ne j} (w_k^U - v_k^L).
\end{aligned}$$

Let $\alpha = \mathrm{diam}([\bar{\mathbf{y}}^L, \bar{\mathbf{y}}^U])$. Then $(w_k^U - v_k^L) \le \alpha$ for each k. If $X' = [\mathbf{v}^L, \mathbf{v}^U] \times [\mathbf{w}^L, \mathbf{w}^U]$, then $(v_j^U - v_j^L) + (w_j^U - w_j^L) \le 2\,\mathrm{diam}(X')$ for each j. Putting all these inequalities together one obtains

$$\mathrm{vol}([\mathbf{v}^L, \mathbf{w}^U]) - \mathrm{vol}([\mathbf{v}^U, \mathbf{w}^L]) \le 2 n_y \alpha^{n_y - 1}\,\mathrm{diam}(X').$$

Thus assuming $[\mathbf{y}^L, \mathbf{y}^U] \subset [\bar{\mathbf{y}}^L, \bar{\mathbf{y}}^U]$ for all $(\mathbf{y}^L, \mathbf{y}^U) \in X$, this establishes that the bounding method described in §9.6.1 is at least first-order convergent.

Chapter 10

Conclusions

10.1 Summary

This thesis has considered the problems of forward reachability and robust design in dynamic systems. Overall, the approaches taken are motivated by or involve deterministic global dynamic optimization. In specific, the direct method of control parameterization for global dynamic optimization is one of the main applications of the methods for enclosing, or bounding, the reachable set developed in this thesis. Consequently, these methods aim to be efficiently implementable. This has been achieved by developing auxiliary initial value problems in ordinary differential equations that are sufficiently regular to be amenable to numerical solution with established methods for numerical integration. In some chemical engineering applications, taking advantage of model or overall problem structure can improve the quality of the bounds without incurring a significant extra computational cost.

Meanwhile, the theoretical and numerical approaches taken to robust design in dynamic systems are inspired by the semi-infinite programming literature. Numerical methods for solving semi-infinite programs have been employed, but the dynamic nature of the problem requires the solution of global dynamic optimization problems or the application of related techniques. The work on the forward reachability problem thus has been critical to these approaches.

10.2 Future work

There are a few areas that merit further investigation. First, the work on ordinary differential equations with a linear program embedded in Ch. 4 has already been extended in [60]. That work even more robustly addresses the domain issues by adding slack variables to the embedded linear program to ensure that it is always feasible. The first objective in the hierarchy of objectives is to minimize the sum of these slack variables, so that the original feasible set (if indeed nonempty) is regained. This is similar in effect to the Phase I method of initializing primal simplex.

The problem of ODEs with LPs embedded could be applied to the numerical method from Ch. 6. A slight modification of the ODEs with LPs embedded formulation is required, to allow a state-dependent cost vector. However, domain issues effectively do not appear in the method for constructing polyhedral bounds from Ch. 6. Further, the alternative numerical method from §7.6.1 avoids the solution of linear programs without a significant loss in the quality of the bounds.

The theory in Ch. 6 extends almost directly to abstract equations of evolution in general normed spaces (see Appendix A). A direction for future research is establishing the usefulness of this generalization; at present, it appears to apply to specific classes of partial differential equations, although further generalization may establish its usefulness for more challenging classes of engineering problems.

The generality of the bounding theory in Ch. 7 is exciting. Future work could entail developing a more tailored numerical method for constructing relaxations by the method in §7.6.2; by taking advantage of the structure of the ordinary differential equations in Proposition 7.6.2, a more efficient numerical integration scheme is likely, along the lines of a staggered corrector in sensitivity analysis [50].

The approaches to solving robust design problems from Ch. 9 show promise and applicability to other problems that are essentially a semi-infinite program constrained by a dynamic system, such as flexibility analysis of dynamic systems as in [43]. Most likely, future work would focus on the specifics of numerical implementations. For instance, reformulation of a dynamic robust design problem to an SIP along the lines of Proposition 9.2.2 and then application of an SIP solution method (such as the one in [122]) would require the iterative solution of global dynamic optimization problems. Perhaps more fundamental is to adapt the SIP solution method from [122] to handle multiple infinite constraints in a more efficient or robust way. As mentioned in §9.7, the solution method from §8.5 (which relates to the SIP solution method from [122]) seems to perform poorly for design centering problems with multiple constraints.

Appendix A

Polyhedral bounds for Banach space ordinary differential equations

A.1 Introduction

This appendix generalizes the theory from Ch. 6 to equations of evolution, specifically initial value problems (IVPs) in ordinary differential equations (ODEs), in Banach spaces. Applications include certain classes of partial differential equations (PDEs); the most obvious case is linear second-order parabolic PDEs, with the theory developed in Ch. 7 of [49]. Consequently, it also seems that this could apply to certain stochastic differential equations through the Fokker-Planck partial differential equation, which is a parabolic PDE.

The bounds are in the form of bounds on linear functionals, and so can be considered "polyhedral" bounds (more discussion in §A.2). Applied to PDEs, this means that we could obtain, for instance, pointwise in time bounds on the (spatial) Fourier coefficients or moments of the solution. However, §A.4.2 provides some insight on how to choose the linear functionals which are to be bounded in the special case that the dynamics are linear with respect to the differential variables. In this case, a (relatively simple) initial value problem in (finite) ODEs can be solved to give the bounds.

A.2 Problem statement

First, a (somewhat pedantic) discussion of notation. All vector spaces in this chapter will be over $\mathbb{R}$. Typically, lowercase bold letters denote vectors and vector-valued mappings. The exception is for linear mappings/operators; scalar-valued linear mappings are denoted by lowercase bold letters (seen as vectors in a dual space, defined below), while vector-valued linear mappings/operators are denoted by uppercase bold letters. As usual the value of an operator $\mathbf{M}$ at a point $\mathbf{v}$ is denoted $\mathbf{Mv}$. For a normed (vector) space V, its norm is denoted $\|\cdot\|_V$, or if there is no confusion, by just $\|\cdot\|$. The dual of a normed space V (space of all continuous linear real-valued mappings on V) is denoted V^*, and similarly the dual norm on V^* is denoted $\|\cdot\|_{V^*}$ or $\|\cdot\|_*$. The scalar product between V^* and V is denoted $\langle\cdot,\cdot\rangle$; i.e. for $\mathbf{a} \in V^*$ and $\mathbf{v} \in V$, $\langle\mathbf{a},\mathbf{v}\rangle$ is $\mathbf{a}$ evaluated at $\mathbf{v}$. For normed spaces V and W with $V \subset W$, V is embedded in W if there exists a scalar $c > 0$ such that $\|\mathbf{v}\|_W \leq c\|\mathbf{v}\|_V$ for all $\mathbf{v} \in V$. In this case we write $V \hookrightarrow W$, and W^* can be considered a subset of V^*; if $a : W \to \mathbb{R}$ is a bounded linear mapping, then for some $b > 0$, $|a(\mathbf{v})| \leq b\|\mathbf{v}\|_W$ and so $|a(\mathbf{v})| \leq bc\|\mathbf{v}\|_V$, and thus is a bounded linear mapping on V as well. Thus if $\mathbf{a} \in W^*$, $\langle\mathbf{a},\mathbf{v}\rangle$ makes sense for $\mathbf{v} \in V$, and has the same value in $\mathbb{R}$ regardless of whether it is interpreted as the scalar product between W^* and W, or V^* and V.

A polyhedron is any subset of a normed space V that can be written as $\{\mathbf{v} \in V : \langle\mathbf{a}_i,\mathbf{v}\rangle \leq d_i, i \in \{1,\dots,m\}\}$, for some $m \in \mathbb{N}$, subset of the dual space $\{\mathbf{a}_1,\dots,\mathbf{a}_m\} \subset V^*$, and $\mathbf{d} \in \mathbb{R}^m$. Thus, a polyhedron is the intersection of a finite number of closed halfspaces, and so polyhedra are always closed, convex sets. Denote the Bochner space $L^1(T,V)$ (the space of Bochner/strongly-measurable mappings $\mathbf{v} : T \to V$ with $\int_T \|\mathbf{v}\|_V < +\infty$, for more background see Appendix E in [49]). For $T \subset \mathbb{R}$, normed spaces V and W with $V \hookrightarrow W$, and mapping $\mathbf{x} : T \to V$, we say that $\mathbf{x}$ has strong derivative $\dot{\mathbf{x}}(t)$ in W at t in the interior of T if $\lim_{h\to 0} \|(\mathbf{x}(t+h)-\mathbf{x}(t))/h - \dot{\mathbf{x}}(t)\|_W = 0$.

The problem statement follows. Let V_x, W_x, V_u be real Banach spaces with $V_x \hookrightarrow W_x$. Let nonempty interval $T = [t_0, t_f] \subset \mathbb{R}$, $D_x \subset V_x$, and $D_u \subset V_u$ be given. For $U \subset D_u$, let the set of time-varying inputs be

$$\mathcal{U} = \left\{\mathbf{u} \in L^1(T,V_u) : \mathbf{u}(t) \in U, a.e.\ t \in T\right\},$$

and let the set of possible initial conditions be $X_0 \subset D_x$. Given $\mathbf{f} : T \times D_u \times D_x \to W_x$, the problem of interest is the initial value problem in ODEs

$$\dot{\mathbf{x}}(t, \mathbf{u}) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t, \mathbf{u})), \quad a.e.\ t \in T, \tag{A.1a}$$

$$\mathbf{x}(t_0, \mathbf{u}) \in X_0. \tag{A.1b}$$

For a given $\mathbf{u} \in \mathcal{U}$, a solution is a mapping $\mathbf{x}(\cdot, \mathbf{u}) : T \to D_x$ with strong derivative $\dot{\mathbf{x}}(t, \mathbf{u})$ in W_x at almost every $t \in T$, such that $\mathbf{x}(t, \mathbf{u})$ and $\dot{\mathbf{x}}(t, \mathbf{u})$ satisfy Equations (A.1). The goal of this work is to construct a polyhedral-valued mapping $B : T \rightrightarrows V_x$ such that $\mathbf{x}(t, \mathbf{u}) \in B(t)$, for all $(t, \mathbf{u}) \in T \times \mathcal{U}$. Specifically, given a $m \in \mathbb{N}$ and $\{\mathbf{a}_1, \dots, \mathbf{a}_m\} \subset V_x^*$, the goal is to find $\mathbf{b} : T \to \mathbb{R}^m$ such that $B(t) = \{\mathbf{z} : \langle \mathbf{a}_i, \mathbf{z} \rangle \leq b_i(t), \forall i\}$. This mapping B will be referred to as polyhedral bounds, or just bounds.

It should be noted that the notion of a solution above is slightly weaker than normal. Typically, for $\mathbf{u} \in \mathcal{U}$, a solution is a mapping $\mathbf{x}(\cdot, \mathbf{u}) : T \to D_x$ such that $\mathbf{x}(\cdot, \mathbf{u}) \in L^1(T, V_x)$ with *weak* derivative $\dot{\mathbf{x}}(\cdot, \mathbf{u}) \in L^1(T, W_x)$ which satisfy Equations (A.1). However in this case, the strong derivative of $\mathbf{x}(\cdot, \mathbf{u})$ exists in W_x at almost every t and coincides with the value of the weak derivative $\dot{\mathbf{x}}(t, \mathbf{u})$. See [49] and the appendix of Ch. 6 of [83], specifically Theorem 6.35.

A.3 Bounding theory

This section presents the general bounding theory. The following section focuses on how to implement this theory.

Lemma A.3.1. *Let $T \subset \mathbb{R}$ be a nonempty interval, V and W be real normed spaces with $V \hookrightarrow W$, $b : T \to \mathbb{R}$ be absolutely continuous, $\mathbf{x} : T \to V$ with strong derivative $\dot{\mathbf{x}}(t)$ in W for almost every $t \in T$, and $\mathbf{a} \in W^*$. Then the real-valued function $g : t \mapsto \max\{0, \langle \mathbf{a}, \mathbf{x}(t) \rangle - b(t)\}$ is absolutely continuous. Further, for almost all t such that $\langle \mathbf{a}, \mathbf{x}(t) \rangle > b(t)$ and for all $\mathbf{w} \in W$ such that $\langle \mathbf{a}, \mathbf{w} \rangle \leq \dot{b}(t)$,*

$$\dot{g}(t) \leq \|\mathbf{a}\|_{W^*} \|\mathbf{w} - \dot{\mathbf{x}}(t)\|_W .$$

Proof. Note that $g_1 : t \mapsto \langle \mathbf{a}, \mathbf{x}(t) \rangle - b(t)$ is absolutely continuous. To see this, note that for almost all t, $(\mathbf{x}(t+h)-\mathbf{x}(t))/h$ converges to $\dot{\mathbf{x}}(t)$ (in W), and so

$$\lim_{h\to 0}\left\langle \mathbf{a}, \frac{\mathbf{x}(t+h)-\mathbf{x}(t)}{h} - \dot{\mathbf{x}}(t)\right\rangle = 0,$$

which, upon rearrangement, yields

$$\lim_{h\to 0}\frac{\langle \mathbf{a}, \mathbf{x}(t+h)\rangle - \langle \mathbf{a}, \mathbf{x}(t)\rangle}{h} = \langle \mathbf{a}, \dot{\mathbf{x}}(t)\rangle .$$

Thus the real-valued function $\langle \mathbf{a}, \mathbf{x}(\cdot)\rangle$ has a derivative almost everywhere and thus is absolutely continuous. Subtract the absolutely continuous function b and we see that g_1 is absolutely continuous. Obviously, $g_2 : t \mapsto 0$ is absolutely continuous, and so g, as the maximum of the two, can be written as $g(t) = {}^{1}\!/_{2}(g_1(t) + g_2(t) + |g_1(t) - g_2(t)|)$. We note this is absolutely continuous since the composition of a Lipschitz continuous function with an absolutely continuous function is absolutely continuous, and again the sum of absolutely continuous functions is absolutely continuous.

Consequently, for almost all t such that $\langle \mathbf{a}, \mathbf{x}(t)\rangle > b(t)$, $g(t) = \langle \mathbf{a}, \mathbf{x}(t)\rangle - b(t)$, and so $\dot{g}(t) = \langle \mathbf{a}, \dot{\mathbf{x}}(t)\rangle - \dot{b}(t)$. Thus, for any $\mathbf{w}$ such that $\langle \mathbf{a}, \mathbf{w}\rangle \leq \dot{b}(t)$, $\dot{g}(t) + \langle \mathbf{a}, \mathbf{w}\rangle \leq \langle \mathbf{a}, \dot{\mathbf{x}}(t)\rangle - \dot{b}(t) + \dot{b}(t)$. It follows that $\dot{g}(t) \leq \langle \mathbf{a}, \dot{\mathbf{x}}(t)\rangle - \langle \mathbf{a}, \mathbf{w}\rangle$ and so $\dot{g}(t) \leq \langle \mathbf{a}, \dot{\mathbf{x}}(t) - \mathbf{w}\rangle$. Finally, from the generalization of the Cauchy-Schwarz inequality (that is, from the definition of the dual norm), we have $\dot{g}(t) \leq \|\mathbf{a}\|_* \|\dot{\mathbf{x}}(t) - \mathbf{w}\|$. □

Assumptions A.3.1 and A.3.2 and Theorem A.3.1 below provide the heart of the general bounding theory.

Assumption A.3.1. *For any $\mathbf{z} \in D_x$, there exists a neighborhood $N(\mathbf{z})$ and $\alpha \in L^1(T, \mathbb{R})$ such that for almost every $t \in T$ and every $\mathbf{p} \in U$*

$$\|\mathbf{f}(t, \mathbf{p}, \mathbf{z}_1) - \mathbf{f}(t, \mathbf{p}, \mathbf{z}_2)\|_{W_x} \leq \alpha(t)\, \|\mathbf{z}_1 - \mathbf{z}_2\|_{V_x},$$

for every $\mathbf{z}_1$ and $\mathbf{z}_2$ in $N(\mathbf{z}) \cap D_x$.

Assumption A.3.2. *Consider the problem stated in §A.2. For $m \in \mathbb{N}$, let $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\} \subset$*

W_x^*. Assume that the linear operator

$$\mathbf{A} : W_x \ni \mathbf{z} \mapsto \begin{bmatrix} \langle \mathbf{a}_1, \mathbf{z} \rangle \\ \vdots \\ \langle \mathbf{a}_m, \mathbf{z} \rangle \end{bmatrix} \in \mathbb{R}^m,$$

$D_M \subset T \times \mathbb{R}^m$, and $M_i : D_M \rightrightarrows V_x$ satisfy the following conditions for each $i \in \{1, \ldots, m\}$.

1. *For any $\mathbf{d} \in \mathbb{R}^m$, if there exists $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\mathbf{A}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\langle \mathbf{a}_i, \mathbf{x}(t, \mathbf{u}) \rangle = d_i$ for some solution $\mathbf{x}(\cdot, \mathbf{u})$ of IVP* (A.1)*, then $(t, \mathbf{d}) \in D_M$ and $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$.*
2. *For any $(t, \mathbf{d}) \in D_M$, there exists a neighborhood $N(\mathbf{d})$ of $\mathbf{d}$, $t' > t$, and $L_M > 0$ such that for any $(s, \mathbf{d}_1)$ and $(s, \mathbf{d}_2)$ in $((t, t') \times N(\mathbf{d})) \cap D_M$ and $\mathbf{z}_1 \in M_i(s, \mathbf{d}_1)$, there exists a $\mathbf{z}_2 \in M_i(s, \mathbf{d}_2)$ such that*

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_{V_x} \leq L_M \|\mathbf{d}_1 - \mathbf{d}_2\|_1 .$$

Theorem A.3.1. *Let Assumptions A.3.1 and A.3.2 hold. If*

1. *$\mathbf{b} : T \to \mathbb{R}^m$ is absolutely continuous and $B : T \ni t \mapsto \{\mathbf{z} \in V_x : \mathbf{A}\mathbf{z} \leq \mathbf{b}(t)\}$,*
2. *$X_0 \subset B(t_0)$,*
3. *for almost every $t \in T$ and each $i \in \{1, \ldots, m\}$, $(t, \mathbf{b}(t)) \in D_M$ and $M_i(t, \mathbf{b}(t)) \subset D_x$,*
4. *for almost every $t \in T$ and each $i \in \{1, \ldots, m\}$,*

$$\langle \mathbf{a}_i, \mathbf{f}(t, \mathbf{p}, \mathbf{z}) \rangle \leq \dot{b}_i(t), \quad \forall (\mathbf{p}, \mathbf{z}) \in U \times M_i(t, \mathbf{b}(t)),$$

then for all $\mathbf{u} \in \mathcal{U}$ and any solution $\mathbf{x}(\cdot, \mathbf{u})$ of IVP (A.1)*, $\mathbf{x}(t, \mathbf{u}) \in B(t)$, for all $t \in T$.*

Proof. Fix $\mathbf{u} \in \mathcal{U}$. If no solution of IVP (A.1) exists for this $\mathbf{u}$, then the conclusion of the theorem holds trivially. Otherwise, choose some solution and for convenience use the abbreviation $\mathbf{x}(t) \equiv \mathbf{x}(t, \mathbf{u})$. For each $t \in T$ and $i \in \{1, \ldots, m\}$, let $g_i(t) = \max\{0, \langle \mathbf{a}_i, \mathbf{x}(t) \rangle - b_i(t)\}$. By Lemma A.3.1, each g_i is absolutely continuous. It follows that $\mathbf{A}\mathbf{x}(t) \leq \mathbf{b}(t) + \mathbf{g}(t)$. Consequently, $\mathbf{g}(t) = \mathbf{0}$ implies $\mathbf{x}(t) \in B(t)$, and by the contrapositive $\mathbf{x}(t) \notin B(t)$ implies $\mathbf{g}(t) \neq \mathbf{0}$. Thus, for a contradiction, assume that the set $T_v = \{t \in T : \|\mathbf{g}(t)\|_1 > 0\}$ is nonempty.

Let $t_1 = \inf T_v$. By Hypothesis 2, $\mathbf{g}(t_0) = \mathbf{0}$ and so by continuity of $\mathbf{g}$, $\|\mathbf{g}(t_1)\|_1 = 0$. Furthermore, there exists $t_2 > t_1$ and index set I such that $g_i(t) = 0$ for $i \notin I$ and $t \in [t_1, t_2)$, and $\langle \mathbf{a}_i, \mathbf{x}(t)\rangle = b_i(t) + g_i(t)$ for $i \in I$ and $t \in [t_1, t_2)$. Explicitly, for each i define $T_i \equiv \{t : g_i(t) > 0\}$. By continuity of $\mathbf{g}$, each T_i is open. Let $I = \{i : t_1 = \inf T_i\}$ (which must be nonempty) and then choose $t_2 > t_1$ such that $(t_1, t_2) \subset \bigcap_{i \in I} T_i$ and $(t_1, t_2) \cap (\bigcup_{i \notin I} T_i) = \varnothing$.

Then by Condition 1 of Assumption A.3.2, $(t, \mathbf{b}(t) + \mathbf{g}(t)) \in D_M$ and $\mathbf{x}(t) \in M_i(t, \mathbf{b}(t) + \mathbf{g}(t))$ for $i \in I$, $t \in [t_1, t_2)$. Without loss of generality, let $N(\mathbf{b}(t_1))$, $t_3 > t_1$, and $L_M > 0$ satisfy Condition 2 of Assumption A.3.2 at the point $\mathbf{b}(t_1)$, for each $i \in I$. Since $\mathbf{b}$ and $\mathbf{g}$ are continuous, there exists a $t_4 \in (t_1, \min\{t_2, t_3\})$ such that $\mathbf{b}(t)$ and $(\mathbf{b}(t) + \mathbf{g}(t)) \in N(\mathbf{b}(t_1))$ for each $t \in (t_1, t_4)$. Along with Hypothesis 3, it follows that for $i \in I$ and almost every $t \in (t_1, t_4)$, there exists an element $\mathbf{z}_i(t) \in M_i(t, \mathbf{b}(t))$ with

$$\|\mathbf{z}_i(t) - \mathbf{x}(t)\|_{V_x} \leq L_M \|\mathbf{g}(t)\|_1 . \tag{A.2}$$

Let $N(\mathbf{x}(t_1))$, and $\alpha \in L^1(T, \mathbb{R})$ satisfy Assumption A.3.1 at the point $\mathbf{x}(t_1)$. Since $\mathbf{x}$ and $\|\mathbf{g}\|_1$ are continuous, using Inequality (A.2) and the triangle inequality

$$\|\mathbf{z}_i(t) - \mathbf{x}(t_1)\| \leq \|\mathbf{z}_i(t) - \mathbf{x}(t)\| + \|\mathbf{x}(t) - \mathbf{x}(t_1)\|,$$

there exists a $t_5 \in (t_1, t_4)$ such that $\mathbf{z}_i(t), \mathbf{x}(t) \in N(\mathbf{x}(t_1))$, for all $i \in I$ and almost every $t \in (t_1, t_5)$. Consequently,

$$\|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t))\|_{W_x} \leq \alpha(t) \|\mathbf{z}_i(t) - \mathbf{x}(t)\|_{V_x}, \ a.e.\ t \in (t_1, t_5). \tag{A.3}$$

But by Hypothesis 4, $\langle \mathbf{a}_i, \mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t))\rangle \leq \dot{b}_i(t)$ which by Lemma A.3.1 means

$$\dot{g}_i(t) \leq \|\mathbf{a}_i\|_{W_x^*} \|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \dot{\mathbf{x}}(t)\|_{W_x} = \|\mathbf{a}_i\|_{W_x^*} \|\mathbf{f}(t, \mathbf{u}(t), \mathbf{z}_i(t)) - \mathbf{f}(t, \mathbf{u}(t), \mathbf{x}(t))\|_{W_x} .$$

Combining this with Inequalities (A.2) and (A.3) we have

$$\dot{g}_i(t) \leq L_M \alpha(t) \|\mathbf{a}_i\|_{W_x^*} \|\mathbf{g}(t)\|_1, a.e.\ t \in (t_1, t_5).$$

Since this holds for each $i \in I$ and $g_i(t) = 0$ for each $i \notin I$,

$$\sum_{i \in I} \dot{g}_i(t) \leq L_M \alpha(t) \sum_{i \in I} \|\mathbf{a}_i\|_{W_x^*} \|\mathbf{g}(t)\|_1 = L_M \alpha(t) \sum_{j \in I} \|\mathbf{a}_j\|_{W_x^*} \sum_{i \in I} g_i(t)$$

to which we can apply Gronwall's inequality (see for instance [209]) to get

$$\sum_{i\in I} g_i(t) \leq \sum_{i\in I} g_i(t_1) \exp\left(\int_{[t_1,t]} L_M \sum_{j\in I} \|\mathbf{a}_j\|_{W_x^*} |\alpha|\right), \forall t \in [t_1, t_5].$$

But since $\sum_i g_i(t_1) = 0$, this yields $\sum_i g_i(t) \leq 0$, and since each g_i is nonnegative always and $g_i(t) = 0$ for each $i \notin I$, we have $g_i(t) = 0$ for all i and all $t \in (t_1, t_5) \subset T_v$, which is a contradiction. Since the choices of $\mathbf{u} \in \mathcal{U}$ and corresponding solution were arbitrary, the result follows. □

A.4 Specific instances

This section describes how to construct the mappings M_i such that they satisfy Assumption A.3.2.

A.4.1 Utilizing *a priori* information

The goal of this section is to define mappings M_i, which, as in [74, 168], allow one to use *a priori* information about the solution set of IVP (A.1) in the form of a polyhedral-valued mapping $G : T \rightrightarrows V_x$ for which it is known that $\mathbf{x}(t, \mathbf{u}) \in G(t)$, for all $t \in T$ and $\mathbf{u} \in \mathcal{U}$ for which a solution exists. The specific conditions are formalized in the following assumption.

Assumption A.4.1. *For $m_g \in \mathbb{N}$, let $\{\mathbf{g}_1, \ldots, \mathbf{g}_{m_g}\} \subset V_x^*$ and $\mathbf{b}_G : T \to \mathbb{R}^{m_g}$. Define $\mathbf{G} : V_x \to \mathbb{R}^{m_g}$ by $\mathbf{G}\mathbf{z} = (\langle \mathbf{g}_1, \mathbf{z}\rangle, \ldots, \langle \mathbf{g}_{m_g}, \mathbf{z}\rangle)$. Assume that for all $\mathbf{u} \in \mathcal{U}$ and any solution $\mathbf{x}(\cdot, \mathbf{u})$ of IVP* (A.1)*, $\mathbf{G}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{b}_G(t)$, for all $t \in T$.*

Before a specific form of the M_i can be defined, some results are needed. First, a decomposition result for Banach spaces is noted.

Lemma A.4.1. *Let V be a real Banach space. For $m \in \mathbb{N}$, let $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\} \subset V^*$. Then there exists $p \in \mathbb{N}$ and $\{\mathbf{e}_1, \ldots, \mathbf{e}_p\} \subset V$ such that for all $\mathbf{v} \in V$ there exists $\mathbf{w} \in V$ and $\mathbf{c} \in \mathbb{R}^p$ such that $\mathbf{v} = \mathbf{w} + \sum_{j=1}^{p} c_j \mathbf{e}_j$ and $\langle \mathbf{a}_i, \mathbf{w}\rangle = 0$ for all i.*

If, in addition, $\mathbf{a}_i \neq \mathbf{0}$ for all i, then for all i there exists j_i such that $\langle \mathbf{a}_i, \mathbf{e}_{j_i}\rangle \neq 0$.

Proof. Let $A = \operatorname{span}\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$, a finite dimensional and thus closed subspace of V^*. Thus $W = A^{\perp} = \{\mathbf{v} \in V : \langle \mathbf{a}, \mathbf{v}\rangle = 0, \forall \mathbf{a} \in A\}$ has finite codimension, by Proposition 11.14

in [35], for example. By definition of finite codimension (see discussion preceding Proposition 11.5 in [35]), there exists a finite dimensional subspace $E \subset V$ such that $W + E = V$ and $W \cap E = \{\mathbf{0}\}$; i.e. for any $\mathbf{v} \in V$, there exist $\mathbf{w} \in W$ and $\mathbf{x} \in E$ such that $\mathbf{v} = \mathbf{w} + \mathbf{x}$. Further, since E is finite dimensional, there exists a finite basis for it $\{\mathbf{e}_1, \dots, \mathbf{e}_p\}$, so that $\mathbf{x} = \sum_i c_i \mathbf{e}_i$ for scalars c_i. Further, by definition of W, $\langle \mathbf{a}_i, \mathbf{w} \rangle = 0$ for all i.

For the final claim, we somewhat inelegantly append a finite number of vectors $\mathbf{e}'_i$ to the set $\{\mathbf{e}_1, \dots, \mathbf{e}_p\}$ so that $\langle \mathbf{a}_i, \mathbf{e}'_i \rangle \neq 0$ for each i. The previous claim is unaffected; for any $\mathbf{v} \in V$ the same $\mathbf{w} \in W$ and $\mathbf{x} \in E$ can be chosen as before, and then we have $\langle \mathbf{a}_i, \mathbf{w} \rangle = 0$ and $\mathbf{x} = \sum_i c_i \mathbf{e}_i + \sum_i 0 \mathbf{e}'_i$. In detail, choose $i \in \{1, \dots, m\}$ and let $A_i = \operatorname{span}\{\mathbf{a}_i\}$. Similarly to before, let $W_i = A_i^{\perp}$. Again, W_i has finite codimension so there exists finite dimensional subspace $E_i \subset V$ such that $W_i + E_i = V$ and $W_i \cap E_i = \{\mathbf{0}\}$. More specifically, the dimension of A_i equals the codimension of W_i equals the dimension of E_i (again, by Proposition 11.14 and discussion in §11.1 in [35]; see also Example 2 in §2.4 in [35]). Thus, if $\mathbf{a}_i \neq \mathbf{0}$, the dimension of A_i and thus the dimension of E_i equals one. Consequently we can assume that there exists $\mathbf{e}'_i \neq \mathbf{0}$ such that $E_i = \operatorname{span}\{\mathbf{e}'_i\}$. We must have $\langle \mathbf{a}_i, \mathbf{e}'_i \rangle \neq 0$; otherwise for all $\alpha \in \mathbb{R}$ we have $\alpha \langle \mathbf{a}_i, \mathbf{e}'_i \rangle = \langle \alpha \mathbf{a}_i, \mathbf{e}'_i \rangle = 0$ which implies $\mathbf{e}'_i \in A_i^{\perp} = W_i$, which contradicts that $W_i \cap E_i = \{\mathbf{0}\}$. □

The following result generalizes the "Lipschitz continuity" of polyhedra with respect to their right-hand sides in finite dimensional spaces.

Lemma A.4.2. *Let V be a real Banach space. For $m \in \mathbb{N}$, let $\{\mathbf{a}_1, \dots, \mathbf{a}_m\} \subset V^*$. For $\mathbf{b} \in \mathbb{R}^m$, let*

$$P(\mathbf{b}) = \{\mathbf{v} \in V : \langle \mathbf{a}_i, \mathbf{v} \rangle \leq b_i, \forall i\}.$$

Then there exists $L \geq 0$ such that for all $(\mathbf{b}, \mathbf{b}') \in \mathbb{R}^m \times \mathbb{R}^m$ such that $P(\mathbf{b})$ and $P(\mathbf{b}')$ are nonempty and all $\mathbf{v} \in P(\mathbf{b})$, there exists a $\mathbf{v}' \in P(\mathbf{b}')$ such that

$$\left\|\mathbf{v} - \mathbf{v}'\right\|_V \leq L \left\|\mathbf{b} - \mathbf{b}'\right\|_1 .$$

Proof. We use the decomposition result from Lemma A.4.1; we can assume that there exist $\{\mathbf{e}_1, \dots, \mathbf{e}_p\} \subset V$ such that for any $\mathbf{v} \in V$, there exists a $\mathbf{w} \in V$ and $\mathbf{c} \in \mathbb{R}^p$ with $\langle \mathbf{a}_i, \mathbf{w} \rangle = 0$ for each i and $\mathbf{v} = \mathbf{w} + \sum_j c_j \mathbf{e}_j$.

Now, choose $\mathbf{b}$ and $\mathbf{b}'$ such that $P(\mathbf{b})$ and $P(\mathbf{b}')$ are nonempty. Choose $\mathbf{v} \in P(\mathbf{b})$. By the above discussion, $\mathbf{v} = \mathbf{w} + \sum_j c_j \mathbf{e}_j$ for $\mathbf{c} \in \mathbb{R}^p$ and $\mathbf{w} \in W$. Since $\mathbf{v}$ satisfies the system of inequalities, for each i we have $\langle \mathbf{a}_i, \mathbf{w} + \sum c_j \mathbf{e}_j \rangle \leq b_i$, which simplifies to $\langle \mathbf{a}_i, \sum c_j \mathbf{e}_j \rangle \leq b_i$ since $\langle \mathbf{a}_i, \mathbf{w} \rangle = 0$. Consequently, $\sum_j c_j \langle \mathbf{a}_i, \mathbf{e}_j \rangle \leq b_i$ for each i, which implies $\mathbf{Mc} \leq \mathbf{b}$ where $\mathbf{M} = [m_{i,j}] \in \mathbb{R}^{m \times p}$ is given by $m_{i,j} = \langle \mathbf{a}_i, \mathbf{e}_j \rangle$.

Similarly, since $P(\mathbf{b}')$ is nonempty by assumption, there exist $\widetilde{\mathbf{v}} \in P(\mathbf{b}')$ which we can decompose as $\widetilde{\mathbf{v}} = \widetilde{\mathbf{w}} + \sum_j \widetilde{c}_j \mathbf{e}_j$. Again, we obtain $\mathbf{M}\widetilde{\mathbf{c}} \leq \mathbf{b}'$. Thus we can apply the finite dimensional results (for instance Lemma 2.4.2) to the mapping $P_F : \mathbf{b} \mapsto \{\mathbf{u} \in \mathbb{R}^p : \mathbf{Mu} \leq \mathbf{b}\}$. Since $\mathbf{c} \in P_F(\mathbf{b})$ and $\widetilde{\mathbf{c}} \in P_F(\mathbf{b}')$, and so they are nonempty, there exists a $L_M > 0$ and $\mathbf{c}' \in P_F(\mathbf{b}')$ such that

$$\left\|\mathbf{c} - \mathbf{c}'\right\|_1 \leq L_M \left\|\mathbf{b} - \mathbf{b}'\right\|_1.$$

Thus, let $\mathbf{v}' = \mathbf{w} + \sum c_j' \mathbf{e}_j$. Note that $\mathbf{v}' \in P(\mathbf{b}')$, since $\left\langle \mathbf{a}_i, \mathbf{w} + \sum_j c_j' \mathbf{e}_j \right\rangle = \sum_j c_j' \langle \mathbf{a}_i, \mathbf{e}_j \rangle \leq b_i'$ for each i since $\mathbf{c}'$ satisfies $\mathbf{Mc}' \leq \mathbf{b}'$. Finally,

$$\begin{aligned}
\left\|\mathbf{v} - \mathbf{v}'\right\|_V &= \left\|\sum c_j \mathbf{e}_j - \sum c_j' \mathbf{e}_j\right\|_V \\
&\leq \sum \left|c_j - c_j'\right| \left\|\mathbf{e}_j\right\|_V \\
&\leq \max\{\left\|\mathbf{e}_j\right\|_V\} \left\|\mathbf{c} - \mathbf{c}'\right\|_1 \\
&\leq \max\{\left\|\mathbf{e}_j\right\|_V\} L_M \left\|\mathbf{b} - \mathbf{b}'\right\|_1.
\end{aligned}$$

Since the choice of $\mathbf{v}$ was arbitrary, the result follows since the Lipschitz constant is independent of $\mathbf{b}$ and $\mathbf{b}'$ which were chosen arbitrarily. □

The next result establishes Lipschitz continuity of the solution sets of certain optimization problems over a polyhedron.

Lemma A.4.3. *Let V be a real Banach space. For $m \in \mathbb{N}$, let $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\} \subset V^*$, and for $\mathbf{b} \in \mathbb{R}^m$, $P(\mathbf{b}) = \{\mathbf{v} \in V : \langle \mathbf{a}_i, \mathbf{v} \rangle \leq b_i, \forall i\}$. For $\mathbf{b} \in F = \{\mathbf{d} : P(\mathbf{d}) \neq \varnothing\}$, let*

$$\begin{aligned}
q_i(\mathbf{b}) &= \sup\{\langle \mathbf{a}_i, \mathbf{v} \rangle : \mathbf{v} \in P(\mathbf{b})\}, \\
S_i(\mathbf{b}) &= \{\mathbf{v} \in P(\mathbf{b}) : \langle \mathbf{a}_i, \mathbf{v} \rangle = q_i(\mathbf{b})\}.
\end{aligned}$$

Then for each i, there exists $L_i > 0$ such that for all $(\mathbf{b}, \mathbf{b}') \in F \times F$ and for any $\mathbf{v} \in S_i(\mathbf{b})$, *there exists a* $\mathbf{v}' \in S_i(\mathbf{b}')$ *with*

$$\left\|\mathbf{v} - \mathbf{v}'\right\|_V \leq L_i \left\|\mathbf{b} - \mathbf{b}'\right\|_1 .$$

Proof. We use the decomposition result from Lemma A.4.1; we can assume that there exist $\{\mathbf{e}_1, \dots, \mathbf{e}_p\} \subset V$ such that for any $\mathbf{v} \in V$, there exists a $\mathbf{w} \in V$ and $\mathbf{c} \in \mathbb{R}^p$ with $\langle \mathbf{a}_j, \mathbf{w} \rangle = 0$ for each j and $\mathbf{v} = \mathbf{w} + \sum_k c_k \mathbf{e}_k$.

Choose $i \in \{1, \dots, m\}$. Define $\mathbf{m}_i \in \mathbb{R}^p$ by $m_{i,k} = \langle \mathbf{a}_i, \mathbf{e}_k \rangle$, and $\mathbf{M} = [m_{j,k}] \in \mathbb{R}^{m \times p}$ by $m_{j,k} = \langle \mathbf{a}_j, \mathbf{e}_k \rangle$ (and so $\mathbf{m}_i^{\mathrm{T}}$ is the i^{th} row of $\mathbf{M}$). Consider the (finite) linear program parameterized by the right-hand side of its constraints

$$p_i : \mathbf{b} \mapsto \sup \left\{ \mathbf{m}_i^{\mathrm{T}} \mathbf{c} : \mathbf{c} \in \mathbb{R}^p, \mathbf{M}\mathbf{c} \leq \mathbf{b} \right\} . \tag{A.4}$$

For $\mathbf{b} \in F$, there exists $\mathbf{v} \in P(\mathbf{b})$, and by the decomposition result there exists $\mathbf{c} \in \mathbb{R}^p$ feasible in LP (A.4). Further, it is clear that $p_i(\mathbf{b})$ is finite (the LP is bounded and feasible); thus $p_i(\mathbf{b}) = \mathbf{m}_i^{\mathrm{T}} \mathbf{c}^*$ for some $\mathbf{c}^*$ such that $\mathbf{M}\mathbf{c}^* \leq \mathbf{b}$. Let $\mathbf{v}^* = \sum_k c_k^* \mathbf{e}_k$. Then $\mathbf{v}^* \in P(\mathbf{b})$ and $\langle \mathbf{a}_i, \mathbf{v}^* \rangle = p_i(\mathbf{b})$. By definition of the supremum, $p_i(\mathbf{b}) \leq q_i(\mathbf{b})$, and if this holds with strict inequality, then there exists a $\widehat{\mathbf{v}} \in P(\mathbf{b})$ with $\langle \mathbf{a}_i, \widehat{\mathbf{v}} \rangle > p_i(\mathbf{b})$. However, this would imply that there exists a $\widehat{\mathbf{c}} \in \mathbb{R}^p$ with $\mathbf{M}\widehat{\mathbf{c}} \leq \mathbf{b}$ and $\mathbf{m}_i^{\mathrm{T}} \widehat{\mathbf{c}} > p_i(\mathbf{b})$, which is a contradiction. Thus, $p_i(\mathbf{b}) = q_i(\mathbf{b})$. Further, $\mathbf{v}^* \in S_i(\mathbf{b})$. By Lemma 2.4.2, p_i is Lipschitz continuous on F, and so q_i is Lipschitz continuous on F.

Now, choose $\mathbf{b}$ and $\mathbf{b}'$ in F. By the above, $S_i(\mathbf{b})$ and $S_i(\mathbf{b}')$ are nonempty. Writing $S_i(\mathbf{b})$ as

$$S_i(\mathbf{b}) = \{\mathbf{v} \in V : \langle -\mathbf{a}_i, \mathbf{v} \rangle \leq -q_i(\mathbf{b}), \langle \mathbf{a}_j, \mathbf{v} \rangle \leq b_j, \forall j\}$$

we apply Lemma A.4.2, to get that for any $\mathbf{v} \in S_i(\mathbf{b})$, there exists $\mathbf{v}' \in S_i(\mathbf{b}')$ such that

$$\left\|\mathbf{v} - \mathbf{v}'\right\|_V \leq L_{S,i} \left\|(\mathbf{b}, q_i(\mathbf{b})) - (\mathbf{b}', q_i(\mathbf{b}'))\right\|_1 = L_{S,i} \left(\left\|\mathbf{b} - \mathbf{b}'\right\|_1 + \left|q_i(\mathbf{b}) - q_i(\mathbf{b}')\right| \right)$$

for some $L_{S,i} > 0$. Combined with the Lipschitz continuity of q_i, this gives

$$\left\|\mathbf{v} - \mathbf{v}'\right\|_V \leq L_{S,i}(1 + L_{q,i}) \left\|\mathbf{b} - \mathbf{b}'\right\|_1$$

and the result follows. □

It is now possible to state a specific instance of the mappings which satisfy Assumption A.3.2.

Proposition A.4.4. *Let Assumption A.4.1 hold. For $m \in \mathbb{N}$, let $\{\mathbf{a}_1, \dots, \mathbf{a}_m\} \subset W_x^*$ and define $\mathbf{A} : W_x \to \mathbb{R}^m$ by $\mathbf{A}\mathbf{z} = (\langle \mathbf{a}_1, \mathbf{z} \rangle, \dots, \langle \mathbf{a}_m, \mathbf{z} \rangle)$. Let*

$$P_M : (t, \mathbf{d}) \mapsto \{\mathbf{z} \in V_x : \mathbf{A}\mathbf{z} \leq \mathbf{d}, \mathbf{G}\mathbf{z} \leq \mathbf{b}_G(t)\}.$$

Then $\mathbf{A}$,

$$D_M = \{(t, \mathbf{d}) \in T \times \mathbb{R}^m : P_M(t, \mathbf{d}) \neq \varnothing\}, \text{ and} \tag{A.5}$$

$$M_i : (t, \mathbf{d}) \mapsto \arg\max\{\langle \mathbf{a}_i, \mathbf{z} \rangle : \mathbf{A}\mathbf{z} \leq \mathbf{d}, \mathbf{G}\mathbf{z} \leq \mathbf{b}_G(t), \mathbf{z} \in V_x\} \tag{A.6}$$

satisfy Assumption A.3.2.

Proof. To see that Condition 1 of Assumption A.3.2 holds, choose any $i \in \{1, \dots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\mathbf{A}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\langle \mathbf{a}_i, \mathbf{x}(t, \mathbf{u}) \rangle = d_i$. Since $\mathbf{G}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{b}_G(t)$, it holds that $\mathbf{x}(t, \mathbf{u}) \in P_M(t, \mathbf{d})$, and thus $(t, \mathbf{d}) \in D_M$. Further, since $\langle \mathbf{a}_i, \mathbf{x}(t, \mathbf{u}) \rangle = d_i$, and any $\mathbf{z}$ such that $\langle \mathbf{a}_i, \mathbf{z} \rangle > d_i$ would be infeasible in LP (A.6), we must have $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$.

Next, note that if $P_M(t, \mathbf{d})$ is nonempty, then $M_i(t, \mathbf{d})$ is nonempty for all i ($M_i(t, \mathbf{d})$ is the solution set of a linear optimization problem that must be feasible and bounded; if necessary, we can repeat the argument in Lemma A.4.3). Then to see that Condition 2 of Assumption A.3.2 holds, choose any $(s, \mathbf{d}_1)$ and $(s, \mathbf{d}_2)$ in D_M. By definition of D_M and the previous observation, $M_i(s, \mathbf{d}_j)$ is nonempty for $i \in \{1, \dots, m\}$ and $j \in \{1, 2\}$. Applying Lemma A.4.3, we have that there exists a $L > 0$ and for each $\mathbf{z}_1 \in M_i(s, \mathbf{d}_1)$, there exists a $\mathbf{z}_2 \in M_i(s, \mathbf{d}_2)$ such that

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_{V_x} \leq L \left\|(\mathbf{d}_1, \mathbf{b}_G(s)) - (\mathbf{d}_2, \mathbf{b}_G(s))\right\|_1 = L \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_1.$$

□

A.4.2 A simplification

This section discusses a specific choice of the set $\{\mathbf{a}_i, \dots, \mathbf{a}_m\} \subset W_x^*$ which simplifies the construction of a bounding system by Theorem A.3.1. In this case, we must assume that D_x is a dense linear subspace of V_x and the dynamics $\mathbf{f}$ have the form

$$\mathbf{f}(t, \mathbf{p}, \mathbf{z}) = \mathbf{F}\mathbf{z} + \mathbf{h}(t, \mathbf{p}),$$

for some linear operator (not necessarily continuous/bounded) $\mathbf{F} : D_x \to W_x$ and $\mathbf{h} : T \times D_u \to W_x$. However, this may also provide some insight into an intelligent choice of the $\mathbf{a}_i$ in the general case. First, the concept of the adjoint of a linear operator should be introduced. The following comes from §2.6 of [35]. Since D_x is a dense linear subspace of V_x, the adjoint of $\mathbf{F}$ is $\mathbf{F}^* : D(\mathbf{F}^*) \to V_x^*$, for some linear subspace $D(\mathbf{F}^*)$ of W_x^*. The fundamental relation between $\mathbf{F}$ and $\mathbf{F}^*$ is

$$\langle \mathbf{y}, \mathbf{F}\mathbf{z} \rangle = \langle \mathbf{F}^*\mathbf{y}, \mathbf{z} \rangle, \quad \forall (\mathbf{z}, \mathbf{y}) \in D_x \times D(\mathbf{F}^*).$$

Next, an alternative definition of the M_i mappings is proposed. As is, it does not take advantage of any *a priori* information, however it could be modified to do so.

Proposition A.4.5. *Assume that D_x is a closed linear subspace of V_x. For $m \in \mathbb{N}$, let $\{\mathbf{a}_1, \dots, \mathbf{a}_m\} \subset W_x^*$ such that $\mathbf{a}_i \neq \mathbf{0}$ for all i, and define $\mathbf{A} : W_x \to \mathbb{R}^m$ by $\mathbf{A}\mathbf{z} = (\langle \mathbf{a}_1, \mathbf{z} \rangle, \dots, \langle \mathbf{a}_m, \mathbf{z} \rangle)$. Then $\mathbf{A}$,*

$$M_i : (t, \mathbf{d}) \mapsto \{\mathbf{z} \in D_x : \langle \mathbf{a}_i, \mathbf{z} \rangle = d_i\}, \text{ and} \tag{A.7}$$

$$D_M = T \times \mathbb{R}^m \tag{A.8}$$

satisfy Assumption A.3.2.

Proof. To see that Condition 1 of Assumption A.3.2 holds, choose any $i \in \{1, \dots, m\}$, $\mathbf{d} \in \mathbb{R}^m$, and $(t, \mathbf{u}) \in T \times \mathcal{U}$ such that $\mathbf{A}\mathbf{x}(t, \mathbf{u}) \leq \mathbf{d}$ and $\langle \mathbf{a}_i, \mathbf{x}(t, \mathbf{u}) \rangle = d_i$. Immediately we have $\mathbf{x}(t, \mathbf{u}) \in M_i(t, \mathbf{d})$, and trivially $(t, \mathbf{d}) \in D_M$.

Since D_x is a closed linear subspace of V_x, it is thus a Banach space (with the norm $\|\cdot\|_{V_x}$). If $\mathbf{a}_i$ is continuous linear functional on V_x, then it is a continuous linear functional on the subspace D_x, and so $\mathbf{a}_i \in D_x^*$. Consequently, we can decompose D_x as in Lemma A.4.1;

there exists $p \in \mathbb{N}$ and $\{\mathbf{e}_1, \dots, \mathbf{e}_p\} \subset D_x$ such that for any $\mathbf{z} \in D_x$, there exists $\mathbf{w} \in D_x$ and $\mathbf{c} \in \mathbb{R}^p$ such that $\mathbf{z} = \mathbf{w} + \sum_j c_j \mathbf{e}_j$, $\langle \mathbf{a}_i, \mathbf{w} \rangle = 0$ for all i, and for all i there is some k_i such that $\langle \mathbf{a}_i, \mathbf{e}_{k_i} \rangle \neq 0$.

To see that Condition 2 of Assumption A.3.2 holds, choose any $i \in \{1, \dots, m\}$ and $(s, \mathbf{d}_1)$, $(s, \mathbf{d}_2) \in D_M$. For any $\mathbf{z}_1 \in M_i(s, \mathbf{d}_1)$, we have $\langle \mathbf{a}_i, \mathbf{z}_1 \rangle = d_{1,i}$. As above, let $\mathbf{z}_1 = \mathbf{w} + \sum_j c_j \mathbf{e}_j$. Let $\alpha = d_{2,i} - d_{1,i}$ and $\mathbf{z}_2 = \mathbf{z}_1 + (\alpha / \langle \mathbf{a}_i, \mathbf{e}_{k_i} \rangle) \mathbf{e}_{k_i}$. Then we have

$$\begin{aligned} \langle \mathbf{a}_i, \mathbf{z}_2 \rangle &= \langle \mathbf{a}_i, \mathbf{z}_1 \rangle + (\alpha / \langle \mathbf{a}_i, \mathbf{e}_{k_i} \rangle) \langle \mathbf{a}_i, \mathbf{e}_{k_i} \rangle \\ &= d_{1,i} + \alpha = d_{2,i}. \end{aligned}$$

Thus $\mathbf{z}_2 \in M_i(s, \mathbf{d}_2)$, and

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_{V_x} \leq \frac{\|\mathbf{e}_{k_i}\|}{|\langle \mathbf{a}_i, \mathbf{e}_{k_i} \rangle|} \|\mathbf{d}_1 - \mathbf{d}_2\|_1 .$$

We can choose a Lipschitz constant large enough so that the desired condition holds. □

The definition of M_i in Eqn. (A.7) is in general much "larger," and so roughly speaking "worse" than the one defined in Eqn. (A.6). However, consider the case when $\mathbf{F}^*$ has nontrivial eigenvectors (i.e. a vector $\mathbf{v} \in D(\mathbf{F}^*)$, $\mathbf{v} \neq \mathbf{0}$, such that $\mathbf{F}^* \mathbf{v} = \lambda \mathbf{v}$ for some real λ). If we choose $\{\mathbf{a}_1, \dots, \mathbf{a}_m\} \subset D(\mathbf{F}^*)$ to be a set of eigenvectors of $\mathbf{F}^*$ with corresponding eigenvalues $\{\lambda_1, \dots, \lambda_m\}$, then Hypothesis 4 in Theorem A.3.1 becomes

$$\langle \mathbf{a}_i, \mathbf{F}\mathbf{z} + \mathbf{h}(t, \mathbf{p}) \rangle \leq \dot{b}_i(t), \quad \forall (\mathbf{p}, \mathbf{z}) \in U \times D_x : \langle \mathbf{a}_i, \mathbf{z} \rangle = b_i(t), \tag{A.9}$$

where M_i has been defined as in Eqn. (A.7). However,

$$\begin{aligned} \langle \mathbf{a}_i, \mathbf{F}\mathbf{z} + \mathbf{h}(t, \mathbf{p}) \rangle &= \langle \mathbf{a}_i, \mathbf{F}\mathbf{z} \rangle + \langle \mathbf{a}_i, \mathbf{h}(t, \mathbf{p}) \rangle , \\ &= \langle \mathbf{F}^* \mathbf{a}_i, \mathbf{z} \rangle + \langle \mathbf{a}_i, \mathbf{h}(t, \mathbf{p}) \rangle , \\ &= \lambda_i \langle \mathbf{a}_i, \mathbf{z} \rangle + \langle \mathbf{a}_i, \mathbf{h}(t, \mathbf{p}) \rangle , \end{aligned}$$

where the properties of the adjoint and its eigenvectors have been used. Plugging this back into (A.9), we have

$$\lambda_i b_i(t) + \langle \mathbf{a}_i, \mathbf{h}(t, \mathbf{p}) \rangle \leq \dot{b}_i(t), \quad \forall \mathbf{p} \in U, \tag{A.10}$$

which in effect has removed all uncertainty with respect to the states. Depending on the form of $\mathbf{h}$, it may be much easier to construct an initial value problem in ordinary differential equations whose solution $\mathbf{b}$ satisfies the inequality (A.10).

As a final note, it is assumed in this section that D_x is a dense subspace of V_x to ensure that the adjoint of $\mathbf{F}$ exists. Combined with the assumption in Proposition A.4.5 that D_x is closed, this implies that $D_x = V_x$.

Bibliography

[1] MOSEK. http://www.mosek.com/, 2015.

[2] SeDuMi: Optimization over symmetric cones. http://sedumi.ie.lehigh.edu/, 2015.

[3] YALMIP. http://users.isy.liu.se/johanl/yalmip/, 2015.

[4] V. Acary. Time-Stepping via Complementarity. In F. Vasca and L. Iannelli, editors, *Dynamics and Control of Switched Electronic Systems*, Advances in Industrial Control, chapter 14, pages 417–450. Springer London, London, 2012.

[5] V. S. K. Adi and C.-T. Chang. A mathematical programming formulation for temporal flexibility analysis. *Computers & Chemical Engineering*, 57:151–158, 2013.

[6] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs - I. Theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.

[7] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[8] M. Anitescu and A. Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications*, 47(2):207–235, Nov. 2010.

[9] E. Asarin, T. Dang, and A. Girard. Reachability Analysis of Nonlinear Systems Using Conservative Approximation. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, pages 20–35. Springer Berlin Heidelberg, 2003.

[10] J.-P. Aubin. *Viability Theory.* Birkhauser, Boston, 1991.

[11] J.-P. Aubin and H. Frankowska. *Set-Valued Analysis.* Birkhauser, Boston, 1990.

[12] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer. *Non-Linear Parametric Optimization.* Birkhauser, Boston, 1983.

[13] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations.* Birkhauser, Boston, 1997.

[14] P. I. Barton and C. K. Lee. Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Transactions on Modeling and Computer Simulation*, 12(4):256–289, Oct. 2002.

[15] A. Bemporad, C. Filippi, and F. D. Torrisi. Inner and outer approximations of polytopes using boxes. *Computational Geometry*, 27(2):151–178, Feb. 2004.

[16] A. Bemporad and M. Morari. Robust Model Predictive Control: A Survey. In A. Garulli and A. Tesi, editors, *Robustness in Identification and Control*, Lecture Notes in Control and Information Sciences, pages 207–226. Springer London, 1999.

[17] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications.* Springer, New York, second edition, 2003.

[18] A. Ben-Tal and A. Nemirovski. Robust Convex Optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[19] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, Aug. 1999.

[20] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications.* SIAM, Philadelphia, 2001.

[21] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massachusetts, second edition, 1999.

[22] D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, Massachusetts, third edition, 2005.

[23] D. P. Bertsekas. *Convex Optimization Theory.* Athena Scientific, Belmont, Massachusetts, 2009.

[24] D. P. Bertsekas and I. B. Rhodes. Recursive state estimation for a set-membership description of uncertainty. *IEEE Transactions on Automatic Control*, 16(2):117–128, Apr. 1971.

[25] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, Massachusetts, 1997.

[26] M. Berz and K. Makino. Verified integration of ODEs and flows using algebraic methods on high-order Taylor models. *Reliable Computing*, 4:361–369, 1998.

[27] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming.* SIAM, Philadelphia, 2010.

[28] B. Bhattacharjee, W. H. Green, and P. I. Barton. Interval Methods for Semi-Infinite Programs. *Computational Optimization and Applications*, 30(1):63–93, Jan. 2005.

[29] B. Bhattacharjee, P. Lemonidis, W. H. Green, and P. I. Barton. Global solution of semi-infinite programs. *Mathematical Programming, Series B*, 103:283–307, 2005.

[30] J. W. Blankenship and J. E. Falk. Infinitely Constrained Optimization Problems. *Journal of Optimization Theory and Applications*, 19(2):261–281, 1976.

[31] O. Bokanowski, N. Forcadel, and H. Zidani. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM Journal on Control and Optimization*, 48(7):4292–4316, 2010.

[32] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, 1994.

[33] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, 2004.

[34] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, Philadelphia, 1996.

[35] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations.* Springer, New York, 2011.

[36] B. Chachuat. MC++: A Versatile Library for McCormick Relaxations and Taylor Models. `http://www.imperial.ac.uk/people/b.chachuat/research.html`, 2015.

[37] B. Chachuat and M. Villanueva. Bounding the solutions of parametric ODEs: When Taylor models meet differential inequalities. In I. D. L. Bogle and M. Fairweather, editors, *Proceedings of the 22nd European Symposium on Computer Aided Process Engineering*, pages 17–20, 2012.

[38] F. L. Chernousko. Optimal Ellipsoidal Estimates of Uncertain Systems: An Overview and New Results. In K. Marti, Y. Ermoliev, and M. Makowski, editors, *Coping with Uncertainty*, volume 633 of *Lecture Notes in Economics and Mathematical Systems*, chapter 7, pages 141–161. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[39] A. Chutinan and B. H. Krogh. Computing polyhedral approximations to flow pipes for dynamic systems. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 2, pages 2089–2094 vol. 2, 1998.

[40] T. Dang and O. Maler. Reachability Analysis via Face Lifting. In T. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control*, pages 96–109. Springer Berlin Heidelberg, 1998.

[41] L. H. de Figueiredo and J. Stolfi. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 37:147–158, 2004.

[42] M. Diehl, B. Houska, O. Stein, and P. Steuermann. A lifting method for generalized semi-infinite programs based on lower level Wolfe duality. *Computational Optimization and Applications*, 54(1):189–210, June 2013.

[43] V. D. Dimitriadis and E. N. Pistikopoulos. Flexibility Analysis of Dynamic Systems. *Industrial & Engineering Chemistry Research*, 34(12):4451–4462, 1995.

[44] A. L. Dontchev and F. Lempio. Difference methods for differential inclusions: A survey. *SIAM Review*, 34(2):263–294, 1992.

[45] K. Du and R. B. Kearfott. The cluster problem in multivariate global optimization. *Journal of Global Optimization*, 5(3):253–265, 1994.

[46] N. C. Duarte, M. J. Herrgård, and B. Ø. Palsson. Reconstruction and validation of Saccharomyces cerevisiae iND750, a fully compartmentalized genome-scale metabolic model. *Genome Research*, 14(7):1298–1309, 2004.

[47] I. S. Duff and J. K. Reid. The design of MA48: a code for the direct solution of sparse unsymmetric linear systems of equations. *ACM Transactions on Mathematical Software*, 22(2):187–226, June 1996.

[48] D. Efimov, T. Raïssi, S. Chebotarev, and A. Zolghadri. Interval state observer for nonlinear time varying systems. *Automatica*, 49(1):200–205, 2013.

[49] L. C. Evans. *Partial Differential Equations.* American Mathematical Society, Providence, Rhode Island, second edition, 2010.

[50] W. F. Feehery, J. E. Tolsma, and P. I. Barton. Efficient sensitivity analysis of large-scale differential-algebraic systems. *Applied Numerical Mathematics*, 25(1):41–54, Oct. 1997.

[51] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides.* Kluwer Academic, Boston, 1988.

[52] C. A. Floudas, Z. H. Gümüş, and M. G. Ierapetritou. Global Optimization in Design under Uncertainty: Feasibility Test and Flexibility Index Problems. *Industrial & Engineering Chemistry Research*, 40:4267–4282, 2001.

[53] C. A. Floudas and O. Stein. The adaptive convexification algorithm: A feasible point method for semi-infinite programming. *SIAM Journal on Optimization*, 18(4):1187–1208, 2007.

[54] T. Gal. *Postoptimal Analyses, Parametric Programming, and Related Topics.* Walter de Gruyter, New York, second edition, 1995.

[55] S. Galán, W. F. Feehery, and P. I. Barton. Parametric sensitivity functions for hybrid discrete/continuous systems. *Applied Numerical Mathematics*, 31(1):17–47, Sept. 1999.

[56] GAMS Development Corporation. GAMS: General Algebraic Modeling System. `http://www.gams.com`, 2014.

[57] E. P. Gatzke, J. E. Tolsma, and P. I. Barton. Construction of Convex Relaxations Using Automated Code Generation Techniques. *Optimization and Engineering*, 3:305–326, 2002.

[58] A. M. Geoffrion. Duality in Nonlinear Programming: A Simplified Applications-Oriented Development. *SIAM Review*, 13(1):1–37, Jan. 1971.

[59] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review*, 47(1):99–131, 2005.

[60] J. A. Gomez, K. Höffner, and P. I. Barton. DFBAlab: a fast and reliable MATLAB code for dynamic flux balance analysis. *BMC Bioinformatics*, 15(409):1–10, 2014.

[61] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

[62] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[63] M. R. Greenstreet and I. Mitchell. Integrating Projections. In T. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control*, pages 159–174. Springer Berlin Heidelberg, 1998.

[64] M. R. Greenstreet and I. Mitchell. Reachability Analysis Using Polygonal Projections. In F. W. Vaandrager and J. H. van Schuppen, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, pages 103–116. Springer Berlin Heidelberg, 1999.

[65] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* SIAM, Philadelphia, second edition, 2008.

[66] I. E. Grossmann, B. A. Calfa, and P. Garcia-Herreros. Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. *Computers & Chemical Engineering*, 70:22–34, 2014.

[67] F. Guerra Vâzquez, J.-J. Rückmann, O. Stein, and G. Still. Generalized semi-infinite programming: A tutorial. *Journal of Computational and Applied Mathematics*, 217(2):394–419, Aug. 2008.

[68] K. P. Halemane and I. E. Grossmann. Optimal Process Design Under Uncertainty. *AIChE Journal*, 29(3):425–433, 1983.

[69] L. Han, A. Tiwari, M. K. Camlibel, and J.-S. Pang. Convergence of time-stepping schemes for passive and extended linear complementarity systems. *SIAM Journal on Numerical Analysis*, 47(5):3768–3796, 2009.

[70] T. J. Hanly and M. A. Henson. Dynamic flux balance modeling of microbial co-cultures for efficient batch fermentation of glucose and xylose mixtures. *Biotechnology and Bioengineering*, 108(2):376–85, Feb. 2011.

[71] T. J. Hanly and M. A. Henson. Dynamic metabolic modeling of a microaerobic yeast co-culture: predicting and optimizing ethanol production from glucose/xylose mixtures. *Biotechnology for Biofuels*, 6(1):44, Jan. 2013.

[72] G. W. Harrison. Dynamic models with uncertain parameters. In X. J. R. Avula, editor, *Proceedings of the First International Conference on Mathematical Modeling*, volume 1, pages 295–304. University of Missouri Rolla, 1977.

[73] G. W. Harrison. Compartmental models with uncertain flow rates. *Mathematical Biosciences*, 43:131–139, 1979.

[74] S. M. Harwood, J. K. Scott, and P. I. Barton. Bounds on reachable sets using ordinary differential equations with linear programs embedded. In S. Tarbouriech and M. Krstic, editors, *Nonlinear Control Systems*, volume 9, pages 62–67, 2013.

[75] S. M. Harwood, J. K. Scott, and P. I. Barton. Bounds on reachable sets using ordinary differential equations with linear programs embedded. *IMA Journal of Mathematical Control and Information*, (in press).

[76] E. M. Hendrix, C. J. Mecking, and T. H. Hendriks. Finding robust solutions for product design problems. *European Journal of Operational Research*, 92(1):28–36, July 1996.

[77] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

[78] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005.

[79] J. L. Hjersted and M. A. Henson. Steady-state and dynamic flux balance analysis of ethanol production by Saccharomyces cerevisiae. *IET Systems Biology*, 3(3):167–79, May 2009.

[80] J. L. Hjersted, M. A. Henson, and R. Mahadevan. Genome-Scale Analysis of Saccharomyces cerevisiae Metabolism and Ethanol Production in Fed-Batch Culture. *Biotechnology and Bioengineering*, 97(5):1190–1204, 2007.

[81] K. Höffner, S. M. Harwood, and P. I. Barton. A reliable simulator for dynamic flux balance analysis. *Biotechnology and Bioengineering*, 110(3):792–802, Mar. 2013.

[82] H. T. Huang, C. S. Adjiman, and N. Shah. Quantitative framework for reliable safety analysis. *AIChE Journal*, 48(1):78–96, 2002.

[83] J. K. Hunter. Notes on partial differential equations. `https://www.math.ucdavis.edu/~hunter/pdes/pde_notes.pdf`, 2014. Lecture Notes, Department of Mathematics, University of California at Davis.

[84] I. Hwang, D. M. Stipanovic, and C. J. Tomlin. Applications of polytopic approximations of reachable sets to linear dynamic games and a class of nonlinear systems. In *Proceedings of the American Control Conference*, volume 6, pages 4613–4619, 2003.

[85] IBM. IBM ILOG CPLEX: High performance mathematical programming solver. `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`, 2014.

[86] J. P. Ignizio. *Goal Programming and Extensions.* Lexington Books, Lexington, Massachusetts, 1976.

[87] H. Isermann. Linear lexicographic optimization. *OR Spektrum*, 4(4):223–228, Dec. 1982.

[88] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, 38(6):1079–1082, 2002.

[89] H. T. Jongen, J.-J. Rückmann, and O. Stein. Generalized semi-infinite optimization: A first order optimality condition and examples. *Mathematical Programming*, 83(1-3):145–158, Jan. 1998.

[90] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82:35–45, 1960.

[91] N. Kanzi and S. Nobakhtian. Necessary optimality conditions for nonsmooth generalized semi-infinite programming problems. *European Journal of Operational Research*, 205(2):253–261, Sept. 2010.

[92] U. Kaplan, M. Türkay, L. T. Biegler, and B. Karasözen. Modeling and simulation of metabolic networks for estimation of biomass accumulation parameters. *Discrete Applied Mathematics*, 157(10):2483–2493, May 2009.

[93] A. Kawamura. Lipschitz Continuous Ordinary Differential Equations are Polynomial-Space Complete. *Computational Complexity*, 19(2):305–332, May 2010.

[94] A. Kawamura, H. Ota, C. Rosnick, and M. Ziegler. Computational complexity of smooth differential equations. *Logical Methods in Computer Science*, 10:1–15, 2014.

[95] L. G. Khachiyan and M. J. Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61(1-3):137–159, 1993.

[96] K. A. Khan and P. I. Barton. Evaluating an element of the Clarke generalized Jacobian of a piecewise differentiable function. *ACM Transactions on Mathematical Software*, 39(4):1–28, 2013.

[97] M. Kieffer and E. Walter. Guaranteed nonlinear state estimator for cooperative systems. *Numerical Algorithms*, 37(1-4):187–198, Dec. 2004.

[98] M. Kieffer and E. Walter. Guaranteed estimation of the parameters of nonlinear continuous-time models : Contributions of interval analysis. *International Journal of Adaptive Control and Signal Processing*, 25:191–207, 2011.

[99] M. Kieffer, E. Walter, and I. Simeonov. Guaranteed nonlinear parameter estimation for continuous-time dynamical models. *Robust Control Design*, 5:685–690, 2006.

[100] D. Klatte and B. Kummer. Stability properties of infima and optimal solutions of parametric optimization problems. *Lecture Notes in Economics and Mathematical Systems*, 255:215–229, 1985.

[101] P. Korhonen and M. Halme. Using lexicographic parametric programming for searching a non-dominated set in multiple-objective linear programming. *Journal of Multi-Criteria Decision Analysis*, 5:291–300, 1996.

[102] A. B. Kurzhanski, I. M. Mitchell, and P. Varaiya. Optimization techniques for state-constrained control and obstacle problems. *Journal of Optimization Theory and Applications*, 128(3):499–521, 2006.

[103] A. B. Kurzhanski and P. Varaiya. Dynamic optimization for reachability problems. *Journal of Optimization Theory and Applications*, 108(2):227–251, 2001.

[104] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability under state constraints. *SIAM Journal on Control and Optimization*, 45(4):1369–1394, Jan. 2006.

[105] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem.* John Wiley & Sons, New York, 1991.

[106] E. S. Levitin and R. Tichatschke. A Branch-and-Bound Approach for Solving a Class of Generalized Semi-infinite Programming Problems. *Journal of Global Optimization*, 13:299–315, 1998.

[107] Y. Lin and M. A. Stadtherr. Validated solutions of initial value problems for parametric ODEs. *Applied Numerical Mathematics*, 57(10):1145–1162, Oct. 2007.

[108] Y. Lin and M. A. Stadtherr. Fault detection in nonlinear continuous-time systems with uncertain parameters. *AIChE Journal*, 54(9):2335–2345, 2008.

[109] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1-3):193–228, 1998.

[110] J. Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. *2004 IEEE International Conference on Computer Aided Control Systems Design*, pages 284–289, 2004.

[111] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, Massachusetts, second edition, 1984.

[112] R. Mahadevan and C. H. Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic Engineering*, 5(4):264–276, Oct. 2003.

[113] K. Makino. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators.* PhD thesis, Michigan State University, 1998.

[114] K. Makino and M. Berz. Remainder Differential Algebras and Their Applications. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation: Techniques, Applications, and Tools*, chapter 5, pages 63–75. SIAM, 1996.

[115] O. L. Mangasarian and T. H. Shiau. Lipschitz Continuity of Solutions of Linear Inequalities, Programs, and Complementarity Problems. *SIAM Journal on Control and Optimization*, 25(3):583–595, 1987.

[116] R. Mattheij and J. Molenaar. *Ordinary Differential Equations in Theory and Practice.* SIAM, Philadelphia, 2002.

[117] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[118] N. Meslem and N. Ramdani. Interval observer design based on nonlinear hybridization and practical stability analysis. *International Journal of Adaptive Control and Signal Processing*, 25:228–248, 2011.

[119] N. Meslem, N. Ramdani, and Y. Candau. Using hybrid automata for set-membership state estimation with uncertain nonlinear continuous-time systems. *Journal of Process Control*, 20:481–489, 2010.

[120] R. Misener and C. A. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 59(2-3):503–526, 2014.

[121] 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, July 2005.

[122] A. Mitsos. Global optimization of semi-infinite programs via restriction of the right-hand side. *Optimization*, 60(10-11):1291–1308, Oct. 2011.

[123] A. Mitsos and P. I. Barton. Parametric mixed-integer 0-1 linear programming: The general case for a single parameter. *European Journal of Operational Research*, 194:663–686, 2009.

[124] A. Mitsos, B. Chachuat, and P. I. Barton. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization*, 20(2):573–601, 2009.

[125] A. Mitsos, P. Lemonidis, and P. I. Barton. Global solution of bilevel programs with a nonconvex inner program. *Journal of Global Optimization*, 42(4):475–513, Dec. 2008.

[126] A. Mitsos, P. Lemonidis, C. K. Lee, and P. I. Barton. Relaxation-based bounds for semi-infinite programs. *SIAM Journal on Optimization*, 19(1):77–113, 2008.

[127] A. Mitsos and A. Tsoukalas. Global optimization of generalized semi-infinite programs via restriction of the right hand side. *Journal of Global Optimization*, 61(1):1–17, 2014.

[128] M. Moisan and O. Bernard. Interval observers for non-montone systems. Application to bioprocess models. *16th IFAC World Congress*, 16:43–48, 2005.

[129] M. Moisan and O. Bernard. An interval observer for non-monotone systems: application to an industrial anaerobic digestion process. *10th International IFAC Symposium on Computer Applications in Biotechnology*, 1:325–330, 2007.

[130] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis.* SIAM, Philadelphia, 2009.

[131] J. R. Munkres. *Topology: a first course.* Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[132] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41(1):35–52, Jan. 2009.

[133] N. S. Nedialkov. *Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation.* PhD thesis, University of Toronto, 1999.

[134] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM, Philadelphia, 1994.

[135] A. Neumaier. *Interval Methods for Systems of Equations.* Cambridge University Press, New York, 1990.

[136] A. Neumaier. Taylor forms - Use and limits. *Reliable Computing*, 9(1):43–79, 2003.

[137] A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. In A. Iserles, editor, *Acta Numerica*, volume 13, pages 271–369. Cambridge University Press, Cambridge, UK, 2004.

[138] V. H. Nguyen and J.-J. Strodiot. Computing a global optimal solution to a design centering problem. *Mathematical Programming*, 53(1-3):111–123, Jan. 1992.

[139] K. Nickel. How to Fight the Wrapping Effect. In K. Nickel, editor, *Interval Mathematics 1985*, pages 121–132. Springer Berlin Heidelberg, 1986.

[140] J. D. Orth, I. Thiele, and B. Ø. Palsson. What is flux balance analysis? *Nature Biotechnology*, 28(3):245–248, 2010.

[141] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer-Verlag, New York, 2003.

[142] A. I. Ovseevich. On equations of ellipsoids approximating attainable sets. *Journal of Optimization Theory and Applications*, 95(3):659–676, 1997.

[143] B. Palsson. *Systems Biology: Properties of Reconstructed Networks.* Cambridge University Press, New York, NY, 2006.

[144] J.-S. Pang and D. E. Stewart. Differential variational inequalities. *Mathematical Programming, Series A*, 113:345–424, Jan. 2008.

[145] T. Park and P. I. Barton. State event location in differential-algebraic models. *ACM Transactions on Modeling and Computer Simulation*, 6(2):137–165, Apr. 1996.

[146] E. Polak. On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Review*, 29(1):21–89, 1987.

[147] I. Polik and T. Terlaky. A Survey of the S-Lemma. *SIAM Review*, 49(3):371–418, 2007.

[148] L. Pourkarimi and M. Zarepisheh. A dual-based algorithm for solving lexicographic multiple objective programs. *European Journal of Operational Research*, 176(3):1348–1356, Feb. 2007.

[149] A. U. Raghunathan, J. R. Perez-Correa, E. Agosin, and L. T. Biegler. Parameter estimation in metabolic flux balance models for batch fermentation - Formulation & Solution using Differential Variational Inequalities. *Annals of Operations Research*, 148(1):251–270, Oct. 2006.

[150] T. Raïssi, N. Ramdani, and Y. Candau. Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40(10):1771–1777, 2004.

[151] D. Ralph and S. Dempe. Directional derivatives of the solution of a parametric nonlinear program. *Mathematical Programming*, 70:159–172, 1995.

[152] N. Ramdani, N. Meslem, and Y. Candau. A hybrid bounding method for computing an over-approximation for the reachable set of uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 54(10):2352–2364, Oct. 2009.

[153] J. Reed, T. Vo, C. Schilling, and B. Ø. Palsson. An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR). *Genome Biology*, 4(9):R54, 2003.

[154] S. M. Robinson and R. H. Day. A sufficient condition for continuity of optimal sets in mathematical programming. *Journal of Mathematical Analysis and Applications*, 45:506–511, 1974.

[155] C. M. Rocco, J. A. Moreno, and N. Carrasquero. Robust design using a hybrid-cellular-evolutionary and interval-arithmetic approach: a reliability application. *Reliability Engineering & System Safety*, 79(2):149–159, Feb. 2003.

[156] J. O. Royset, A. Der Kiureghian, and E. Polak. Reliability-based optimal structural design by the decoupling approach. *Reliability Engineering and System Safety*, 73(3):213–221, 2001.

[157] J.-J. Rückmann and A. Shapiro. First-Order Optimality Conditions in Generalized Semi-Infinite Programming. *Journal of Optimization Theory and Applications*, 101(3):677–691, 1999.

[158] W. Rudin. *Principles of Mathematical Analysis.* McGraw-Hill, New York, third edition, 1976.

[159] N. V. Sahinidis. BARON 14.0.3: Global Optimization of Mixed-Integer Nonlinear Programs, User's Manual. `http://www.minlp.com/downloads/docs/baronmanual.pdf`, 2014.

[160] A. M. Sahlodin. *Global Optimization of Dynamic Process Systems using Complete Search Methods.* PhD thesis, McMaster Univeristy, 2013.

[161] P. Saint-Pierre. Approximation of Slow Solutions to Differential Inclusions. *Applied Mathematics & Optimization*, 22:311–330, 1990.

[162] P. Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics & Optimization*, 29(2):187–209, 1994.

[163] S. D. Schaber. *Tools for dynamic model development.* PhD thesis, Massachusetts Institute of Technology, 2014.

[164] J. Schellenberger, J. Park, T. Conrad, and B. Ø. Palsson. BiGG: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinformatics*, 11(1):213, 2010.

[165] J. M. Schumacher. Complementarity systems in optimization. *Mathematical Programming, Series B*, 101:263–295, 2004.

[166] J. K. Scott. *Reachability Analysis and Deterministic Global Optimization of Differential-Algebraic Systems.* PhD thesis, Massachusetts Institute of Technology, 2012.

[167] J. K. Scott and P. I. Barton. Tight, efficient bounds on the solutions of chemical kinetics models. *Computers & Chemical Engineering*, 34(5):717–731, May 2010.

[168] J. K. Scott and P. I. Barton. Bounds on the reachable sets of nonlinear control systems. *Automatica*, 49(1):93–100, 2013.

[169] J. K. Scott and P. I. Barton. Improved relaxations for the parametric solutions of ODEs using differential inequalities. *Journal of Global Optimization*, 57:143–176, 2013.

[170] J. K. Scott, B. Chachuat, and P. I. Barton. Nonlinear convex and concave relaxations for the solutions of parametric ODEs. *Optimal Control Applications and Methods*, 34(2):145–163, 2013.

[171] A. Seifi, K. Ponnambalam, and J. Vlach. A unified approach to statistical design centering of integrated circuits with correlated parameters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 46(1):190–196, 1999.

[172] L. F. Shampine and M. W. Reichelt. The MATLAB ODE Suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, Jan. 1997.

[173] S. Shan and G. G. Wang. Reliable design space and complete single-loop reliability-based design optimization. *Reliability Engineering and System Safety*, 93(8):1218–1230, 2008.

[174] W. Shi and C. Zhang. Error analysis of generalized polynomial chaos for nonlinear random ordinary differential equations. *Applied Numerical Mathematics*, 62(12):1954–1964, 2012.

[175] A. B. Singer. *Global Dynamic Optimization.* PhD thesis, Massachusetts Institute of Technology, 2004.

[176] A. B. Singer and P. I. Barton. Bounding the Solutions of Parameter Dependent Nonlinear Ordinary Differential Equations. *SIAM Journal on Scientific Computing*, 27(6):2167–2182, Jan. 2006.

[177] A. B. Singer and P. I. Barton. Global Optimization with Nonlinear Ordinary Differential Equations. *Journal of Global Optimization*, 34(2):159–190, Feb. 2006.

[178] A. B. Singer, J. W. Taylor, P. I. Barton, and W. H. Green. Global dynamic optimization for parameter estimation in chemical kinetics. *Journal of Physical Chemistry A*, 110(3):971–976, 2006.

[179] B. Srinivasan, M. Amrhein, and D. Bonvin. Reaction and Flow Variants/Invariants in Chemical Reaction Systems with Inlet and Outlet Streams. *AIChE Journal*, 44(8):1858–1867, 1998.

[180] O. Stauning. *Automatic validation of numerical solutions.* PhD thesis, Technical University of Denmark, 1997.

[181] O. Stein. A semi-infinite approach to design centering. In S. Dempe and V. Kalashnikov, editors, *Optimization with Mulitvalued Mappings*, chapter 1, pages 209–228. Springer, 2006.

[182] O. Stein. How to solve a semi-infinite optimization problem. *European Journal of Operational Research*, 223(2):312–320, June 2012.

[183] O. Stein and G. Still. On generalized semi-infinite optimization and bilevel optimization. *European Journal of Operational Research*, 142(3):444–462, Nov. 2002.

[184] 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.

[185] O. Stein and A. Winterfeld. Feasible Method for Generalized Semi-Infinite Programming. *Journal of Optimization Theory and Applications*, 146(2):419–443, Mar. 2010.

[186] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application.* John Wiley & Sons, New York, 1986.

[187] G. Still. Generalized semi-infinite programming: Theory and methods. *European Journal of Operational Research*, 119:301–313, 1999.

[188] G. Still. Generalized semi-infinite programming: Numerical aspects. *Optimization*, 49(3):223–242, 2001.

[189] G. Still. Solving generalized semi-infinite programs by reduction to simpler problems. *Optimization*, 53(1):19–38, Feb. 2004.

[190] J. Stolfi and L. H. de Figueiredo. Self-validated numerical methods and applications, 1997.

[191] G. Strang. *Linear Algebra and its Applications.* Thomson Brooks/Cole, Belmont, California, fourth edition, 2006.

[192] M. D. Stuber and P. I. Barton. Robust simulation and design using semi-infinite programs with implicit functions. *International Journal of Reliability and Safety*, 5(3-4):378–397, 2011.

[193] M. D. Stuber and P. I. Barton. Semi-Infinite Optimization with Implicit Functions. *Industrial & Engineering Chemistry Research*, 54(5):307–317, 2015.

[194] J. F. Sturm. Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

[195] R. E. Swaney and I. E. Grossmann. An Index for Operational Flexibility in Chemical Process Design - Part I: Formulation and Theory. *AIChE Journal*, 31(4):621–630, 1985.

[196] R. E. Swaney and I. E. Grossmann. An Index for Operational Flexibility in Chemical Process Design - Part II: Computational Algorithms. *AIChE Journal*, 31(4):631–641, 1985.

[197] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

[198] K. L. Teo, C. J. Goh, and K. H. Wong. *A unified computational approach to optimal control problems.* Longman Scientific and Technical, New York, 1991.

[199] R. E. H. Thabet, T. Raïssi, C. Combastel, D. Efimov, and A. Zolghadri. An effective method to interval observer design for time-varying systems. *Automatica*, 50(10):2677–2684, 2014.

[200] J. E. Tolsma and P. I. Barton. DAEPACK: An Open Modeling Environment for Legacy Models. *Industrial & Engineering Chemistry Research*, 39(6):1826–1839, June 2000.

[201] A. Tsoukalas, B. Rustem, and E. N. Pistikopoulos. A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems. *Journal of Global Optimization*, 44(2):235–250, July 2009.

[202] M. E. Villanueva, B. Houska, and B. Chachuat. Unified framework for the propagation of continuous-time enclosures for parametric nonlinear ODEs. *Journal of Global Optimization*, (in press).

[203] W. Walter. *Differential and Integral Inequalities.* Springer, New York, 1970.

[204] X. Wang and T.-S. Chang. An Improved Univariate Global Optimization Algorithm with Improved Linear Lower Bounding Functions. *Journal of Global Optimization*, 8:393–411, 1996.

[205] Z. Wang and X. Wu. Piecewise Integration of Differential Variational Inequality. In T. Simos, G. Psihoyios, and C. Tsitouras, editors, *Numerical Analysis and Applied Mathematics, Vols. 1 and 2*, volume 2, pages 912–915. AIP, 2009.

[206] A. Wechsung. *Global optimization in reduced space*. PhD thesis, Massachusetts Institute of Technology, 2013.

[207] A. Wechsung, S. D. Schaber, and P. I. Barton. The cluster problem revisited. *Journal of Global Optimization*, 58(3):429–438, 2014.

[208] R. J.-B. Wets. On the continuity of the value of a linear program and of related polyhedral-valued multifunctions. *Mathematical Programming Study*, 24:14–29, 1985.

[209] D. Willett and J. S. W. Wong. On the Discrete Analogues of Some Generalizations of Gronwall's Inequality. *Monatshefte für Mathematik*, 69(4):362–367, 1965.

[210] A. Winterfeld. Application of general semi-infinite programming to lapidary cutting problems. *European Journal of Operational Research*, 191(3):838–854, Dec. 2008.

[211] D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[212] L. X. Yu. Pharmaceutical quality by design: Product and process development, understanding, and control. *Pharmaceutical Research*, 25(4):781–791, 2008.

[213] Z. Yuan, B. Chen, and J. Zhao. An overview on controllability analysis of chemical processes. *AIChE Journal*, 57(5):1185–1201, 2011.

[214] Y. Zhao and M. A. Stadtherr. Rigorous Global Optimization for Dynamic Systems Subject to Inequality Path Constraints. *Industrial & Engineering Chemistry Research*, 50(22):12678–12693, Nov. 2011.

[215] X. Zhuang, R. Pan, and X. Du. Enhancing product robustness in reliability-based design optimization. *Reliability Engineering & System Safety*, 138:145–153, 2015.