Parametric Programming in Control Theory

Thesis by

Jørgen Spjøtvold

Submitted in partial fulfillment of the requirements for the degree of philosophiae doctor, control engineering

Department of Engineering Cybernetics
Norwegian University of Science and Technology
N-7491 Trondheim, Norway
May 2008

Report 2008-5-W

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics
N-7491 Trondheim
Norway

Ph. D. Thesis: 2008:94
ITK Report: 2008-5-W

ISBN: 978-82-471-7880-5 (Electronic)
ISBN: 978-82-471-7877-5 (Printed)
ISSN: 1503-8181

Summary

The main contributions in this thesis are advances in parametric programming. The thesis is divided into three parts; theoretical advances, application areas and constrained control allocation. The first part deals with continuity properties and the structure of solutions to convex parametric quadratic and linear programs. The second part focuses on applications of parametric quadratic and linear programming in control theory. The third part deals with control allocation. This thesis is mainly a collection of articles where each article has been slightly modified. Note that several definitions differ from paper to paper (chapter to chapter).

Chapter 2 presents a novel method for obtaining a unique polyhedral representation of a continuous selection for convex parametric quadratic programs (which includes parametric linear programs). The main contribution is to utilize a two-level optimization method to obtain the *minimum norm solution*.

Chapter 3 introduces the so-called *facet-to-facet property* for parametric programs. It is pointed out that the correctness of several existing algorithms depend upon this property being satisfied. It is relatively simple to set up examples for convex parametric quadratic programs where the facet-to-facet property fails, however, for *strictly convex* quadratic programs it is rarely seen. It is exemplified that the facet-to-facet property does not hold for strictly convex parametric quadratic programs. A new exploration strategy is proposed to remedy this problem.

In Chapter 4 a method for obtaining explicit solutions to $\inf - \sup$ control of constrained discrete–time discontinuous piecewise affine system subject to state- and input-dependent disturbances is presented. For this problem a solution is not guaranteed to exist. When a solution does not exist, a sub-optimal solution is obtained and a bound on the sub-optimality is given. The method allows for the degree of sub-optimality to be specified *a priori*.

Chapter 5 presents a method that utilizes the dynamics of a discrete-time system to reduce the computational effort needed to evaluate the piecewise affine control law. The explicit control law and structure of the dynamic system makes it possible to map a polyhedral set one step forward in time. We demonstrate how this *one-step forward reach set* can be utilized to speed up the evaluation of the control law at the next sample instant.

Chapter 6 presents a case study and experimental results for constrained control allocation for a scale model of a *thruster-controlled floating platform*. An explicit solution to a convexified problem is computed and experimental results document

the performance.

In Chapter 7 a decomposition strategy for constrained linear control allocation problems is presented. The problem is divided into a master and a set of sub-problems for the purpose of obtaining a feasible, but possibly sub-optimal, solution. The decomposition strategy provides flexibility for the designer, for instance a mix of online optimization and explicit solutions can be employed. We also illustrate how the decomposition strategy can be utilized on the allocation problem for the thruster-controlled floating platform from Chapter 6.

Appendices A-D provide relevant background knowledge and minor results that may be useful for implementation of procedures and reproduction of results in the main part of the thesis.

Please note that this thesis is a paper collection and that emphasis have been put on keeping each chapter self-contained. Some restating of results are therefore necessary.

Preface

This thesis is the result of my Ph. D. studies conducted in the period June 2003 to May 2008. The project was funded partially by the Norwegian University of Science and Technology and partially by the Norwegian Research Council through the strategic university program on Computational Methods in Non-linear Motion Control headed by Professor Tor A. Johansen. Most of the work has been carried out at the Department of Engineering Cybernetics, Norwegian University of Science and Technology, under the supervision of Professor Tor A. Johansen and co-supervisor Dr. Petter Tøndel. In the period January 2006 to July 2006 I visited the control group at the Department of Electrical Engineering, Imperial College, London, where my work was supervised by Dr. Eric C. Kerrigan. The experiments with CyberRig I was carried out at the Marine Cybernetics Laboratory at Tyholt, Trondheim.

First and foremost I would like to thank Professor Tor A. Johansen. His help has been invaluable both on theoretical and practical topics. I thank my co-supervisor and office mate Dr. Petter Tøndel for fruitful discussions during the first two years. Dr. Eric C. Kerrigan deserves my gratitude for his guidance and ideas during my visit to Imperial. I would also like to thank the rest of my collaborators; Dr. S. V. Raković for his help and enthusiasm, Dr. C. Jones for valuable discussions and exchange of ideas, and finally Professor D. Q. Mayne for his comments. I thank Stefano Bertelli, Eivind Ruth and Torgeir Wahl for their help with CyberRig I.

The environment at NTNU has been fabulous and I wish to thank my friends and colleagues for making the years in Trondheim a memorable time. Special thanks to Dr. Bjørnar Bøhagen, Jostein Bakkenheim, Johannes Tjønnås, Dr. Petter Tøndel, Trond Skogstad, Pål Johansen, Dr. Stian Johansen, Luca Pivano, Dr. Erik Kyrkjebø, Francesco Scibilia, Giancarlo Marafioti, Markus Nygård and Christian Skaar. I wish to express my gratitude to my friends from Bærum, Henrik Jessen, Joachim Flak, Christian Pettersen, Joakim Kolstad, Andrè Fiskum and Eivind Løken Pettersen, for regularly coming to visit.

In London I was welcomed with open arms and I am grateful to Dr. S. V. Raković, Dr. Milan Prodanovic, Prisca Randimbivololona, Dr. Maria Tomas-Rodriguez and Dr. Vincent Andrieu for including me in their group.

Last, but not least, I want to thank my whole family for their never ending support and encouragement. I dedicate this thesis to my mother Elin and father Per for their efforts the last 30 years.

Jørgen Spjøtvold

May 30, 2008

Notation and Nomenclature

Mathematical terms used in this thesis are defined in the beginning of each chapter, however, for convenience we will define some terms that are frequently used. Unless otherwise is explicitly stated, these are the definitions used.

Definition 0.1 (Affine Hull) *The* affine hull *of a set S is the intersection of all affine sets containing S, and is denoted* $\mathrm{aff}(S)$.

Definition 0.2 (Dimension of a Set) *The* dimension of a set $S \subseteq \mathbb{R}^n$ *is the dimension of* $\mathrm{aff}(S)$*, and is denoted* $\dim(S)$*; if* $\dim(S) = n$*, then S is said to be full-dimensional.*

Definition 0.3 (Relative Interior) *The* relative interior *of a set S is the interior relative to* $\mathrm{aff}(S)$*, i.e.*

$$\mathrm{relint}(S) := \{x \in S \mid B(x,r) \cap \mathrm{aff}(S) \subseteq S \textit{ for some } r > 0\},$$

where the ball $B(x,r) := \{y \mid \|y - x\| \leq r\}$ and $\|\cdot\|$ is any norm.

Definition 0.4 (Polyhedron) *A* polyhedron *is the intersection of a finite set of closed halfspaces.*

Definition 0.5 (Polygon) *A* polygon *is a union of finite number of polyhedra.*

Definition 0.6 (Partition of a Set) *A* partition *of a set S is a collection of subsets of S such that the union of the subsets equal to S and the subsets are mutually disjoint.*

Definition 0.7 (Polyhedral Cover of a Set) *A* polyhedral cover *of a polyhedron S is a collection of subsets of S such that the union of the subsets equal to S, the subsets have non-intersecting interiors and each subset is a polyhedron.*

Definition 0.8 (Piecewise Affine (PWA) Function) *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is* piecewise affine *(PWA) on its domain if the domain is the union of finitely many open, closed and/or neither open nor closed polyhedra, relative to each of which $f(\cdot)$ is affine.*

Definition 0.9 (Power Set) *If $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$, then 2^Y is the* power set *(set of all subsets) of Y.*

Definition 0.10 (Set-Valued Map) *If $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$, then a set-valued map is defined as $F : X \to 2^Y$.*

Several places throughout this thesis double arrows are used to specify that a mapping is set-valued, i.e. set-valued maps are specified as $F : X \rightrightarrows Y$.

Definition 0.11 (Selection of a Set-Valued Map) *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is a* selection *of the set-valued map $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ if $f(x) \in F(x)$ for all x belonging to the domain of F.*

Contents

Chapter 1

Introduction

Receding Horizon Control (RHC) or Model Predictive Control (MPC) has been applied with great success in the process industry over the last decades. The success of MPC within this industry is mainly due to the method's ability to handle constraints for complex multi-variable systems. The so-called control allocation approach to controller synthesis is an optimization based method for distributing control actions within a redundant set of actuators and effectors. Common for methods in control that are based on constrained optimization is that they have until recently been limited to slow systems due to the high online computational load. Recently it has been shown that some constrained MPC and control allocation problems can be cast as parametric programs and solved explicitly, and hence, making the methods applicable to faster systems. Consequently, parametric programming (see e.g. (Bank, Guddat, Klatte, Kummer, and Tammer 1983; Gal 1995; Gal and Nedoma 1972; van der Panne 1975; Schechter 1987; Yu and Zeleny 1976; Zhang and Liu 1990; Fiacco 1983; Gal and Greenberg 1997; Berkelaar, Roos, and Terlaky 1997; Best and Ding 1972; Gal 1997; Gal 1980; Geoffrion and Graves 1977; Luc and Dien 1997) and references therein) has been subject to a resurgence of interest (Bemporad, Morari, Dua, and Pistikopoulos 2002; Acevedo and Pistikopoulos 1997; Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006b; Baotić 2002; Borrelli, Bemporad, and Morari 2003; Dua, Bozinis, and Pistikopoulos 2002; Dua and Pistikopoulos 2000; Tøndel, Johansen, and Bemporad 2003c; Spjøtvold, Kerrigan, Jones, Johansen, and Tøndel 2004; Spjøtvold, Tøndel, and Johansen 2005b; Spjøtvold, Tøndel, and Johansen 2005a; Spjøtvold 2005; Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006a; Jones, Kerrigan, and Maciejowski 2007; Spjøtvold, Tøndel, and Johansen 2007; Jones and Maciejowski 2006; Bemporad and Filippi 2006; Jones and Morari 2006) in recent years.

The goal in parametric programming is to solve a parameter dependent optimization problem for all possible values of the parameter. A solution to a parametric optimization problem is typically a piecewise function defined on a partition of the parameter space. In particular, the observation that the Constrained Linear Quadratic Regulator Problem could be solved via parametric programming (Bemporad, Morari, Dua, and Pistikopoulos 2002) by viewing the initial state as a vector of parameters and thereby moving the online optimization off-line, triggered research on several topics. Before we give a short introduction to some of these topics, we will in the next two sections recall the fundamentals in parametric programming and explicit model predictive control.

1.1 Parametric Programming Fundamentals

Consider the parametric optimization problem:

$$J^*(\theta) := \inf_x \{ f(x,\theta) \mid (x,\theta) \in P \}, \tag{1.1}$$

where $x \in \mathbb{R}^{n_x}$, $\theta \in \mathbb{R}^{n_\theta}$ and $P \subseteq \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta}$. The goal in parametric programming is to solve (1.1) for all values of $\theta \in \Theta$, where the set Θ is usually defined as the domain (or a subset) of J^*. At first glance solving (1.1) may seem like an impossible task since Θ is an uncountable set. However, it turns out that when $f(\cdot)$ and P have certain structures, a finite set of functions that are optimal for (1.1) when restricted to a subset of Θ, can be identified.

Definition 1.1 (Solutions to parametric optimization problems) *A solution to the problem* (1.1) *is defined as a function $x^* : \Theta \to \mathbb{R}^{n_x}$ such that*

$$x^*(\theta) \in \arg\min_x \{ f(x,\theta) \mid (x,\theta) \in P \}$$

for all $\theta \in \Theta$. Moreover, we say that an exact representation of the solution to (1.1) *exists if and only if there exists a finite set of functions $\{x_1^*(\cdot), x_2^*(\cdot), \dots, x_K^*(\cdot)\}$ and a finite collection of sets $\{R_1, R_2, \dots, R_K\}$ such that $x^*(\theta) = x_i^*(\theta)$ if $\theta \in R_i$ and $\{R_1, R_2, \dots, R_K\}$ forms a partition of Θ.*

An algorithm used to obtain the functions $\{x_1^*(\cdot), x_2^*(\cdot), \dots, x_K^*(\cdot)\}$ and the associated collection of sets $\{R_1, R_2, \dots, R_K\}$ is referred to as *parametric programming*. The solution properties for the most common parametric optimization problems for which an exact representation of the solution can be obtained are summarized below. The results for pLPs are taken from (Bank, Guddat, Klatte, Kummer, and Tammer 1983; Gal and Nedoma 1972; Borrelli, Bemporad, and Morari 2003), for pQPs (Bemporad, Morari, Dua, and Pistikopoulos 2002; Bank, Guddat, Klatte, Kummer, and Tammer 1983) and for pMILPs (Bank, Guddat, Klatte, Kummer, and Tammer 1983).

Theorem 1.1 (Parametric linear program (pLP)) *Consider* (1.1) *and let*

$$\begin{aligned} f(x,\theta) &:= c^T x, \\ P &:= \{ (x,\theta) \mid Ax + S\theta \le b \}, \end{aligned}$$

where c, A, S and b are matrices with suitable dimensions. We have:

(i) *There exists a function* $x^* : \Theta \to \mathbb{R}^{n_x}$ *that is continuous, piecewise affine (PWA) and satisfies*

$$x^*(\theta) \in \arg\min_x \{f(x,\theta) \mid (x,\theta) \in P\}, \quad \forall \theta \in \Theta.$$

(ii) *The function* $J^* : \Theta \to \mathbb{R}$ *is continuous, convex and PWA.*

(iii) *The domain of* J^* *is convex, closed and polyhedral.*

Theorem 1.2 (Parametric quadratic program (pQP)) *Consider* (1.1) *and let*

$$\begin{aligned} f(x,\theta) &:= \frac{1}{2}x^T H x + \theta^T F^T x + c^T x, \\ P &:= \{(x,\theta) \mid Ax + S\theta \le b\}, \end{aligned}$$

where c*,* A*,* S*,* b*,* F *and* $H = H^T \ge 0$ *are matrices with suitable dimensions. We have:*

(i) *There exists a function* $x^* : \Theta \to \mathbb{R}^{n_x}$ *that is PWA and satisfies*

$$x^*(\theta) \in \arg\min_x \{f(x,\theta) \mid (x,\theta) \in P\}, \quad \forall \theta \in \Theta.$$

(ii) *The function* $J^* : \Theta \to \mathbb{R}$ *is continuous, convex and piecewise quadratic.*

(iii) *The domain of* J^* *is convex and polyhedral.*

(iv) *If* $H > 0$*, then* $x^*(\cdot)$ *is unique and continuous on* Θ *and the domain of* J^* *is closed.*

Theorem 1.3 (Parametric mixed-integer linear program) *Consider* (1.1) *and let* $x \in \mathbb{R}^{n_r} \times \{0,1\}^{n_b}$ *where* $n_x = n_r + n_b$ *and*

$$\begin{aligned} f(x,\theta) &:= c^T x + d^T \theta, \\ P &:= \{(x,\theta) \mid Ax + S\theta \le b\}, \end{aligned}$$

where c*,* A*,* S*,* b*,* d *are matrices with suitable dimensions. We have:*

(i) *There exists a function* $x^* : \Theta \to \mathbb{R}^{n_x}$ *that is PWA and satisfies*

$$x^*(\theta) \in \arg\min_x \{f(x,\theta) \mid (x,\theta) \in P\}, \quad \forall \theta \in \Theta.$$

(ii) *The function* $J^* : \Theta \to \mathbb{R}$ *is PWA.*

1.2 Explicit Model Predictive Control

We recall the traditional approach utilizing online optimization in Section 1.2.1 and the explicit version is presented in Section 1.2.2.

1.2.1 Model predictive Control

Consider the discrete-time system on the form:

$$x^+ = f(x, u),$$

where x is the (measured) state, x^+ is the successor state, u is the input, $f(\cdot)$ is the system update function. The state and input are subject to constraints $(x, u) \in \mathcal{Y}$. Let $\pi := \{u_0, u_1, \ldots, u_{N-1}\}$ denote a control sequence over the horizon N. Moreover, let $\phi(i; x, \pi)$ denote the solution to $x^+ = f(x, u)$ at iteration i for the initial state x and control sequence π. The cost is defined as

$$V_N(x, \pi) := V_f(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i),$$

where $x_i := \phi(i; x, \pi)$. The optimal control problem considered is given by

$$\mathbb{P}(x) : \; V^*(x) := \min_{\pi \in \Pi_N(x)} V_N(x, \pi), \tag{1.2}$$

where the set of admissible control sequences is

$$\Pi_N(x) := \left\{ \pi \;\middle|\; \begin{array}{l} (x_i, u_i) \in \mathcal{Y},\; i = 0, 1, \ldots, N-1, \\ x_N \in X_f \end{array} \right\}.$$

and X_f is some terminal constraint. The prediction horizon N, terminal weight function $V_f(\cdot)$ and terminal set X_f are often chosen such that one can find an optimal feedback controller $u_0 = \gamma(x)$ that renders X_f forward invariant under $x^+ = f(x, \gamma(x))$. Model predictive control is a form of optimal control where the following procedure is utilized:

1. Measure the current state x of the system.
2. Solve $\mathbb{P}_N(x)$ to obtain an optimal $\pi(x)$.
3. Apply the control $u_0(x)$ to the plant.
4. Return to step 1.

Model predictive control was first introduced for linear discrete-time systems on the form

$$x^+ = Ax + Bu,$$

where A and B are matrices with suitable dimensions. For linear systems the the stage cost $l(\cdot)$ and terminal cost $V_f(\cdot)$ are assumed to be linear or quadratic ($p \in \{1, 2, \infty\}$):

$$\begin{aligned} l(x, u) &:= \|Qx\|_p + \|Ru\|_p, \\ V_f(x) &:= \|Px\|_p, \end{aligned}$$

where P, Q, and R are suitably defined weight matrices. Moreover, $\mathcal{Y}$ and X_f are assumed to be closed polyhedra. If $p \in \{1, \infty\}$, then the optimal control problem $\mathbb{P}_N(\cdot)$ is a linear program. On the other hand, if $p = 2$ (or to be accurate, the quadratic norm is used), then $\mathbb{P}_N(\cdot)$ is a quadratic program. For more details on MPC and how to choose N, X_f and $V_f(\cdot)$ the reader is referred to (Mayne, Rawlings, Rao, and Scokaert 2000) and references therein.

1.2.2 Model predictive Control via Parametric Programming

Under some assumptions on the weight matrices, the optimal control problem $\mathbb{P}_N(\cdot)$ can be cast as either a parametric linear program (if one or infinity norms are used) or as a parametric quadratic program (if the quadratic norm is used). The linear version can be formulated as:

$$J^*(x) := \min_{z \in P(x)} \; c^T z, \tag{1.3a}$$

$$P(x) = \{z \mid Cz + Dx \leq e\}, \tag{1.3b}$$

where $z := [u_0^T \; u_1^T \; \ldots \; u_{N-1}^T \; \varepsilon^T]^T$, ε is a vector of slack variables and c, C, D and e are suitably defined matrices.

If quadratic norms are used and $P = P^T \geq 0$, $R = R^T > 0$ and $Q = Q^T \geq 0$, then $\mathbb{P}_N(\cdot)$ can be recast a strictly convex parametric program:

$$J^*(x) := \min_{\pi \in \Pi(x)} \; \pi^T H \pi, \tag{1.4}$$

where $H = H^T > 0$ is a suitably defined matrix.

Recalling the solution properties for (1.3) and (1.4) from Section 1.1 it easy to see that optimal control feedback law is PWA and can be represented as:

$$\pi^*(x) = K_i x + k_i \quad \text{if} \quad x \in P_i,$$

where $\mathcal{P} := \cup_{i=1}^{I} P_i$ forms a polyhedral cover of the part of the state space that renders $\mathbb{P}_N(\cdot)$ feasible and bounded, and each K_i and k_i are matrices. The online optimization in receding horizon control can be moved off-line and the approach is modified to

1. Measure the current state x of the system.
2. Identify the the region P_i of the state space cover $\mathcal{P}$ that contains the current state (this will be referred to as the point location problem).
3. Apply the first element $u_0^*(x)$ of the optimal control sequence $\pi(x)$ associated with the region identified in step 2 to the plant.
4. Return to step 1.

1.2.3 Pros and cons with explicit MPC

The main advantages of the explicit approach are: $i)$ removing the need for sophisticated optimization software on the processor, $ii)$ the correctness of the solution can be verified off-line, which is a key issue in safety critical applications, $iii)$ the worst case number of arithmetic operations needed to find the solution can easily be computed, $iv)$ the average and worst case number of arithmetic operations needed to find the solution is usually greatly reduced, and $v)$ evaluation of the PWA function can be implemented using fixed point arithmetic. The main drawbacks, on the other hand, are that $i)$ the problem class for which this solution strategy is applicable is limited, and in cases where an exact solution can be found $ii)$ obtaining an explicit solution may be computationally intractable and $iii)$ the storage space required to represent the solution may exceed the available memory.

1.3 Recent research on parametric programming and its applications

As mentioned earlier, the observation that explicit solutions to MPC problems could be computed explicitly by utilizing parametric programming triggered research on several related topics. We briefly summarize some of the research in the next subsections.

1.3.1 Efficient evaluation of piecewise affine functions

Both parametric linear programs (pLP) and convex parametric quadratic programs (pQP) yield piecewise affine optimizer functions. As model predictive control problems can be formulated as both pLPs and pQPs, the need to evaluate a piecewise affine function at a high frequency arose. More specifically, given a point in the parameter space, which piece of the piecewise function is optimal. This problem is in many cases equivalent to the *point location problem* in geometry; given a collection of sets and a point contained in the union of these sets, which set contains the point? Efficient methods are presented in (Christophersen, Kvasnica, Jones, and Morari 2007; Tøndel, Johansen, and Bemporad 2003b; Jones, Grieder, and Raković 2006).

1.3.2 Improved parametric programming algorithms

Several problems in control theory can be formulated as parametric programs with a certain structure. Some of the resulting parametric programs include parametric linear programs (pLP), parametric quadratic programs (pQP), parametric mixed integer programs (pMILP), parametric non-linear programs (pNLP) and parametric linear complementarity problems (pLCP). However, for some problems algorithms did not exist, were too slow or too numerically sensitive to be applied to problems of moderate size, and consequently new algorithms were developed. Recent algorithms for pLPs can be found in (Jones, Kerrigan, and Maciejowski 2007; Borrelli, Bemporad, and Morari 2003; Spjøtvold, Tøndel, and Johansen 2005a; Jones and Maciejowski 2006)), for pQPs in (Bemporad, Morari, Dua, and Pistikopoulos 2002; Tøndel, Johansen, and Bemporad 2003a; Seron, Goodwin, and Doná 2003; Baotić 2002; Spjøtvold, Tøndel, and Johansen 2007), for pMILPs in (Acevedo and Pistikopoulos 1997; Dua, Bozinis, and Pistikopoulos 2002), for pNLP in (Johansen 2002; Bemporad and Filippi 2006) and for pLCPs in (Jones and Morari 2006). Some problem classes are inherently difficult and it is therefore reason to believe that new and improved algorithms still can be developed.

1.3.3 Algorithms for obtaining approximate and fixed complexity solutions to parametric programs

Researchers quickly realized that the solution complexity of the MPC problems at hand were sometimes so high that it was computationally intractable to obtain the solution. Moreover, often the pieces of the function where almost identical for ”neighboring” pieces. This motivated research into how one could obtain low complexity approximate solutions (Johansen 2004a; Bemporad and Filippi 2006; Bemporad and Filippi 2003; Johansen and Grancharova 2002; Filippi 2004), or even fixed complexity approximate solutions (Jones, Barić, and Morari 2007).

1.3.4 Obtaining explicit solutions to constrained control allocation problems

Clearly, model predictive control problems are not the only problems in control theory that can be formulated as parametric programs. In constrained control allocation (see e.g. (Bodson 2002) and references therein) the task is to generate a specified generalized force/virtual control from a redundant set of actuators while fulfilling the constraints on the actuators and control effectors. When the problem has a solution there is often an uncountable number of combinations of control inputs that achieve the desired generalized force/vitual control. This redundancy leaves room for optimizing the performance, for instance minimizing energy consumption and/or mechanical tear and wear. Since constrained control allocation is a special form of inner loop controller synthesis, where the inner loop is often regarded as instantaneous, the need for reliable and quick optimization was acknowledged. Explicit solutions to control allocation problems are utilized in e.g. (Johansen, Fuglseth, Tøndel, and Fossen 2007; Spjøtvold and Johansen 2007; Johansen, Fossen, and Tøndel 2005).

1.3.5 Link between parametric programming and geometry

Projection of sets from higher to lower dimensions is a difficult problem that has been the subject of substantial research. There are similarities between projecting a polyhedron to a lower dimension and solving a parametric linear program. The similarity is apparent from the fact that both the epigraph of the cost function for a parametric linear program and the set of parameters for which the the minimum is bounded, are polyhedra. These facts motivated utilizing ideas from projection in parametric programming and vice versa, see (Jones, Kerrigan, and Maciejowski 2008; Jones 2005). Authors have also pointed out that parametric programming can be used to compute Voronoi diagrams and Delauny triangulations, which are fundamental structures in geometry, see (Raković, Grieder, and Jones 2004).

1.3.6 Obtaining explicit solutions to advanced formulations of model predictive control problems

There exists formulations for model predictive control for non-linear systems and piecewise affine systems. Consequently, algorithms for solving the resulting parametric programs have been developed, see e.g. (Johansen 2002; Dua and Pistikopoulos 2000; Mayne and Raković 2002). In addition, parametric programming is used as a tool in dynamic programming approaches to optimal control (de la Peña, Alamo, Bemporad, and Camacho 2002; Raković, Kerrigan, Mayne, and Lygeros 2006; Grieder, Kvasnica, Baotić, and Morari 2005; Spjøtvold, Kerrigan, Raković, Johansen, and Mayne 2007a; Grancharova and Johansen 2005). Dynamic programming is often used for robust optimal control, commonly referred to as min-max control. In this case optimization is performed over control policies instead of control sequences and parametric programming is a tool in these algorithms. Explicit solutions to minimum-time optimal control can also be obtained by means of parametric programming (Grieder, Kvasnica, Baotić, and Morari 2005).

1.3.7 Explicit solutions to constrained estimation

Moving horizon constrained estimation (see (Rao, Rawlings, and Lee 2001) for details) is a problem that is closely related to receding horizon control. Not surprisingly one can also obtain explicit solutions to this problem by utilizing parametric programming (Zhuo, Doná, and Seron 2005).

1.4 Motivation, contribution and thesis structure

The aim of this thesis is not to give a complete overview of parametric programming or all its applications in control theory, but discuss in more detail some selected topics and application areas. As this thesis is organized as paper collection, the detailed introductions to the different topics are deferred to the respective chapters.

The first part of the thesis deals with two theoretical topics within parametric programming. Chapter 2 is based on (Spjøtvold, Tøndel, and Johansen 2005a; Spjøtvold, Tøndel, and Johansen 2005b; Spjøtvold, Tøndel, and Johansen 2007) and focus on unique polyhedral representations and continuous selections to convex parametric quadratic programs (pQP). The motivation for this chapter was first and foremost to obtain continuous control laws for model predictive control problems based on linear programming. However, in addition to our theoretical interest in the topic, it turns out that convex pQPs arise in several areas of control theory. Some of these topics are $i)$ Control Allocation: there is no *correct* choice for cost function in control allocation and the author has experienced that some natural formulations lead to a convex, as opposed a strictly convex, QP. For instance, using the quadratic norm to penalize the difference between the actual and desired generalized force results in a convex pQP. $ii)$ If slack variables are added to the QP problem and an exact penalty function is desired, which is the case for the soft constrained linear-quadratic regulator and for some constrained control allocation problems, the QP becomes convex. $iii)$ Sub-problems in parametric nonlinear programming algorithms are convex pQPs (Johansen 2002).

Chapter 3 is based on (Spjøtvold, Kerrigan, Jones, Johansen, and Tøndel 2004; Spjøtvold 2005; Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006b; Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006a). We consider strictly convex pQPs and focus on a particular property of the solution that is referred to as the *facet-to-facet* property. It is relatively simple to set up examples for which the facet-to-facet property fails if the pQP is convex as opposed to strictly convex (Spjøtvold 2005). For strictly convex pQPs it does not fail that often, but we show by example that it does not hold in general. The motivation behind this chapter is twofold, the first being to point out a fundamental difference between the geometry of the solution to pQPs and pLPs. Secondly, some previously developed algorithms for strictly convex pQPs, although seemingly working well in practice, rely on the facet-to-facet property to be theoretically just.

The second part of the thesis treats some selected applications of parametric programming in control theory. In Chapter 4 we consider $\inf-\sup$ (commonly referred to as $\min-\max$) control of constrained discontinuous discrete–time piecewise affine systems subject to state- and input-dependent disturbances. The main motivation behind this article was that for the mentioned problem class, a solution might not exist. Consequently, a procedure that finds an optimal solution when one exists and a sub-optimal when one does not, was needed. Chapter 4 is based on (Spjøtvold, Kerrigan, Raković, Johansen, and Mayne 2007b; Spjøtvold, Kerrigan, Raković, Johansen, and Mayne 2007a)

Chapter 5 is based on (Spjøtvold, Raković, Tøndel, and Johansen 2006). Here we propose a method for evaluating piecewise affine control laws. Recently results in reachability analysis for discrete-time dynamical systems allow for the polyhedral sets on which each affine control law is defined to be mapped one step forward in time, yielding the so-called *one-step forward reach set.* The reach set is then utilized to reduce the set of polyhedral regions that can contain the system state at the next sample instant. The main motivation for this paper was to improve the average time the microchip spends on evaluating the feedback law and thereby saving energy or freeing the processor for other tasks.

The third part of the thesis focus on constrained control allocation. Chapter 6 presents results for control allocation for a scale model of a thruster-controlled floating platform. Chapter 6 considers a convexification of the allocation problem and an explicit solution is found. In addition, an explicit solution for any single point failure (single thruster, machine room or switchboard) are computed. We present experimental results form The Marine Cybernetics Laboratory at NTNU in Trondheim. The main motivation behind the Chapter 6 is to illustrate the usefulness of explicit solutions to control allocation problems. Moreover, the control allocation for the thruster-controlled problem is a challenging problem, for which no satisfactory method have been presented in the literature. Chapter 6 is based on (Spjøtvold and Johansen 2007).

In Chapiter 7 a decomposition strategy for constrained linear control allocation problems is presented. This chapter is based on (Spjøtvold, Tøndel, and Johansen 2006). In the decomposition strategy we divide the control allocation problem into a master and a set of sub-problems, where the solution of the master problem is input to the sub-problems. The main motivation behind the decomposition strategy is to provide a flexible framework for control allocation synthesis that allows for a mix of online optimization and explicit solutions to the problem. Moreover, features, such as taking into account actuator/effector dynamics, can be incorporated in sub-problems, yielding a modular design. Finally, we briefly discuss how the decomposition strategy can be utilized to obtain a solution to a non-convex version of the allocation problem in Chapter 6.

In Chapter 8 a brief conclusion on the various topics presented in the thesis is given. We also give some comments on future research directions.

1.5 Publications

Most of the material presented in this thesis has either been published, accepted or recently been submitted for publication. Below we give the connections between the papers and the chapters of the thesis:

- Chapter 2: (Spjøtvold, Tøndel, and Johansen 2005a; Spjøtvold, Tøndel, and Johansen 2005b; Spjøtvold, Tøndel, and Johansen 2007)
- Chapter 3: (Spjøtvold, Kerrigan, Jones, Johansen, and Tøndel 2004; Spjøtvold 2005; Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006b; Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006a)
- Chapter 4: (Spjøtvold, Kerrigan, Raković, Johansen, and Mayne 2007b; Spjøtvold, Kerrigan, Raković, Johansen, and Mayne 2007a)
- Chapter 5: (Spjøtvold, Raković, Tøndel, and Johansen 2006)
- Chapter 6: (Spjøtvold and Johansen 2007)
- Chapter 7: (Spjøtvold, Tøndel, and Johansen 2006)

Part I

Selected Topics in Parametric Programming

Chapter 2

A Continuous Selection and Unique Polyhedral Representation of Solutions to Convex Parametric Quadratic Programs

2.1 Introduction

Substantial work has been done on the continuity properties of the value function and optimal solution set for parametric programs (Fiacco 1983; Zhao 1997; Bank, Guddat, Klatte, Kummer, and Tammer 1983; Best and Ding 1972; Hogan 1973a; Hogan 1973b). Continuity of the optimal set mapping is closely related to the stability of the optimization problem and the stability of quadratic programs is studied in (Best and Chakravarti 1990) and (Phu and Yen 2001). Continuity and stability results for parametric programs are often derived from set theory presented in (Berge 1963), (Dantzig, Folkman, and Shapiro 1967) and (Hausdorff 1957).

The algorithm presented by Bemporad et al. (Bemporad, Morari, Dua, and Pistikopoulos 2002) obtains solutions to strictly convex parametric quadratic programs (pQPs). With some modifications, it can also be used for convex problems (Tøndel, Johansen, and Bemporad 2003c). Borrelli et al. (Borrelli, Bemporad, and Morari 2003) proposed a geometric algorithm for parametric linear programs (pLPs), which is fundamentally different from the algorithm by Gal and Nedoma (Gal and Nedoma 1972). The algorithm explores the parameter space in the same manner as in (Bemporad, Morari, Dua, and Pistikopoulos 2002). Common to the algorithms for convex pQPs, including the geometric algorithm for pLPs, is that the optimizer function may be discontinuous even if the optimal solution set is a continuous point-to-set map and therefore admits a continuous selection (Michaels

1956). Moreover, the set of polyhedra associated with the piecewise affine optimizer function is generally non-unique.

Our main motivation for obtaining continuous solutions to convex pQPs, besides the theoretical aspects, is that this problem arises in explicit model predictive control with a linear cost function (Bemporad, Borrelli, and Morari 2002). It is illustrated in (Bemporad, Borrelli, and Morari 2002) that the solution to an MPC problem with a linear cost function may be discontinuous if the algorithm in (Borrelli, Bemporad, and Morari 2003) is used. Discontinuities of the optimizer function may lead to chattering in an optimal control approach, and hence, a method which yields a continuous optimizer function is desirable. A unique representation also gives benefits in terms of i) Repeatability: The same solution is always obtained. ii) Parallelization: The parameter space can be divided into subsets that are explored individually without increasing the solution complexity when the results are merged. iii) Exploration strategy: We have some freedom in the choice of algorithm used to explore the parameter space.

This chapther is based on (Spjøtvold, Tøndel, and Johansen 2005a; Spjøtvold, Tøndel, and Johansen 2005b; Spjøtvold, Tøndel, and Johansen 2007) and is organized as follows: We first point out that using the normal cone optimality condition to construct parametric regions of optimality for strictly convex pQPs (Mayne and Raković 2003) results in a unique collection of polyhedral sets. For convex pQPs a unique set of non-intersecting polyhedra is obtained by always choosing the optimizer with the least Euclidian norm and using the normal cone optimality condition. If the pQP has non-unique solutions, a strictly convex pQP is formulated such that the norm of the solution vector is minimized subject to the optimality conditions of the original problem. We prove that if the optimal set mapping for the convex pQP is continuous, the minimum norm selection will be a continuous mapping from the feasible subset of the parameter space to the solution space.

2.2 Preliminaries

2.2.1 Notation and basic definitions

If $\mathcal{I}$ is an index set, then $|\mathcal{I}|$ denotes the cardinality of $\mathcal{I}$ and $\mathcal{I}_i$ refers to the i^{th} element in $\mathcal{I}$. When referring to a set of indices $\mathcal{I}$, we assume that the set is ordered, i.e. for the i^{th} element in $\mathcal{I}$ we have $\mathcal{I}_i < \mathcal{I}_j$, $\forall j \in \{i+1, \ldots, |\mathcal{I}|\}$. If $A \in \mathbb{R}^{n\times m}$ is a matrix or column vector, then $A_i \in \mathbb{R}^{1\times m}$ denotes the i^{th} row of A and $A_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|\times m}$ denotes the matrix $\left[A_{\mathcal{I}_1}^T, \ldots, A_{\mathcal{I}_{|\mathcal{I}|}}^T\right]^T$. If $f : X \to Y$ is a function, then the restriction of f to the domain $D \subseteq X$ is written $f|_D : D \to Y$. The *closure*, *interior*, and *boundary* of a set S is denoted $\mathrm{cl}(S)$, $\mathrm{int}(S)$, and $\mathrm{bd}(S)$, respectively. The abbreviation s.t. will denote *subject to*.

Recall that the set of affine combinations of points in a set $S \subset \mathbb{R}^n$ is called the *affine hull* of S, and is denoted $\mathrm{aff}(S)$. The *dimension of a set* $S \subset \mathbb{R}^n$ is the dimension of $\mathrm{aff}(S)$, and is denoted $\dim(S)$; if $\dim(S) = n$, then S is said to be full-dimensional. A *polyhedron* is the intersection of a finite number of closed halfspaces. A non-empty set F is a *face* of the polyhedron $P \subset \mathbb{R}^n$ if there exists a hyperplane $\{z \in \mathbb{R}^n \mid a^T z = b\}$, where $a \in \mathbb{R}^n$, $b \in \mathbb{R}$, such that $F = P \cap \{z \in \mathbb{R}^n \mid a^T z = b\}$ and $a^T z \leq b$ for all $z \in P$. Given an s-dimensional polyhedron $P \subset \mathbb{R}^n$, where $s \leq n$, the *facets* of P are the $(s-1)$-dimensional faces of P.

2.2.2 Problem setup

The problem that will be considered is given by

$$J^*(\theta) := \min_{x \in \mathbb{R}^n} f(x,\theta) \quad \text{s.t.} \quad Ax \leq b + S\theta, \tag{2.1}$$

where $f(x,\theta) := \frac{1}{2}x^T H x + \theta^T F^T x + c^T x$, $\theta \in \mathbb{R}^s$ is the *parameter* of the optimization problem, and the vector $x \in \mathbb{R}^n$ is to be optimized for all values of $\theta \in \Theta$, where $\Theta \subseteq \mathbb{R}^s$ is a polyhedral set such that the minimum in (2.1) exists. Moreover, $H = H^T \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $A \in \mathbb{R}^{q \times n}$, $b \in \mathbb{R}^{q \times 1}$ and $S \in \mathbb{R}^{q \times s}$. If in addition, $H \geq 0$ or $H > 0$, the pQP is convex or strictly convex, respectively. If $H = 0$, then (2.1) is a special form of parametric linear programs (pLP), which is a subclass of the problem addressed in this chapter. The point-to-set maps $X : \Theta \to 2^{\mathbb{R}^n}$ and $X^* : \Theta \to 2^{\mathbb{R}^n}$ are defined as $X(\theta) := \{x \in \mathbb{R}^n \mid Ax \leq b + S\theta\}$ and $X^*(\theta) := \{x \in \mathbb{R}^n \mid Ax \leq b + S\theta, f(x,\theta) = J^*(\theta)\}$, respectively. The sets $X(\theta)$ and $X^*(\theta)$ are referred to as the *feasible*- and *optimal set*, respectively.

Without loss of generality, the following standing assumption is made (Bemporad, Morari, Dua, and Pistikopoulos 2002; Borrelli, Bemporad, and Morari 2003):

Assumption 2.1 *The set of admissible parameters Θ is full-dimensional.*

Definition 2.1 (Active set) *Let x be a feasible point of* (2.1) *for a given θ. The* active constraints *are the constraints that fulfill $A_i x - b_i - S_i\theta = 0$, and the* inactive constraints *are the constraints that fulfill $A_i x - b_i - S_i\theta < 0$. The* active set *$\mathcal{A}(x,\theta)$ is the set of indices of the active constraints, that is,*

$$\mathcal{A}(x,\theta) := \{i \in \{1,2,\ldots,q\} | A_i x - b_i - S_i\theta = 0\}.$$

Definition 2.2 (Optimal active set) *Let θ be given. The* optimal active set *$\mathcal{A}^*(\theta)$ is the set of indices of the constraints that are active for all $x \in X^*(\theta)$, that is,*

$$\mathcal{A}^*(\theta) := \{i | i \in \mathcal{A}(x,\theta), \forall x \in X^*(\theta)\} = \bigcap_{x \in X^*(\theta)} \mathcal{A}(x,\theta).$$

Definition 2.3 (Critical region) *Given an index set $\mathcal{A} \subseteq \{1,2,\ldots,q\}$, the critical region $\Theta^{\mathcal{A}}$ associated with $\mathcal{A}$ is the set of parameters for which the optimal active set is equal to $\mathcal{A}$, i.e.*

$$\Theta^{\mathcal{A}} := \{\theta \in \Theta \mid \mathcal{A}^*(\theta) = \mathcal{A}\}.$$

In the above definition, note that if $\mathcal{A}$ is not the optimal active set for some parameter, then $\Theta^{\mathcal{A}}$ is the empty set. Hence, when referring to a critical region $\Theta^{\mathcal{A}}$, we will assume that $\mathcal{A}$ is the optimal active set for some $\theta \in \Theta$.

Definition 2.4 (LICQ) *For a non-empty index set $\mathcal{A} \subseteq \{1, 2, \ldots, q\}$, we say that the linear independence constraint qualification (LICQ) holds for $\mathcal{A}$ if the gradients of the set of constraints indexed by $\mathcal{A}$ are linearly independent, i.e. $A_{\mathcal{A}}$ has full row rank.*

The following was established in (Bemporad, Morari, Dua, and Pistikopoulos 2002; Tøndel, Johansen, and Bemporad 2003c):

Theorem 2.1 (Solution properties) *Consider the pQP in* (2.1).

i) There exists a minimizer function $x^* : \Theta \rightarrow \mathbb{R}^n$, $\theta \mapsto x^*(\theta) \in X^*(\theta)$, that is piecewise affine (PWA) in the sense that there exists a finite set of full-dimensional polyhedra $\mathcal{R} := \{R^1, \ldots, R^K\}$ such that $\Theta = \cup_{k=1}^K R^k$, $\mathrm{int}(R^i) \cap \mathrm{int}(R^j) = \emptyset$ for all $i \neq j$ and the restriction $x^*|_{R^k}(\cdot)$ is affine for all $k \in \{1, \ldots, K\}$.

ii) The value function $J^* : \Theta \rightarrow \mathbb{R}$ is continuous and piecewise quadratic (PWQ) in the sense that the restriction $J^*|_{R^k}(\cdot)$ is quadratic for all $R^k \in \mathcal{R}$.

In the sequel we let $x^* \in X^*(\theta)$ denote a minimizer to (2.1) for a given θ, $x^*(\cdot)$ denotes a piecewise affine selection function, and $x^*(\theta)$ denotes $x^*(\cdot)$ evaluated at θ.

Our main objective is to obtain a unique set of full-dimensional polyhedra $\mathcal{R}$ and simultaneously ensure that the function $x^*(\cdot)$ is continuous. Thus, given $\theta \in \Theta$, we will first consider how to select the affine restriction $x^*|_{R^k}(\cdot)$ and find its polyhedral domain R^k. A polyhedral domain $R^k \in \mathcal{R}$ will be referred to as a *sub-region* since in subsequent sections it will become apparent that each R^k must be contained in the closure of a critical region. For the purpose of computing the domain R^k, the next section summarizes the normal cone optimality condition.

2.2.3 Normal cone optimality condition

Recall that a set C is called a *cone* if for every $x \in C$ and scalar $\xi \geq 0$, we have $\xi x \in C$. Moreover, a cone represented by the intersection of a finite number of closed half-spaces is called an $\mathcal{H}$*-cone*, and the set of all nonnegative combinations of a set of vectors $\{v_1, \ldots, v_n\}$ is called a $\mathcal{V}$*-cone*. If A is a matrix, then $\mathrm{cone}(A)$ denotes the set of all nonnegative combinations ($\mathcal{V}$*-cone*) of the column-vectors of A.

Consider the following problem:

$$
\begin{aligned}
&z^* := \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in \Omega, \\
&\Omega := \{x \in \mathbb{R}^n \mid g_j(x) \leq 0, i \in \mathcal{I}, h_j(x) = 0, j \in \mathcal{J}\},
\end{aligned}
$$

where $f(\cdot)$, $g_i(\cdot)$ and $h_j(\cdot)$ are smooth, real-valued, functions defined on a subset of $\mathbb{R}^n$ and the index sets $\mathcal{I}$ and $\mathcal{J}$ contain a finite number of elements. The following definitions are taken from (Nocedal and Wright 1999):

Definition 2.5 (Tangent and Normal cones) *i) A vector $w \in \mathbb{R}^n$ is tangent to Ω at $x \in \Omega$ if for all vector sequences $\{x_i\}$ with $x_i \to x$ and $x_i \in \Omega$, and all positive scalar sequences $t_i \downarrow 0$, there is a sequence $w_i \to w$ such that $x_i + t_i w_i \in \Omega$ for all i.*

ii) The tangent cone $T^{\Omega}(x)$ is the collection of all tangent vectors to Ω at x.

iii) The normal cone to Ω at x, $N^{\Omega}(x)$, is the orthogonal complement of the tangent cone, that is

$$N^{\Omega}(x) := \left\{ v | v^T w \leq 0, \quad \forall w \in T^{\Omega}(x) \right\}.$$

Theorem 2.2 (First order necessary optimality condition) *If $\bar{x}$ is a local minimizer of f in Ω, then*

$$-\nabla_x f(\bar{x}) \in N^{\Omega}(\bar{x}). \tag{2.2}$$

PROOF: *See (Nocedal and Wright 1999).*

If $f(\cdot)$ and Ω are convex, then $\bar{x}$ is a global minimum and (2.2) is also sufficient.

Given the polyhedron $P := \{x | Ax \leq b\}$, let $\bar{x}$ be a point on the boundary of P. Let $\mathcal{A}$ be the non-empty set of indices of the inequalities that are active at $\bar{x}$, hence $A_i \bar{x} = b_i$ for $i \in \mathcal{A}$, and $A_i \bar{x} < b_i$ for $i \notin \mathcal{A}$. Note that for a polyhedron the tangent cone $T^P(\bar{x})$ at $\bar{x}$, is equal to the set of feasible directions at $\bar{x}$ (Bertsekas, Nedic, and Ozdaglar 2003), i.e. $T^P(\bar{x}) := \{d | A_{\mathcal{A}} d \leq 0\}$. The normal cone at $\bar{x}$ is the $\mathcal{V}$-cone $N^P(\bar{x}) = \text{cone}(A_{\mathcal{A}}^T)$. The $\mathcal{H}$-cone representing $N^P(\bar{x})$ can always be written as

$$N^P(\bar{x}) = \{y \mid L_I y \leq 0, L_E y = 0\},$$

where L_I (L_E) is a matrix representing the inequality (equality) part of the cone. Note that an active set $\mathcal{A}$ at $\bar{x}$ uniquely defines a normal cone, thus, we introduce the notation $C^P(\mathcal{A}) := \left\{ y \mid L_I^{\mathcal{A}} y \leq 0, L_E^{\mathcal{A}} y = 0 \right\} = N^P(\bar{x})$. The optimality condition (2.2) becomes

$$L_I \nabla_x f(\bar{x}) \geq 0,$$
$$L_E \nabla_x f(\bar{x}) = 0.$$

2.2.4 Continuity properties of solutions to pQPs

Before we introduce the minimum norm selection method, we consider the continuity properties of the optimal set mapping for a convex pQP. For convenience we recall Berge's definitions of a continuous, upper semicontinuous, and lower semicontinuous point-to-set map (Berge 1963):

Definition 2.6 (Upper and lower semicontinuous point-to-set map) *The point-to-set map $P : X \to Y$ is* lower semicontinuous *at x_0, if for each open set Ω satisfying $\Omega \cap P(x_0) \neq \emptyset$ there is a neighborhood $U(x_0)$ such that*

$$x \in U(x_0) \Rightarrow P(x) \cap \Omega \neq \emptyset.$$

Moreover, P is lower semicontinuous on X if it is lower semicontinuous at each $x \in X$.

The point-to-set map $P : X \to Y$ is upper semicontinuous *at x_0, if for each open set Ω containing $P(x_0)$ there exists a neighborhood $U(x_0)$ such that*

$$x \in U(x_0) \Rightarrow P(x) \subset \Omega.$$

Moreover, P is upper semicontinuous on X if it is upper semicontinuous at each $x \in X$.

Definition 2.7 (Continuous point-to-set map) *The point-to-set map $P : X \to Y$ is continuous at x_0 if it is both upper semicontinuous and lower semicontinuous at x_0. It is continuous on X if and only if it is continuous at every $x \in X$.*

Theorem 2.3 *Consider problem (2.1) and let the point-to-set map $X^*(\cdot)$ be continuous on Θ. The PWA selection function $x^*(\cdot)$ with the least Euclidean norm is continuous on Θ.*

PROOF: *It is obvious that the problem of finding the minimum norm solution can be stated as:*

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T x \quad \text{s.t.} \quad x \in X^*(\theta).$$

The minimizer of a strictly convex function over a continuous point-to-set map is a continuous function (Berge 1963, Theorem VI.3.3), (Aubin and Frankowska 1990, Corollary 9.3.3).

$\square$

Since the existence of a continuous selection for problem (2.1) is ensured by the continuity of $X^*(\cdot)$ on Θ we state the following corollary based on (Bank, Guddat, Klatte, Kummer, and Tammer 1983, Theorem 3.2.2, Theorem 3.2.3, and Theorem 5.3.2), which shows that $X^*(\cdot)$ is in fact continuous in some cases.

Corollary 2.1 *Consider problem (2.1). The point-to-set map $X^*(\cdot)$ is continuous on Θ if*

i)
$$\forall \theta \in \Theta \; \nexists d \in \mathbb{R}^n \backslash \{0\} : Hd = 0 \wedge \; (c + F\theta)^T d = 0,$$

or

ii) $F = 0$.

PROOF: *Note first that $X^*(\cdot)$ is upper semicontinuous on Θ (Bank, Guddat, Klatte, Kummer, and Tammer 1983, Theorem 5.3.2).*

i) By (Bank, Guddat, Klatte, Kummer, and Tammer 1983, Theorem 3.2.3) $X^*(\cdot)$ is lower semicontinuous on Θ if the lineality space

$$M(\theta) := \{d \in \mathbb{R}^n | Hd = 0, (c + F\theta)^T d = 0, A_{\mathcal{A}^*(\theta)} d = 0\}$$

has the same dimension $\forall \theta \in \Theta$. If $i)$ holds, then $\dim(M(\theta)) = 0$, $\forall \theta \in \Theta$.

ii) It follows from (Bank, Guddat, Klatte, Kummer, and Tammer 1983, Theorem 3.2.2 and page 47) that if $X^*(\theta)$ can be written as

$$\{x \in \mathbb{R}^n \mid g_i(x, \theta) := h_i(x) + t_i(\theta) \le 0, i \in \mathcal{I}\}$$

where for all $i \in \mathcal{I}$ the functions $h_i(\cdot)$ are convex on $\mathbb{R}^n$, $t_i(\cdot)$ are continuous on Θ, and $g_i(\cdot, \cdot)$ are continuous on $\mathbb{R}^n \times \Theta$, then $X^*(\cdot)$ is lower semicontinuous on Θ. This is clearly the case for

$$X^*(\theta) := \left\{ x \in \mathbb{R}^n \;\middle|\; Ax \le b + S\theta, \frac{1}{2} x^T H x + c^T x \le J^*(\theta) \right\}.$$

$\square$

Remark 2.1 *One (informal) way of interpreting condition $i)$ of Corollary 2.1 is that there does not exist a value for x (apart from $x = 0$) for which the objective function vanishes and at the same time some perturbation of θ will yield a positive objective function, while some other perturbation renders it negative. Looking at condition $i)$ in conjunction with the proof we see that the condition is sufficient and not necessary. A less restrictive condition is listed in the proof, however, it is much harder to check in practice as it must be checked for all optimal active sets.*

2.3 Strictly Convex pQP

In this section we emphasize that for strictly convex pQPs a unique set of polyhedra $\mathcal{R}$ can be found simply by identifying the closures of the full-dimensional critical regions. If $H > 0$ in (2.1), the problem can be recast such that only a quadratic term remains in the objective function (Bemporad, Morari, Dua, and Pistikopoulos 2002). Without loss of generality, we use the following formulation for strictly convex pQP

$$J^*(\theta) := \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T H x \quad \text{s.t.} \quad Ax \le b + S\theta. \tag{2.3}$$

The following corollary was established in (Bemporad, Morari, Dua, and Pistikopoulos 2002):

Corollary 2.2 (Corollary to Theorem 2.1) *The PWA minimizer function $x^* : \Theta \to \mathbb{R}^n$ to* (2.3) *is continuous.*

A method for computing the expression for the restriction $x^*|_{R^k}(\cdot)$ and its polyhedral domain R^k is summarized below. The KKT conditions for (2.3) are:

$$\begin{aligned} Hx + A^T\lambda = 0, &\quad \lambda \in \mathbb{R}^q, \\ \lambda_i\left(A_i x - b_i - S_i\theta\right) = 0, &\quad \forall i \in \{1,\dots,q\}, \\ Ax - b - S\theta \leq 0, & \\ \lambda_i \geq 0, &\quad \forall i \in \{1,\dots,q\}, \end{aligned}$$

where λ are the Lagrange multipliers. Assume that an index set $\mathcal{A}$ is given such that it is an optimal active set for some parameter $\theta \in \Theta$ and let $\mathcal{N} := \{1,2,\dots,q\}\backslash\mathcal{A}$. If LICQ holds for $\mathcal{A}$, then the KKT conditions can be manipulated (Bemporad, Morari, Dua, and Pistikopoulos 2002) to obtain the following two affine functions:

$$\begin{aligned} x^{\mathcal{A}}(\theta) &:= -H^{-1}A_{\mathcal{A}}^T\lambda_{\mathcal{A}}(\theta), \\ \lambda_{\mathcal{A}}(\theta) &:= -(A_{\mathcal{A}}H^{-1}A_{\mathcal{A}}^T)^{-1}(b_{\mathcal{A}} + S_{\mathcal{A}}\theta). \end{aligned}$$

If R^k is the closure of the critical region associated with $\mathcal{A}$, i.e.

$$R^k := \mathrm{cl}(\Theta^{\mathcal{A}}) = \left\{\theta \in \Theta \;\middle|\; \begin{array}{l} A_{\mathcal{N}}x^{\mathcal{A}}(\theta) \leq b_{\mathcal{N}} + S_{\mathcal{N}}\theta \\ \lambda_{\mathcal{A}}(\theta) \geq 0 \end{array}\right\},$$

then the restriction of the minimizer function $x^*(\cdot)$ to the polyhedron R^k is given by $x^*|_{R^k}(\theta) = x^{\mathcal{A}}(\theta)$. However, if LICQ is violated for $\mathcal{A}$, the normal cone optimality condition can be used as proposed in (Mayne and Raković 2003); the affine function $x^{\mathcal{A}}(\cdot)$ is defined as the (unique) solution to

$$\begin{bmatrix} L_E^{\mathcal{A}}H \\ A_{\mathcal{A}} \end{bmatrix} x = \begin{bmatrix} 0 \\ b_{\mathcal{A}} \end{bmatrix} + \begin{bmatrix} 0 \\ S_{\mathcal{A}} \end{bmatrix}\theta. \tag{2.4}$$

In (Mayne and Raković 2003) the following was established:

Theorem 2.4 (Closure of a critical region) *The minimizer function $x^{\mathcal{A}}(\cdot)$, associated with the optimal active set $\mathcal{A}$, which is defined as the solution to* (2.4)*, is optimal in the polyhedron defined by*

$$R^{\mathcal{A}} := \mathrm{cl}\left(\Theta^{\mathcal{A}}\right) = \left\{\theta \in \Theta \;\middle|\; \begin{array}{l} A_{\mathcal{N}}x^{\mathcal{A}}(\theta) \leq b_{\mathcal{N}} + S_{\mathcal{N}}\theta \\ L_I^{\mathcal{A}}Hx^{\mathcal{A}}(\theta) \geq 0 \end{array}\right\}, \tag{2.5}$$

where $L_I^{\mathcal{A}}$ is the inequality part of a representation of the normal cone defined by the optimal active set $\mathcal{A}$ and $\mathcal{N} := \{1,\dots,q\}\backslash\mathcal{A}$.

PROOF: *See (Mayne and Raković 2003).*

□

Thus, where it is clear from the context, R^k will refer to the k^{th} set in $\mathcal{R}$ and $R^{\mathcal{A}}$ will refer to the set in $\mathcal{R}$ associated with the optimal active set $\mathcal{A}$.

Remark 2.2 *Note that since the set $R^{\mathcal{A}}$ is constructed using optimality conditions, we do not treat the cases where $\mathcal{A}^*(\theta) = \emptyset$ or $\mathcal{A}^*(\theta) = \{1,\dots,q\}$ explicitly. It should be apparent that we do not need to demand the solution feasible if $\mathcal{A}^*(\theta) = \{1,\dots,q\}$, and there is no equation $A_{\mathcal{A}} = b_{\mathcal{A}} + S_{\mathcal{A}}\theta$ that needs to be satisfied if $\mathcal{A} = \mathcal{A}^*(\theta) = \emptyset$.*

For a strictly convex pQP the function $x^*(\cdot)$ is then defined as

$$x^*(\theta) := x^{\mathcal{A}}(\theta) \quad \text{if} \quad \theta \in R^{\mathcal{A}}, \tag{2.6}$$

and the collection $\mathcal{R}$ in Theorem 2.1 becomes

$$\mathcal{R} = \left\{ R^{\mathcal{A}} \;\middle|\; \dim\left(R^{\mathcal{A}} \cap \Theta\right) = s \text{ and } \mathcal{A} = \mathcal{A}^*(\theta) \text{ for some } \theta \in \Theta \right\}.$$

Corollary 2.3 (Uniqueness of the solution) *For a strictly convex pQP, the collection $\mathcal{R}$ obtained by defining sub-regions as in (2.5), is unique and satisfies the properties in Theorem 2.1.*

PROOF: *Uniqueness of $R^{\mathcal{A}}$ follows directly from uniqueness of the optimal active set $\mathcal{A}$, $x^{\mathcal{A}}(\theta)$, and the normal cone $N^{X(\theta)}(x^*(\theta))$ for all $\theta \in \Theta$. All $R^{\mathcal{A}} \in \mathcal{R}$ are closures of critical regions, and since critical regions do not intersect, $\operatorname{int}\left(R^i\right) \cap \operatorname{int}\left(R^j\right) = \emptyset$, $i \neq j$. Since the number of optimal active sets is finite, $\cup_{k=1}^{K} R^k = \Theta$ trivially holds.*

$\square$

2.4 Convex pQP

Consider problem (2.1). In the rest of the chapter H is only restricted to be positive semi-definite and symmetric (this includes $H = 0$). Compared to the strictly convex case, a number of difficulties arise with regards to obtaining a unique set $\mathcal{R}$ and a continuous selection function $x^*(\cdot)$. We will illustrate this with a simple example.

Example 2.1 *Consider the pLP:*

$$\min_{x \in \mathbb{R}} 0x \quad \textit{s.t.} \quad x \in X^*(\theta),$$

$$X^*(\theta) := \left\{ x \in \mathbb{R} \;\middle|\; \begin{array}{rcl} -x & \leq & \sqrt{2} + \theta_2 \\ x & \leq & 12 - \sqrt{2}\theta_1 + \theta_2 \\ x & \leq & 2 + \sqrt{2}\theta_1 + \theta_2 \\ -x & \leq & 2 - \sqrt{2}\theta_1 - \theta_2 \\ -x & \leq & -8 + \sqrt{2}\theta_1 - \theta_2 \\ x & \leq & \sqrt{2} - \theta_2 \end{array} \right\}, \theta \in \mathbb{R}^2.$$

The only optimal active set yielding a full dimensional critical region is $\mathcal{A} = \emptyset$, and hence, $X(\theta) = X^*(\theta)$ for all $\theta \in \Theta$. Moreover, we can redefine $\Theta \leftarrow \Theta^{\emptyset}$. Figure 2.1(a) shows the solution set $X^*(\theta)$ and the critical region $\Theta^{\emptyset}$. If we try to define one affine function over $\Theta^{\emptyset}$, then the plane is specified by the 6 vertices of $X^*(\theta)$ corresponding to the vertices of $\Theta^{\emptyset}$, see Figure 2.1(a). This gives an inconsistent system of 6 equations with 3 unknowns. In Figure 2.1(b) four affine functions have been arbitrarily chosen, namely the basic solutions corresponding to the active sets $\mathcal{A} = \{1\}, \mathcal{A} = \{3\}, \mathcal{A} = \{4\}$ and $\mathcal{A} = \{6\}$. The associated polyhedral domains $R^{\mathcal{A}}$ have intersecting interiors, see Figure 2.1(b). If one arbitrarily cuts the sub-regions (Borrelli, Bemporad, and Morari 2003; Tøndel, Johansen, and Bemporad 2003c) in order to ensure that they do not intersect, then the resulting piecewise affine minimizer may become discontinuous, and the set $\mathcal{R}$ is non-unique.

■

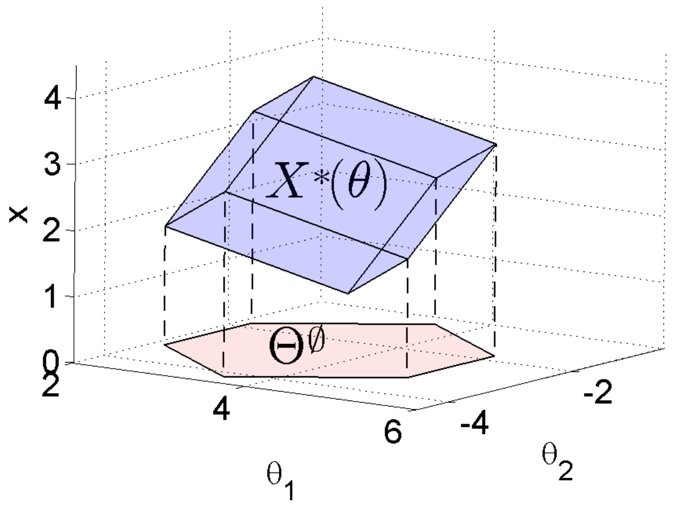
The example above has a zero cost function, however, the example is included to illustrate what happens when there are multiple optima, and a higher dimensional problem may degenerate to cases similar to that of Example 1. The example illustrates some of the following difficulties:

i) The minimizer may be non-unique, so an arbitrary selection may be discontinuous. Moreover, a continuous solution might not exist. Arbitrary selection is used in (Borrelli, Bemporad, and Morari 2003) and (Tøndel, Johansen, and Bemporad 2003c).

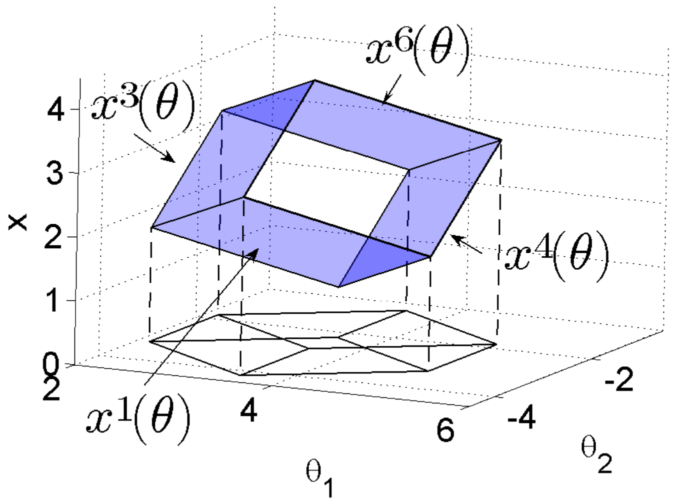
ii) If the domain $R^{\mathcal{A}}$ of the restriction $x^*|_{R^{\mathcal{A}}}(\cdot)$ is defined as the closure of a critical region, i.e. $R^{\mathcal{A}} := \mathrm{cl}\left(\Theta^{\mathcal{A}}\right)$, then $x^*|_{R^{\mathcal{A}}}(\cdot)$ may have to be piecewise affine in order to ensure that $x^*|_{R^{\mathcal{A}}}(\theta)$ is optimal for all $\theta \in R^{\mathcal{A}}$. In (Borrelli, Bemporad, and Morari 2003; Gal and Nedoma 1972; Tøndel, Johansen, and Bemporad 2003c) a piecewise affine function is defined over $\mathrm{cl}\left(\Theta^{\mathcal{A}}\right)$.

iii) When the solution to (2.1) is non-unique and an active set $\mathcal{A} \supseteq \mathcal{A}^*(\theta)$ is chosen to define the restriction $x^*|_{R^{\mathcal{A}}}(\cdot)$ and its polyhedral domain $R^{\mathcal{A}} \subseteq \mathrm{cl}\left(\Theta^{\mathcal{A}^*(\theta)}\right)$, the domain $R^{\mathcal{A}}$ may intersect with the domains of other selections for $\theta \in \mathrm{cl}\left(\Theta^{\mathcal{A}^*(\theta)}\right)$. The method of choosing an active set $\mathcal{A} \supseteq \mathcal{A}^*(\theta)$ is used in (Borrelli, Bemporad, and Morari 2003; Tøndel, Johansen, and Bemporad 2003c), but the polyhedral domains are cut in order to ensure that the sub-regions do not intersect.

The rest of this chapter is devoted to overcoming the problems described above. We will first propose a method that selects the minimizer function with the least Euclidian norm and guarantees that the set $\mathcal{R}$ is unique and satisfies the properties in Theorem 2.1. The norm of the solution vector will be minimized subject to optimality conditions of the original problem.



(a) Illustration of $X^*(\theta)$ and the critical region $\Theta^\emptyset$. $\Theta^\emptyset$ is equal to the projection of $X^*(\theta)$ onto Θ. The figure illustrates that it is impossible to define an affine function that is optimal and covers $\Theta^\emptyset$.



(b) Illustration of an arbitrary selection of minimizer functions, where the superscript denotes the active set used to compute the function. Clearly, the selections have intersecting domains.

Figure 2.1: Illustration for Example 2.1

2.4.1 Normal cone optimality condition for convex pQP

The normal cone optimality condition was defined for a general class of optimization problems in Section 2.2.3. We state the optimality condition explicitly for (2.1):

$$L_I^{\mathcal{A}}(Hx + F\theta + c) \geq 0, \tag{2.7a}$$

$$L_E^{\mathcal{A}}(Hx + F\theta + c) = 0, \tag{2.7b}$$

where the pair $(L_I^{\mathcal{A}}, L_E^{\mathcal{A}})$ is a representation of the normal cone matrix associated with the active set $\mathcal{A}$. Since the minimizer $x^* \in X^*(\theta)$ is generally non-unique for a given θ, the optimality condition (2.7) is also non-unique. The following lemma establishes that given a parameter $\theta \in \Theta$, the negative gradient of the cost function evaluated at any $x^* \in X^*(\theta)$ is contained in the normal cone associated with the *optimal active set*.

Lemma 2.1 *Consider problem* (2.1). *Let θ be given and $\mathcal{A}^*(\theta)$ be the optimal active set. For any optimal solution $x^* \in X^*(\theta)$ we have*

$$-(Hx^* + F\theta + c) \in C^{X(\theta)}(\mathcal{A}^*(\theta)). \tag{2.8}$$

PROOF: *For $\mathcal{A}(x^*, \theta) = \mathcal{A}^*(\theta)$ this is the normal cone condition for optimality. We must show that* (2.8) *holds when $\mathcal{A}(x^*, \theta) \supset \mathcal{A}^*(\theta)$. Consider an optimal solution $x^* \in X^*(\theta)$ for which $\mathcal{A}(x^*, \theta) = \mathcal{A}^*(\theta)$. Let $y^* \in X^*(\theta)$ be such that $\mathcal{A}(y^*, \theta) \supset \mathcal{A}^*(\theta)$ is the active set. If $Hy^* = Hx^*$, then the statement holds since*

$$-(Hy^* + F\theta + c) = -(Hx^* + F\theta + c) \in C^{X(\theta)}(\mathcal{A}^*(\theta)).$$

Moreover, if y and z, $y \neq z$, are optimal solutions to the convex quadratic program

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T H x + c^T x \quad \text{s.t.} \quad Ax \leq b,$$

we can express y in terms of z, that is, $y = z + \alpha d$, where α is a positive scalar and d is a direction in which the objective value remains unchanged. Since the objective function evaluated at z and y are equal and the minimizer is restricted to an affine subspace, the directional derivative of the objective function at z must be zero: $(Hz + c)^T d = 0$. Using the cost function yields

$$\frac{1}{2}(z + \alpha d)^T H (z + \alpha d) + c^T (z + \alpha d) = \frac{1}{2} z^T H z + c^T z \Rightarrow$$
$$\frac{1}{2}\alpha d^T H d + (Hz + c)^T d = 0 \Rightarrow d^T H d = 0.$$

Since $H = H^T \geq 0$ we can write $H = K^T K$ and conclude that $Kd = 0 \Rightarrow Hd = 0$. We get $Hy = H(z + \alpha d) = Hz + \alpha Hd = Hz$.

$\square$

Remark 2.3 *Note that it is proven in (Berkelaar, Roos, and Terlaky 1997) that $Hy^* = Hx^*$ for a slightly different formulation of the QP. To be sure that the result was also valid for our formulation, the result was proved here even though the formulations might be equivalent.*

Recall that the normal cone optimality condition for the convex pQP for a given θ is:

$$-\left(Hx + F\theta + c\right) \in C^{X(\theta)}(\mathcal{A}(x,\theta)),$$

for some x. By using the above lemma, it is easy to see that the optimality condition can be substituted with

$$-\left(Hx + F\theta + c\right) \in C^{X(\theta)}(\mathcal{A}^*(\theta)),$$

which yields a unique set of optimality conditions for a given θ. Henceforth we therefore only use the normal cone associated with the optimal active set.

2.4.2 Minimum norm selection for convex pQP

In this section we point out that selecting the minimum norm solution can be done by solving a strictly convex pQP over the closure of each of the critical regions for (2.1) in which the minimizer is non-unique.

Lemma 2.2 *Consider problem (2.1). For every optimal active set $\mathcal{A}$ there exists an associated optimal set mapping $\bar{X}^{\mathcal{A}} : \Theta^{\mathcal{A}} \to 2^{\mathbb{R}^n}$, which can be represented as*

$$\bar{X}^{\mathcal{A}}(\theta) := \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} L_E^{\mathcal{A}}\left(Hx + F\theta + c\right) = 0 \\ L_I^{\mathcal{A}}\left(Hx + F\theta + c\right) \geq 0 \\ A_{\mathcal{A}}x = b_{\mathcal{A}} + S_{\mathcal{A}}\theta \\ A_{\mathcal{N}}x \leq b_{\mathcal{N}} + S_{\mathcal{N}}\theta \end{array} \right\}, \tag{2.9}$$

where the pair $(L_E^{\mathcal{A}}, L_I^{\mathcal{A}})$ defines a normal cone matrix associated with the optimal active set $\mathcal{A}$, and $\mathcal{N} := \{1,\ldots,q\}\backslash\mathcal{A}$.

PROOF: *Since we have $\mathcal{A} = \mathcal{A}^*(\theta)$ for all $\theta \in \Theta^{\mathcal{A}}$, both optimality and feasibility is ensured $\forall x \in \bar{X}^{\mathcal{A}}(\theta)$ for all $\theta \in \Theta^{\mathcal{A}}$.*

□

Since the critical region $\Theta^{\mathcal{A}}$ may be an open set, we let $X^{\mathcal{A}} : \mathrm{cl}\left(\Theta^{\mathcal{A}}\right) \to 2^{\mathbb{R}^n}$ denote the extension of the mapping in (2.9) that is defined on the closure of $\Theta^{\mathcal{A}}$.

Lemma 2.3 (Uniqueness of the solution) *The minimizer function $y^* : \mathrm{cl}\left(\Theta^{\mathcal{A}}\right) \to \mathbb{R}^n$ of the following pQP*

$$\min_{y \in \mathbb{R}^n} \frac{1}{2} y^T y \quad \textit{s.t.} \quad y \in X^{\mathcal{A}}(\theta), \tag{2.10}$$

where $\theta \in \mathrm{cl}\left(\Theta^{\mathcal{A}}\right)$, is unique, continuous and piecewise affine.

PROOF: *Eliminating the equality constraints allows the problem to be written on the form* (2.3)*, which is a strictly convex pQP and Corollary 2.3 applies.*

□

Note that (2.10) can be written as

$$\min_{y\in\mathbb{R}^n} \frac{1}{2}y^T y \quad \text{s.t.} \quad \tilde{A}y \leq \tilde{b} + \tilde{S}\theta, \quad \theta \in \mathrm{cl}\left(\Theta^{\mathcal{A}}\right) \tag{2.11}$$

where the equalities have been removed with the standard procedure given in (Nocedal and Wright 1999), and $\tilde{A} \in \mathbb{R}^{t\times n}$, $\tilde{b} \in \mathbb{R}^{t\times 1}$, and $\tilde{S} \in \mathbb{R}^{t\times s}$. The pQP (2.11) is a function of the optimal active set $\mathcal{A}$ for (2.1), consequently, both the optimizer $y^*(\theta)$ and the optimal active set for (2.11) are also functions of $\mathcal{A}$, however, for notational simplicity, we let $\mathcal{B}^*(\theta)$ denote the optimal active set for (2.11).

Lemma 2.4 *Given two optimal active sets, $\mathcal{A}$ for* (2.1) *and $\mathcal{B}$ for* (2.11)*, and let $\tilde{L}_E^{\mathcal{B}}$ ($\tilde{L}_I^{\mathcal{B}}$) be the equality (inequality) part of the normal cone to $\left\{y \;\middle|\; \tilde{A}y \leq \tilde{b} + \tilde{S}\theta\right\}$ defined by $\mathcal{B}$. If $X^*(\cdot)$ is continuous on Θ, then the function $y^{\mathcal{A},\mathcal{B}}(\cdot)$ that is the unique solution to*

$$\begin{bmatrix} \tilde{L}_E^{\mathcal{B}} \\ \tilde{A}_{\mathcal{B}} \end{bmatrix} y = \begin{bmatrix} 0 \\ \tilde{b}_{\mathcal{B}} \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{S}_{\mathcal{B}} \end{bmatrix} \theta, \tag{2.12}$$

is optimal for (2.1) *and* (2.11) *when restricted to*

$$\begin{aligned} R^{\mathcal{A},\mathcal{B}} &:= \mathrm{cl}\left(\{\theta \in \Theta \mid \mathcal{A} = \mathcal{A}^*(\theta), \mathcal{B} = \mathcal{B}^*(\theta)\}\right) \\ &= \left\{\theta \in \Theta \;\middle|\; \begin{array}{l} \tilde{A}_{\mathcal{N}} y^{\mathcal{A},\mathcal{B}}(\theta) \leq \tilde{b}_{\mathcal{N}} + \tilde{S}_{\mathcal{N}}\theta \\ \tilde{L}_I^{\mathcal{B}} y^{\mathcal{A},\mathcal{B}}(\theta) \geq 0 \end{array} \right\}, \end{aligned} \tag{2.13}$$

where $\mathcal{N} := \{1,\ldots,t\}\backslash\mathcal{B}$.

PROOF: *This is fulfilled by construction $\forall\theta \in R^{\mathcal{A},\mathcal{B}} \cap \Theta^{\mathcal{A}}$ even if $X^*(\cdot)$ is not continuous. Since the optimal set mapping $X^*(\cdot)$ and the function $y^*(\cdot)$ are continuous, $y^*(\theta) \in X^*(\theta)$ for all $\theta \in \mathrm{bd}\left(R^{\mathcal{A},\mathcal{B}} \cap \mathrm{cl}\left(\Theta^{\mathcal{A}}\right)\right)$ and the statement of the lemma follows.*

□

To be able to use the proposed procedure when the continuity of $X^*(\cdot)$ is not guaranteed, we must show that $y^*(\theta) \in X^*(\theta)$ for all $\theta \in \mathrm{cl}\left(\Theta^{\mathcal{A}}\right)$, but that $y^*(\cdot)$ is not necessarily the solution to (2.1) with least Euclidean norm when evaluated on the boundary of $\mathrm{cl}\left(\Theta^{\mathcal{A}}\right)$.

Lemma 2.5 *The following holds: $y^*(\theta) \in X^*(\theta)$ for all $\theta \in \mathrm{cl}\left(\Theta^{\mathcal{A}}\right)$, where $\mathcal{A}$ is an optimal active set.*

PROOF: *The lemma obviously holds for $\forall\theta \in \Theta^{\mathcal{A}}$. If $\mathcal{N} = \{1,\ldots,q\}\backslash\mathcal{A}$, we have*

$$\{\theta \in \Theta \mid \exists x : A_{\mathcal{A}}x = b_{\mathcal{A}} + S_{\mathcal{A}}\theta, A_{\mathcal{N}}x \leq b_{\mathcal{N}} + S_{\mathcal{N}}\theta\} \supset \mathrm{cl}\left(\Theta^{\mathcal{A}}\right),$$

and hence, $y^*(\theta)$ is feasible for $\forall\theta \in \mathrm{bd}\left(\mathrm{cl}\left(\Theta^{\mathcal{A}}\right)\right)$. Optimality of $y^*(\theta)$ for all $\theta \in \mathrm{bd}\left(\mathrm{cl}\left(\Theta^{\mathcal{A}}\right)\right)$ follows directly from the continuity of $y^*(\cdot)$ and value function $J^*(\cdot)$ for (2.1).

□

We can now define the mapping $x^* : \Theta \to \mathbb{R}^n$ in Theorem 2.1 as:

$$x^*(\theta) = y^{\mathcal{A},\mathcal{B}}(\theta) \quad \text{if} \quad \theta \in R^{\mathcal{A},\mathcal{B}} \tag{2.14}$$

and the set of polyhedra on which x^* is defined is given by

$$\mathcal{R} = \left\{ R^{\mathcal{A},\mathcal{B}} \,\middle|\, \dim\left(\Theta \cap R^{\mathcal{A},\mathcal{B}}\right) = s \right\}. \tag{2.15}$$

In the following theorem the statements are fulfilled by construction and is therefore presented without proof:

Theorem 2.5 (Solution properties) *Consider* (2.1).

i) The set $\mathcal{R}$ defined in (2.15) *is unique and satisfies the properties in Theorem 2.1.*

ii) The function $x^* : \Theta \to \mathbb{R}$ defined by (2.14) *is continuous and the solution to* (2.1) *with the least Euclidean norm if the optimal set mapping $X^*(\cdot)$ is continuous on Θ.*

Remark 2.4 *Note that if $X^*(\cdot)$ is not continuous on Θ, the existence of a continuous solution is not guaranteed and $x^*(\cdot)$ may not be the minimum norm selection for all $\theta \in \Theta$ since* (2.11) *is not solved for optimal active sets $\mathcal{A}$ whose corresponding critical regions $\Theta^{\mathcal{A}}$ are lower-dimensional. Only if no continuous solutions exists when passing from one sub-region to another, and only on the boundary between these regions, may $x^*(\cdot)$ in* (2.14) *be different from the minimum norm solution and/or set-valued. Thus, an ordering of the sets in $\mathcal{R}$ is necessary for $x^*(\theta)$ to be uniquely defined when θ lies on the boundary of a sub-region.*

We revisit our example to illustrate the proposed method.

Example 2.2 (Example 2.1 continued) *Consider Example 1. The minimum norm problem is*

$$\min_x \frac{1}{2}x^2 \quad \textit{s.t.} \quad x \in X^*(\theta), \quad \theta \in \mathrm{cl}\left(\Theta^{\emptyset}\right) \tag{2.16}$$

The solution consist of 3 restrictions and their polyhedral domains, see Figure 2.2.

■

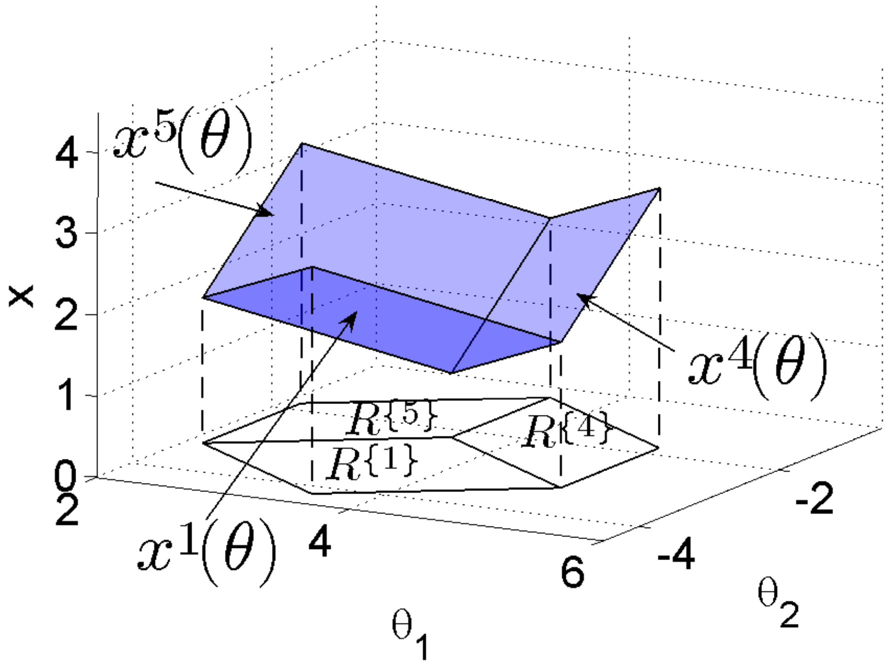


Figure 2.2: Minimum norm selection

Example 2.3 *The following example illustrates the proposed minimum norm method. Consider the following convex pQP:*

$$J^*(\theta) := \min_{\{x \in X(\theta) \,|\, \theta \in \Theta\}} \left\{ \frac{1}{2} x_4^2 - x_1 - x_2 - x_3 + 0.5 x_5 \right\},$$

$$X(\theta) := \left\{ x \in \mathbb{R}^5 \;\middle|\; \begin{array}{rcl} x_1 + x_2 + x_3 & \leq & 10 - \theta_1 - \theta_2 \\ x_1 - 2x_2 & \leq & 4 - \theta_1 - 2\theta_2 \\ -x_1 - 2x_3 & \leq & 3 - \theta_1 - 2\theta_2 \\ -x_3 + x_5 & \leq & 2 + \theta_2 \\ -x_5 & \leq & 2 + \theta_1 \\ x_4 - x_5 & \leq & -\theta_1 \\ -3 \leq & x_1 & \leq 3 \\ -3 \leq & x_2 & \leq 3 \\ -3 \leq & x_3 & \leq 3 \end{array} \right\},$$

$$\Theta := \left\{ \theta \in \mathbb{R}^2 \;\middle|\; \begin{array}{rcl} -1 \leq & \theta_1 & \leq 2.5 \\ 0 \leq & \theta_2 & \leq 3 \end{array} \right\}.$$

Since $F = 0$, $X^*(\cdot)$ is continuous on Θ. The unique $\mathcal{R}$ obtained by the proposed method is depicted in Figure 2.3(a), where the superscripts denote which constraints in (2.1) that are active at the optimum in (2.11)*. The constraints are indexed in the order in which they are listed, e.g. $-3 \leq x_1$ is constraint number 7 and $x_1 \leq 3$ is number 8. The optimizer is non-unique in several of the regions* *in Figure 2.3(a), and by following the procedure in (Tøndel, Johansen, and Bemporad 2003c) there are several possible solutions, one of which is depicted in Figure 2.3(b), where the superscripts denote which constraints have been arbitrarily selected to construct the regions. Not only is the solution non-unique, but the optimizer is also discontinuous for some selections, see Figure 2.3(d).*

■

Example 2.4 *By adding $\theta_1 x_5$ to the objective function in Example 2.3, neither point $i)$ nor $ii)$ in Corollary 2.1 hold, and consequently $X^*(\cdot)$ may not be continuous. By following the proposed method, a unique solution is obtained, however, a continuous solution does not exist, see Figures 2.4(a)-2.4(b).*

■

It should be noted that even though a continuous solution may not exist for (2.1), condition $ii)$ in Corollary 2.1 shows that the minimum norm method always finds a continuous solution for LP-based model predictive control problems.

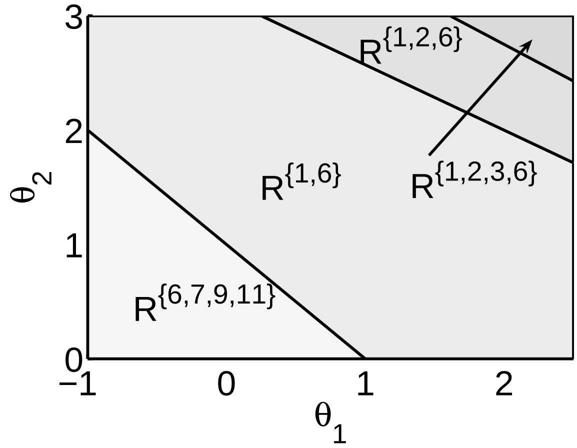
A natural question with regards to the proposed minimum norm method is whether one can simply add $\frac{1}{2}\epsilon x^T x$, where ϵ is a sufficiently small scalar, to the objective function in (2.1) and obtain the same solution? Clearly, this approach is not exact and the quadratic term will dominate the linear parts sufficiently far from the origin. Choosing the value of ϵ is not trivial; too small a value will result in a poorly conditioned Hessian and too large a value will result in a solution that is too far from optimal. The proposed minimum norm method can therefore be argued to be more numerically robust compared to adding a quadratic term.

2.4.3 Parametric linear programs

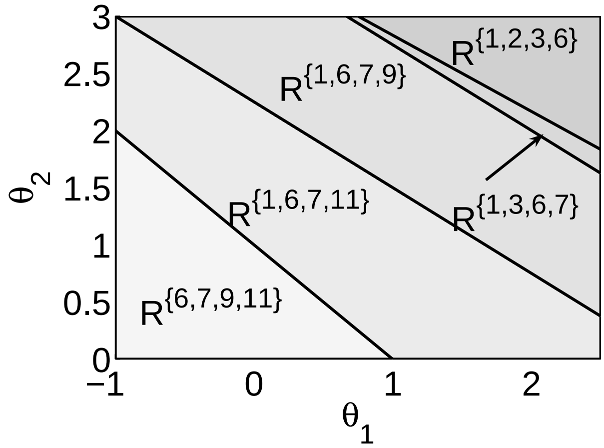
Although the results from the preceding sections are valid for pLPs, some of the results follow easier in the linear case. In particular, if $f(x,\theta) := c^T x$, then selecting the optimizer with the least Euclidian norm can be done by solving the following strictly convex pQP

$$V^*(\theta) := \min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}x^T x \;\middle|\; Ax \le b + S\theta, c^T x = J^*|_{\mathrm{cl}(\Theta^{\mathcal{A}})}(\theta) \right\}, \theta \in \mathrm{cl}\left(\Theta^{\mathcal{A}}\right), \tag{2.17}$$

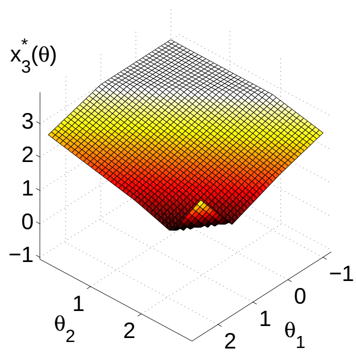
since J^* is piecewise affine in the linear case. One can avoid computing the normal cone by using the KKT-conditions to characterize the sub-regions (Bemporad, Morari, Dua, and Pistikopoulos 2002). However, since the analytical expressions for $x^{\mathcal{A}}$ and $R^{\mathcal{A}}$ obtained by manipulating the KKT-conditions rely on the LICQ assumption, we replace (2.17) with another pQP for which LICQ is not always violated.



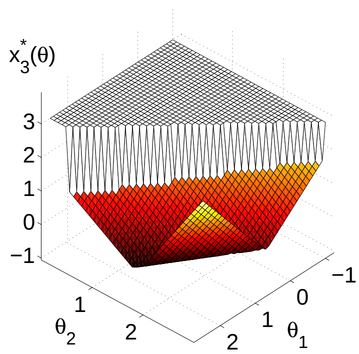
(a) The set $\mathcal{R}$, proposed method.



(b) A set $\mathcal{R}$, arbitrary selection.

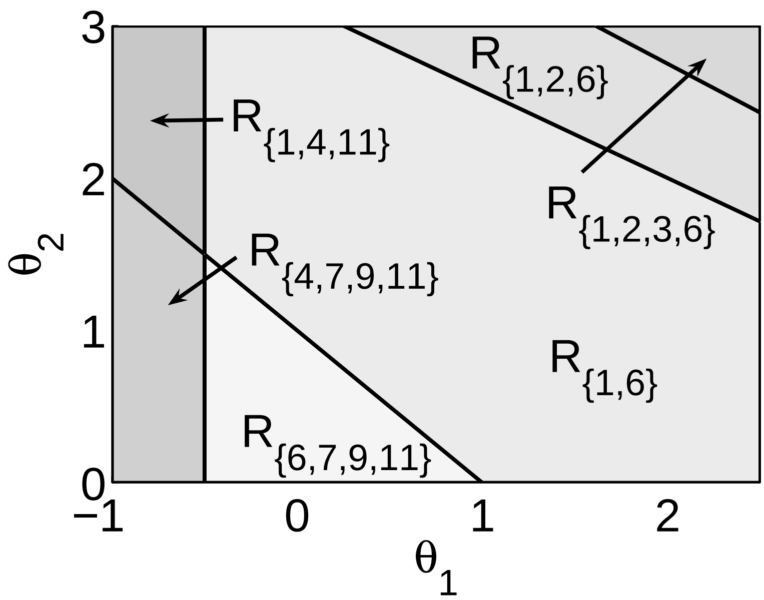


(c) $x_3^*(\cdot)$ continuous, proposed method.

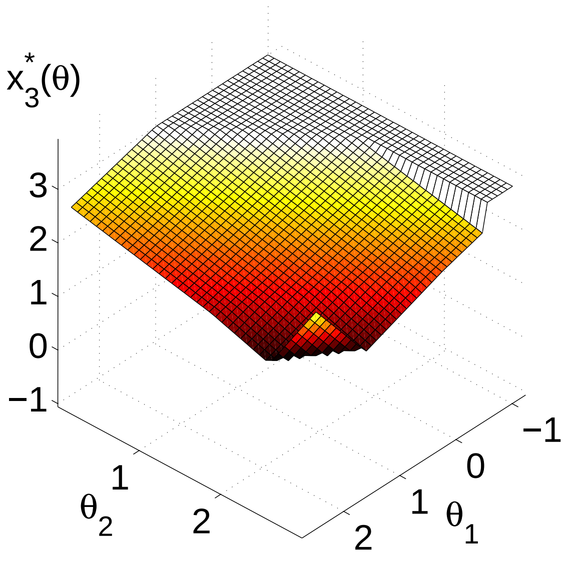


(d) $x_3^*(\cdot)$ discontinuous, arbitrary selection.

Figure 2.3: Illustration of the proposed method and arbitrary selection .



(a) The set $\mathcal{R}$ with $F \neq 0$, proposed method.



(b) $x_3^*(\cdot)$ discontinuous, proposed method.

Figure 2.4: Illustration of discontinuous solutions.

Lemma 2.6 *LICQ is violated for all optimal active sets for* (2.17). *Moreover, the* *following pQP is equivalent to* (2.17)*:*

$$V^*(\theta) := \min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}x^T x \;\middle|\; A_{\mathcal{A}}x = b_{\mathcal{A}} + S_{\mathcal{A}}\theta, A_{\mathcal{N}}x \le b_{\mathcal{N}} + S_{\mathcal{N}}\theta \right\}, \theta \in \mathrm{cl}\left(\Theta^{\mathcal{A}}\right), \tag{2.18}$$

where $\mathcal{A}$ is an optimal active set and $\mathcal{N} := \{1, \dots, q\}\backslash\mathcal{A}$.

PROOF: *First it is shown that the constraints in* (2.17) *and* (2.18) *define the same set. This holds trivially if $c = 0$. Let $c \neq 0$. It is clear that $\mathcal{F} := X(\theta) \cap \left\{x \;\middle|\; c^T x = J^*(\theta)\right\}$ is a face of $X(\theta)$. The constraints fulfilled with equality $\forall x \in \mathcal{F}$ are exactly the constraints whose indices are in the optimal active set $\mathcal{A}$. From (Jones, Kerrigan, and Maciejowski 2004, Definition 8 and Theorem 2.12) we have that it is a one to one mapping from these constraints to the faces of $X(\theta)$, and that $\mathcal{F} = \{x \mid A_{\mathcal{A}}x = b_{\mathcal{A}} + S_{\mathcal{A}}\theta\} \cap X(\theta)$. Since the sets defined by the constraints in* (2.17) *and* (2.18) *are equal, the LICQ assertion holds trivially.*

□

Note that LICQ can be violated for the optimal active set in (2.18), but one can choose a subset of the active constraints to characterize a sub-region, as explained in (Bemporad, Morari, Dua, and Pistikopoulos 2002). Following the approach in section 4.1.1 in (Bemporad, Morari, Dua, and Pistikopoulos 2002), the resulting collection of polyhedra $\mathcal{R}$ in Theorem 2.1 does not generally satisfy the property $\mathrm{int}\left(R^i\right) \cap \mathrm{int}\left(R^j\right) \neq \emptyset$, $i \neq j$, however, the following holds; for every pair $(R^i, R^j) \in \mathcal{R} \times \mathcal{R}$

$$\dim\left(R^j \cap R^i\right) = s \Rightarrow x^*|_{R^j}(\theta) = x^*|_{R^i}(\theta).$$

Thus, even if the set $\mathcal{R}$ in Theorem 2.1 is non-unique, x^* is single-valued and continuous, and each sub-region $R^k \in \mathcal{R}$ can be found without constructing the normal cone.

2.5 Remarks on exploration strategies and related work

2.5.1 Exploration strategies

With small modifications the exploration strategies presented in (Bemporad, Morari, Dua, and Pistikopoulos 2002; Borrelli, Bemporad, and Morari 2003; Tøndel, Johansen, and Bemporad 2003a; Tøndel, Johansen, and Bemporad 2003c; Baotić 2002; Grieder, Borrelli, Torrisi, and Morari 2004) can be used to explore the parameter space. The simplest approach is to modify the algorithms to only identify critical regions, and then solve the minimum norm problem over the critical regions in which the optimizer is non-unique. A more efficient approach would be to use utilize the uniqueness of the sub-regions to avoid explicitly computing the critical regions. Another important issue that has recently been pointed out (Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006a), is that for pQPs, even those with $H > 0$, a critical region may have more than one adjacent critical region along a given facet, and hence rendering some of the mentioned algorithms without guarantees that the entire parameter space will be explored. Thus, we recommend that the exploration strategy in (Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006a) is utilized.

2.5.2 Related work

An algorithm that obtains a continuous minimizer function for single-parametric LPs is presented in (Zhang and Liu 1990), while (Böhm 1975) indicates how to construct a continuous solution for pLPs on the form

$$\min \left\{ c^T x \mid Ax = b + S\theta, x \geq 0 \right\} .$$

The latter, however, is not a complete algorithm for obtaining solutions to pLPs. No region of optimality is constructed for the associated element of the continuous selection and difficulties that arise when both the primal and dual solutions of the pLP are non-unique, are not discussed.

Only recently a new approach has been proposed (Jones, Kerrigan, and Maciejowski 2007) for the purpose of obtaining unique and continuous solutions to pLPs on the form $\min \left\{ c^T x \mid Ax \leq b + S\theta \right\}$. The algorithm theoretically perturbs the problem in order to ensure that the primal and dual solutions are unique for all parameters. The main advantages of this approach is that a pQP solver is not needed and that the solution satisfies the facet-to-facet (only one adjacent critical region for each facet) property (Spjøtvold, Kerrigan, Jones, Tøndel, and Johansen 2006a), which allows for the point location problem to be solved in logarithmic time (Jones, Grieder, and Raković 2006). The main drawback is that the epigraph of the value function must be a polyhedron, and hence, the algorithm cannot handle pLPs with parameters in the cost and the constraints.

2.6 Concluding Remarks

We have showed that using the normal cone optimality condition to construct critical regions (Mayne and Raković 2003) yields a PWA function defined on a unique set of polyhedra for strictly convex pQPs. For convex pQPs uniqueness of the solution is ensured by choosing the minimum norm optimizer and using the normal cone to characterize the parametric region in which the restriction remains optimal. Continuity of the piecewise affine minimizer function is guaranteed if the optimal set mapping is continuous on the set of admissible parameters.

Chapter 3

The Facet-to-Facet Property of Solutions to Convex Parametric Quadratic Programs

The algorithms proposed in (Bemporad, Morari, Dua, and Pistikopoulos 2002) and (Borrelli, Bemporad, and Morari 2003) introduce artificial cuts in the parameter space in the search for the solution, while in (Seron, Goodwin, and Doná 2003) an algorithm based on considering all combinations of constraints is presented. In (Baotić 2002) and (Grieder, Borrelli, Torrisi, and Morari 2004) the authors propose a method for exploring the parameter space, which is conceptually and computationally more efficient than in (Bemporad, Morari, Dua, and Pistikopoulos 2002), (Borrelli, Bemporad, and Morari 2003) and (Seron, Goodwin, and Doná 2003); by stepping a sufficiently small distance over the boundary of a so-called critical region[2] and solving an LP or QP for the resulting parameter, a new critical region is defined. This procedure looks promising, but implicitly relies on the assumption that the facets of the closures of adjacent critical regions satisfy a certain property, namely that their intersection is a facet of both regions. We will refer to this as the facet-to-facet property. This property is not satisfied for pQPs that are convex (as opposed to strictly convex), see (Spjøtvold 2005) for an example. It is considerably more difficult to conclude whether or not the property holds for strictly convex pQPs.

In (Tøndel, Johansen, and Bemporad 2003a) and (Tøndel, Johansen, and Bemporad 2003c) the authors propose a method in which each facet of the critical region is examined and, depending on whether the facet ensures feasibility or optimality, the active set in the neighboring critical region is found by adding or removing a constraint from the current active set. The examination of each facet relies on a number of non-degeneracy assumptions and in cases where they are not satisfied, the algorithm assumes that the facet-to-facet property holds when stepping a small

[2] A critical region is defined as the set of parameters for which some fixed set of constraints are fulfilled with equality at all solutions of an optimization problem.

distance over a facet to determine the active set in the adjacent region. The algorithms presented in (Baotić 2002), (Bemporad, Morari, Dua, and Pistikopoulos 2002), (Grieder, Borrelli, Torrisi, and Morari 2004), (Seron, Goodwin, and Doná 2003) and (Tøndel, Johansen, and Bemporad 2003a) are applied to strictly convex pQPs and utilized to obtain explicit solutions to model predictive control problems. We show by an example that for the class of convex pQPs a critical region may have more than one adjacent critical region for each facet. Consequently, the facet-to-facet property does not generally hold. A simple modification of the algorithm in (Tøndel, Johansen, and Bemporad 2003a), based on results from (Bemporad, Morari, Dua, and Pistikopoulos 2002), that does not rely on the facet-to-facet property, is presented. Finally, numerical results indicate that the proposed method has a lower computational complexity than the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) for pQPs whose solution contains a large number of critical regions.

3.1 Preliminaries

If A is a matrix or column vector, then A_i denotes the i^{th} row of A and $A_{\mathcal{I}}$ denotes the sub-matrix of the rows of A corresponding to the index set $\mathcal{I}$. Recall that the set of affine combinations of points in a set $S \subset \mathbb{R}^n$ is called the *affine hull* of S, and is denoted $\operatorname{aff}(S)$. The *dimension of a set* $S \subset \mathbb{R}^n$ is the dimension of $\operatorname{aff}(S)$, and is denoted $\dim(S)$; if $\dim(S) = n$, then S is said to be full-dimensional. The *closure* and *interior* of a set S is denoted $\operatorname{cl}(S)$ and $\operatorname{int}(S)$, respectively. The *relative interior* of a set S is the interior relative to $\operatorname{aff}(S)$, i.e. $\operatorname{relint}(S) := \{x \in S \mid B(x,r) \cap \operatorname{aff}(S) \subseteq S \text{ for some } r > 0\}$, where the ball $B(x,r) := \{y \mid \|y - x\| \leq r\}$ and $\|\cdot\|$ is any norm. A *polyhedron* is the intersection of a finite number of closed halfspaces. A non-empty set F is a *face* of the polyhedron $P \subset \mathbb{R}^n$ if there exists a hyperplane $\{z \in \mathbb{R}^n \mid a^T z = b\}$, where $a \in \mathbb{R}^n$, $b \in \mathbb{R}$, such that $F = P \cap \{z \in \mathbb{R}^n \mid a^T z = b\}$ and $a^T z \leq b$ for all $z \in P$. Given an s-dimensional polyhedron $P \subset \mathbb{R}^n$, where $s \leq n$, the *facets* of P are the $(s-1)$-dimensional faces of P.

Consider the following strictly convex parametric quadratic program:

$$V^*(\theta) := \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} x^T H x \;\middle|\; Ax \leq b + S\theta \right\}, \tag{3.1}$$

where $\theta \in \mathbb{R}^s$ is the *parameter* of the optimization problem, and the vector $x \in \mathbb{R}^n$ is to be optimized for all values of $\theta \in \Theta$, where $\Theta \subseteq \mathbb{R}^s$ is some polyhedral set. Moreover, $H = H^T \in \mathbb{R}^{n \times n}$, $H > 0$, $A \in \mathbb{R}^{q \times n}$, $b \in \mathbb{R}^{q \times 1}$, and $S \in \mathbb{R}^{q \times s}$. For a given parameter θ, the minimizer to (3.1) is denoted by $x^*(\theta)$. Without loss of generality, the following standing assumption is made (Bemporad, Morari, Dua, and Pistikopoulos 2002; Borrelli, Bemporad, and Morari 2003):

Assumption 3.1 *The set of admissible parameters Θ is full-dimensional, and for* *all $\theta \in \Theta$, the set of feasible points $X(\theta) := \{x \in \mathbb{R}^n \mid Ax \le b + S\theta\}$ is non-empty.*

For convenience we restate some of the definitions and results from the previous chapter.

Definition 3.1 (Optimal active set) *Let x be a feasible point of (3.1) for a given θ. The active constraints are the constraints that fulfill $A_i x - b_i - S_i\theta = 0$. The indices of the constraints that are active at the solution $x^*(\theta)$ is referred to as the* optimal active set *and it is denoted by $\mathcal{A}^*(\theta)$, i.e.*

$$\mathcal{A}^*(\theta) := \{i \in \{1, 2, \dots, q\} \mid A_i x^*(\theta) - b_i - S_i\theta = 0\}.$$

Definition 3.2 (Critical region) *Given an index set $\mathcal{A} \subseteq \{1, 2, \dots, q\}$, the critical region $\Theta_{\mathcal{A}}$ associated with $\mathcal{A}$ is the non-empty set of parameters for which the optimal active set is equal to $\mathcal{A}$, i.e.*

$$\Theta_{\mathcal{A}} := \{\theta \in \Theta \mid \mathcal{A}^*(\theta) = \mathcal{A}\}.$$

Definition 3.3 (LICQ) *For a non-empty index set $\mathcal{A} \subseteq \{1, 2, \dots, q\}$, we say that the linear independence constraint qualification (LICQ) holds for $\mathcal{A}$ if the gradients of the set of constraints indexed by $\mathcal{A}$ are linearly independent, i.e. $A_{\mathcal{A}}$ has full row rank.*

Theorem 3.1 ((Bemporad, Morari, Dua, and Pistikopoulos 2002)) *Consider the pQP in* (3.1). *The value function $V^* : \Theta \to \mathbb{R}$ is convex and continuous. The minimizer function $x^* : \Theta \to \mathbb{R}^n$ is continuous and piecewise affine in the sense that there exists a finite set of full-dimensional polyhedra $\mathcal{R} := \{R_1, \dots, R_K\}$ such that $\Theta = \cup_{k=1}^{K} R_k$, $\operatorname{int}(R_i) \cap \operatorname{int}(R_j) = \emptyset$ for all $i \neq j$ and the restriction $x^*|_{R_k} : R_k \to \mathbb{R}^n$ is affine for all $k \in \{1, \dots, K\}$.*

A method for computing the expression for the restriction (affine function) $x^*|_{R_k}$ and its polyhedral domain R_k is summarized below. The KKT conditions for (3.1) are:

$$\begin{aligned} Hx + A^T\lambda &= 0, \quad && \lambda \in \mathbb{R}^q, \\ \lambda_i \left(A_i x - b_i - S_i\theta\right) &= 0, && \forall i \in \{1, \dots, q\}, \\ Ax - b - S\theta &\le 0, && \\ \lambda_i &\ge 0, && \forall i \in \{1, \dots, q\}, \end{aligned}$$

where λ are the Lagrange multipliers. Assume that an index set $\mathcal{A}$ is given such that it is an optimal active set for some parameter $\theta \in \Theta$ and let $\mathcal{N} := \{1, 2, \dots, q\} \backslash \mathcal{A}$. If LICQ holds for $\mathcal{A}$, then the KKT conditions can be manipulated (Bemporad, Morari, Dua, and Pistikopoulos 2002) to obtain the following two affine functions:

$$\begin{aligned} x^*_{\mathcal{A}}(\theta) &:= -H^{-1}A^T_{\mathcal{A}}\lambda^*_{\mathcal{A}}(\theta), \\ \lambda^*_{\mathcal{A}}(\theta) &:= -(A_{\mathcal{A}}H^{-1}A^T_{\mathcal{A}})^{-1}(b_{\mathcal{A}} + S_{\mathcal{A}}\theta). \end{aligned}$$

If R_k is the closure of the critical region associated with $\mathcal{A}$:

$$R_k := \mathrm{cl}(\Theta_{\mathcal{A}}) = \left\{ \theta \in \Theta \;\middle|\; \begin{array}{l} A_{\mathcal{N}} x^*_{\mathcal{A}}(\theta) \leq b_{\mathcal{N}} + S_{\mathcal{N}}\theta \\ \lambda^*_{\mathcal{A}}(\theta) \geq 0 \end{array} \right\} \tag{3.2}$$

then the restriction of the minimizer function x^* to the polyhedron R_k is given by $x^*|_{R_k}(\theta) = x^*_{\mathcal{A}}(\theta)$. If LICQ does not hold, then closure of a critical region associated with an optimal active set can be found by projecting a polyhedron in the (x, λ)-space onto the parameter space (Bemporad, Morari, Dua, and Pistikopoulos 2002; Tøndel, Johansen, and Bemporad 2003c) or by utilizing the normal cone optimality condition as described in (Mayne and Raković 2002) and in Chapter 2.

In the sequel, the closure of a critical region will be written in the more compact form

$$\mathrm{cl}(\Theta_{\mathcal{A}}) =: \{\theta \in \Theta \mid C_i\theta \leq d_i,\ i = 1, \ldots, J\},$$

which is obtained from (3.2), by projection or from utilizing the normal cone optimality condition. An inequality $C_i\theta \leq d_i$ in the description of $\mathrm{cl}(\Theta_{\mathcal{A}})$ is said to be *facet-defining* if $\{\theta \mid C_i\theta = d_i\}$ equals the affine hull of one of the facets of $\mathrm{cl}(\Theta_{\mathcal{A}})$. If there exists more than one facet-defining inequality for a given facet, these inequalities are referred to as *coinciding inequalities*. A representation of $\mathrm{cl}(\Theta_{\mathcal{A}})$ where every redundant inequality has been removed is referred to as an *irredundant* representation (note that an irredundant representation does not have any coinciding inequalities).

3.2 Algorithms for exploring the parameter space

The goal of most algorithms for solving pQPs is to identify only the closures of the full-dimensional critical regions (Baotić 2002; Bemporad, Morari, Dua, and Pistikopoulos 2002; Borrelli, Bemporad, and Morari 2003; Grieder, Borrelli, Torrisi, and Morari 2004; Tøndel, Johansen, and Bemporad 2003a; Tøndel, Johansen, and Bemporad 2003c). For this purpose we introduce the notion of *adjacent critical regions*.

Definition 3.4 (Adjacent critical regions) *Two full-dimensional critical regions* $\Theta_{\mathcal{A}} \subset \mathbb{R}^s$ *and* $\Theta_{\mathcal{B}} \subset \mathbb{R}^s$ *are said to be adjacent if* $\dim(\mathrm{cl}(\Theta_{\mathcal{A}}) \cap \mathrm{cl}(\Theta_{\mathcal{B}})) = s - 1$.

The framework for studying the various algorithms is given in Algorithm 3.1, where the auxiliary set $\mathcal{U}$ is defined as the set of closures of identified regions whose adjacent regions have not been found. The output of Algorithm 3.1 is a collection $\mathcal{R}$ of closures of full-dimensional critical regions for (3.1). From this point on, we will let K denote the number of sets in $\mathcal{R}$. Where it is clear from the context, R_k will refer to the k^{th} set in $\mathcal{R}$ and $R_{\mathcal{A}}$ will refer to the set in $\mathcal{R}$ associated with the optimal active set $\mathcal{A}$.

Algorithm 3.1 Exploring the parameter space.

Input: Data to problem (3.1).
Output: Set of closures of full-dimensional critical regions $\mathcal{R}$.
1: Find a $\theta \in \Theta$ such that $\dim\left(\mathrm{cl}\left(\Theta_{\mathcal{A}^*(\theta)}\right)\right) = s$.
2: $\mathcal{R} \leftarrow \{\mathrm{cl}\left(\Theta_{\mathcal{A}^*(\theta)}\right)\}$ and $\mathcal{U} \leftarrow \{\mathrm{cl}\left(\Theta_{\mathcal{A}^*(\theta)}\right)\}$.
3: **while** $\mathcal{U} \neq \emptyset$ **do**
4: Choose any element $U \in \mathcal{U}$.
5: $\mathcal{U} \leftarrow \mathcal{U} \backslash \{U\}$.
6: **for** all facets f of U **do**
7: Find the set $\mathcal{S}$ of full-dimensional critical regions adjacent to U along the facet f.
8: $\mathcal{U} \leftarrow \mathcal{U} \cup (\mathcal{S} \backslash \mathcal{R})$.
9: $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{S}$.
10: **end for**
11: **end while**

We will consider the algorithms in (Tøndel, Johansen, and Bemporad 2003a), (Baotić 2002), (Grieder, Borrelli, Torrisi, and Morari 2004) and (Tøndel, Johansen, and Bemporad 2003c). It should be noted that, on a conceptual level, these algorithms differ only in step 7 in Algorithm 3.1 and that the different strategies may not always yield a satisfactory result. This will be addressed in the rest of this section.

3.2.1 Identifying adjacent regions from a QP

The procedure used in (Baotić 2002) and (Grieder, Borrelli, Torrisi, and Morari 2004) as step 7 of Algorithm 3.1 is given in Procedure 3.1. This method is also one of the methods used in The Multi Parametric Toolbox (MPT) (Kvasnica, Grieder, and Baotić 2005). Note that at most one adjacent critical region is identified for each facet of the region under consideration. The implementation of the procedure will not be discussed.

3.2.2 Identifying adjacent regions from inequalities

Let $\mathcal{A}$ be a given optimal active set for some $\theta \in \Theta$. The objective is to identify a critical region adjacent to $\Theta_{\mathcal{A}}$ along a given facet f of its closure. Consider the following conditions (Tøndel, Johansen, and Bemporad 2003a):

1. LICQ holds for $\mathcal{A}$.

2. There are no coinciding inequalities for facet f in (3.2), where redundant constraints have not yet been removed.

3. There are no weakly active constraints at $x^*(\theta)$ for all $\theta \in \mathrm{cl}\left(\Theta_{\mathcal{A}}\right)$, that is, $\nexists\, i \in \mathcal{A} \Rightarrow \lambda_i^*(\theta) = 0, \forall \theta \in \mathrm{cl}\left(\Theta_{\mathcal{A}}\right)$.

Procedure 3.1 Finding an adjacent full-dimensional critical region along a given facet.

Input: Irredundant representation of the closure of a full-dimensional critical region $U =: \{\theta \mid C_i\theta \leq d_i,\ i = 1, \ldots, J\}$ and the index j whose corresponding inequality defines facet f.

Output: Closure of a full-dimensional critical region $\mathcal{S}$ adjacent to U along the facet f.

1: $\mathcal{S} \leftarrow \emptyset$.
2: Choose any $\hat{\theta} \in \mathrm{relint}(f)$.
3: **if** the facet f is not on the boundary of Θ **then**
4: Choose any scalar $\varepsilon > 0$ such that $\theta := \hat{\theta} + \varepsilon C_j^T \in \Theta$ and θ is in a full-dimensional critical region adjacent to U.
5: Compute $\mathcal{A}^*(\theta)$ by solving the QP (3.1).
6: $\mathcal{S} \leftarrow \left\{\mathrm{cl}\left(\Theta_{\mathcal{A}^*(\theta)}\right)\right\}$.
7: **end if**

If these conditions hold, then (Tøndel, Johansen, and Bemporad 2003a) proves that there is only one critical region adjacent to $\Theta_{\mathcal{A}}$ along facet f and that the corresponding optimal active set can be found by determining what type of inequality defines f. If the inequality that defines f is of the type $\lambda_i \geq 0$, then i is removed from $\mathcal{A}$, hence $\mathcal{S} = \left\{\mathrm{cl}\left(\Theta_{\mathcal{A}\setminus\{i\}}\right)\right\}$. On the other hand, if the inequality is of the type $A_i x^*(\theta) \leq b_i + S_i\theta$, then i is added to $\mathcal{A}$, hence $\mathcal{S} = \left\{\mathrm{cl}\left(\Theta_{\mathcal{A}\cup\{i\}}\right)\right\}$. If the conditions do not hold, then Procedure 3.1 is used. Clearly, as in Section 3.2.1, only one adjacent critical region is identified for each facet with this strategy.

3.2.3 Required solution properties

Consider now the question: What conditions must the solution to (3.1) satisfy in order to ensure that the strategies in Section 3.2.1 or 3.2.2 guarantee that $\bigcup_{k=1}^{K} R_k = \Theta$? For this purpose, we introduce the following definition:

Definition 3.5 (Facet-to-facet) *Let $\mathcal{P} := \{P_i \mid i \in \mathcal{I}\}$ be a finite collection of full-dimensional polyhedra in $\mathbb{R}^s$, where $\mathrm{int}(P_i) \cap \mathrm{int}(P_j) = \emptyset$ for all (i, j), $i \neq j$. We say that the* facet-to-facet *property holds for $\mathcal{P}$ if $F_{(i,j)} := P_i \cap P_j$ is a facet of both P_i and P_j for all $(s-1)$-dimensional intersections $F_{(i,j)}$.*

It is clear that the facet-to-facet property is important when referring to the set of full-dimensional critical regions of (3.1). If the set of closures of the full-dimensional critical regions do not satisfy the facet-to-facet property, then it may be insufficient to only identify one adjacent region for each facet, as illustrated in Figure 3.1. The following example illustrates that the facet-to-facet property does not generally hold for strictly convex pQPs. Hence, the algorithms in (Baotić 2002), (Grieder, Borrelli, Torrisi, and Morari 2004), (Tøndel, Johansen, and Bemporad 2003a) and (Tøndel, Johansen, and Bemporad 2003c) cannot guarantee that the entire parameter space will be explored.

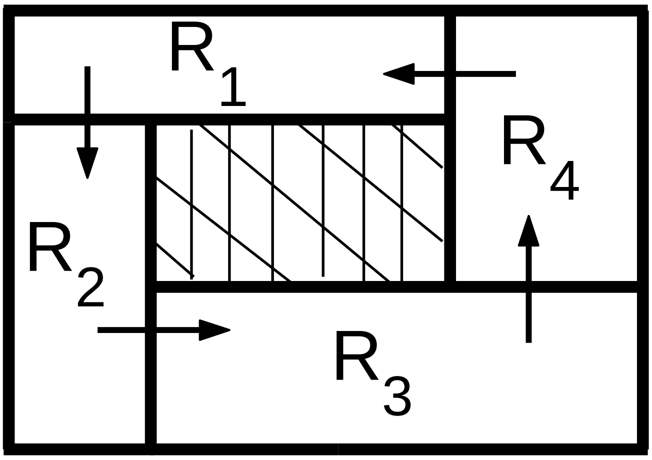


Figure 3.1: Illustration that Algorithm 3.1 may fail to identify all the critical regions if the facet-to-facet property does not hold, the strategies in Section 3.2.1 or 3.2.2 are employed at step 7 of Algorithm 3.1 and no additional assumptions on the problem are given. The shaded region is unexplored.

Example 3.1 *Consider the problem:*

$$V^*(\theta) := \min_{x \in \mathbb{R}^3} \left\{ \frac{1}{2} x^T x \;\middle|\; x \in \mathcal{P}(\theta) \right\},\ \theta \in \Theta,$$

$$\mathcal{P}(\theta) := \left\{ x \in \mathbb{R}^3 \;\middle|\; \begin{aligned} x_1 - x_3 &\le -1 + \theta_1 \\ -x_1 - x_3 &\le -1 - \theta_1 \\ x_2 - x_3 &\le -1 - \theta_2 \\ -x_2 - x_3 &\le -1 + \theta_2 \\ \tfrac{3}{4}x_1 + \tfrac{16}{25}x_2 - x_3 &\le -1 + \theta_1 \\ -\tfrac{3}{4}x_1 - \tfrac{16}{25}x_2 - x_3 &\le -1 - \theta_1 \end{aligned} \right\},$$

$$\Theta := \left\{ \theta \in \mathbb{R}^2 \;\middle|\; -\tfrac{3}{2} \le \theta_i \le \tfrac{3}{2},\ i = 1, 2 \right\}.$$

The unique set of full-dimensional critical regions is depicted in Figure 3.2, where we have indexed the critical regions with the optimal active sets. The critical regions $R_{\{1,4,5\}}$, $R_{\{1,3,6\}}$, $R_{\{2,4,5\}}$, *and* $R_{\{2,3,6\}}$ *have more than one adjacent critical region along one of their facets, hence the facet-to-facet property is violated for the set of closures of full-dimensional critical regions.*

In (Spjøtvold 2005) it is verified analytically *that LICQ holds for all optimal active sets, that the KKT conditions hold for* $(x^*(\theta), \lambda^*(\theta))$ *for a parameter in the interior of each full-dimensional critical region, and numerically verified that every other combination of active constraints yield empty or lower-dimensional critical regions. Thus, the violation of the facet-to-facet property is not a consequence of numerical inaccuracies. However, there is a lower-dimensional critical region of particular interest, namely the critical region defined by* $\mathcal{A} = \{1, \ldots, 6\}$, *which is analytically computed in (Spjøtvold 2005) as*

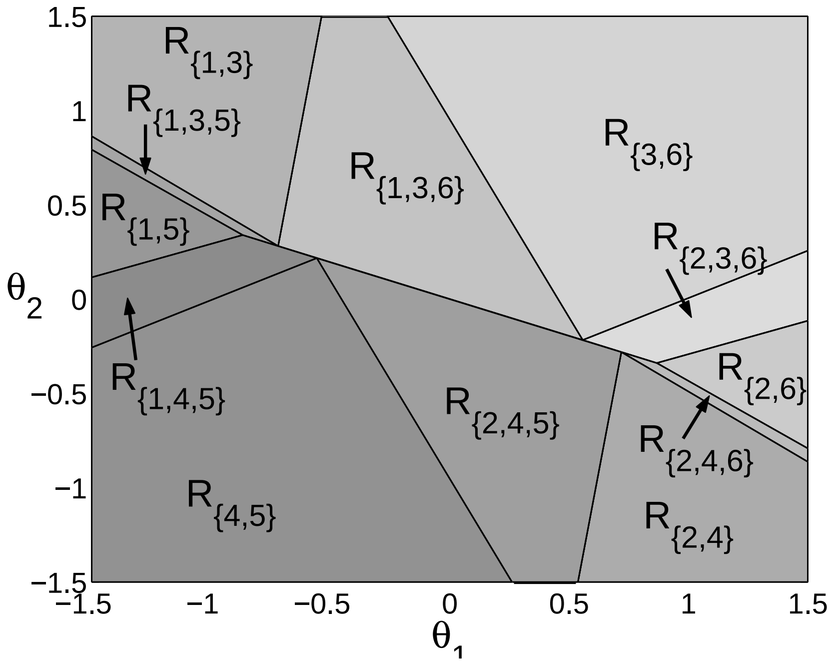


Figure 3.2: Facet-to-facet property violated.

$$\mathrm{cl}(\Theta_{\{1,\ldots,6\}}) = \left\{ \theta \;\middle|\; \theta_1 = -\frac{64}{25}\theta_2, -\frac{1600}{4721} \leq \theta_2 \leq \frac{1600}{4721} \right\}.$$

The representations of $R_{\{1,4,5\}}$, $R_{\{1,3,6\}}$, $R_{\{2,4,5\}}$, $R_{\{2,3,6\}}$, $R_{\{1,3,5\}}$, *and* $R_{\{2,4,6\}}$ *obtained from* (3.2) *all have three coinciding inequalities along the line* $\theta_1 = -\frac{64}{25}\theta_2$. *This suggests that, due to the statements in Section 3.2.2, coinciding inequalities in the description of the critical regions may be the reason for the violation of the facet-to-facet property. Empirical examination also shows that the presented example is not an isolated incident of the facet-to-facet property being violated. By letting the constant values on the right hand side be written as* $-[1, 1, 1, 1+\alpha, 1+\alpha, 1+\alpha]^T$, *the facet-to-facet property is violated for any* $\alpha \in [-\frac{1}{10}, \frac{2}{5}]$.

■

3.3 A new exploration strategy

The algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) does not rely on the facet-to-facet property but, as mentioned in the introduction, introduces a number of artificial cuts in the parameter space as it searches for the solution. As a consequence the performance degrades as the number of critical regions become large. In (Tøndel, Johansen, and Bemporad 2003a) the authors propose a more efficient way of exploring the parameter space, but it relies on the facet-to-facet property. We aim at modifying the algorithm in (Tøndel, Johansen, and Bemporad 2003a) in order to ensure its correctness.

The proposed method finds all critical regions adjacent to a critical region along a given facet and in order to preserve the computational advantages of the algorithm in (Tøndel, Johansen, and Bemporad 2003a) compared to the one in (Bemporad, Morari, Dua, and Pistikopoulos 2002), the modification is to be utilized *only* when the conditions in Section 3.2.2 do not hold. We use the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) to explore the parameter space in a small polyhedral subset $M \subset \Theta$ and discard the artificial cuts once the solution has been found. For a given optimal active set $\mathcal{A}$, if the goal is to identify the critical regions adjacent to $\Theta_{\mathcal{A}}$ along a given facet f of its closure, then the polyhedron M must be full-dimensional and satisfy the property:

$$\mathrm{cl}(\Theta_{\mathcal{A}}) \cap M = f.$$

For use in the proposed method, the set of optimal active sets associated with the polyhedron M is defined as:

$$\mathcal{C}(M) := \{\mathcal{A} \subseteq \{1, 2, \ldots, q\} \mid \dim\left(M \cap \mathrm{cl}\left(\Theta_{\mathcal{A}}\right)\right) = s\}\,.$$

A method for obtaining all adjacent regions is given in Procedure 3.2. Note that the number of critical regions that intersect M is expected to be small, hence the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) is well suited. Moreover, the artificial cuts made inside M are discarded once the exploration terminates, thus the artificial cuts do not cause the performance to degrade to the same extent as in (Bemporad, Morari, Dua, and Pistikopoulos 2002). The choice of ε in step 6 is arbitrary from a theoretical point of view, but it is important to note that too small a value will cause numerical problems and too large a value may result in an unnecessary increase in the computational effort. This issue will be further discussed in Section 3.4. Note that $\mathcal{C}(M_j)$ may define additional critical regions that are not adjacent to the critical region considered and/or critical regions that have already been discovered. However, this is not a problem since one can either choose to keep them as identified regions or discard them. In Procedure 3.2 we have chosen to return all those critical regions which are not adjacent to U and those that have already been discovered; step 8 of Algorithm 3.1 can be replaced by $\mathcal{U} \leftarrow \mathcal{U} \cup (\mathcal{S} \backslash \mathcal{R}) \cup (\mathcal{T} \backslash \mathcal{R})$ and step 9 by $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{S} \cup \mathcal{T}$.

We illustrate the difference between the exploration strategy in (Bemporad, Morari, Dua, and Pistikopoulos 2002) and the proposed method with an example.

Example 3.2 *Assume that the set of closures of full-dimensional critical regions for a pQP is as depicted in Figure 3.3(a). The first step of the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) is to find an initial critical region* R_1 *and then partition the rest of the parameter space into a set of unexplored polyhedra* $\mathcal{U}$*, see Figure 3.3(b). It then continues by exploring one of these polyhedra, for instance* U_1*, finds a new region* R_2 *and partitions the space again, see Figure 3.3(c). A possible third iteration is depicted in Figure 3.3(d). In Figure 3.3(e)*

Procedure 3.2 Identifying all adjacent full-dimensional critical regions along a given facet.

Input: Irredundant representation of the closure of a full-dimensional critical region $U =: \{\theta \mid C_i\theta \leq d_i,\ i = 1, \ldots, J\}$ and the index j whose corresponding inequality defines facet f.

Output: Set $\mathcal{S}$ of closures of full-dimensional critical regions adjacent to U along the facet f, and set $\mathcal{T}$ which is a subset of the full-dimensional critical regions not adjacent to U.

$\mathcal{S} \leftarrow \emptyset$ and $\mathcal{T} \leftarrow \emptyset$.
if the facet f is not on the boundary of Θ **then**
  if the conditions in Section 3.2.2 hold **then**
    Find the critical region $\Theta_{\mathcal{A}}$ that is adjacent to U along f as described in Section 3.2.2 and let $\mathcal{S} \leftarrow \{\mathrm{cl}\,(\Theta_{\mathcal{A}})\}$.
  else
    Choose any scalar $\varepsilon > 0$ and construct the polyhedron

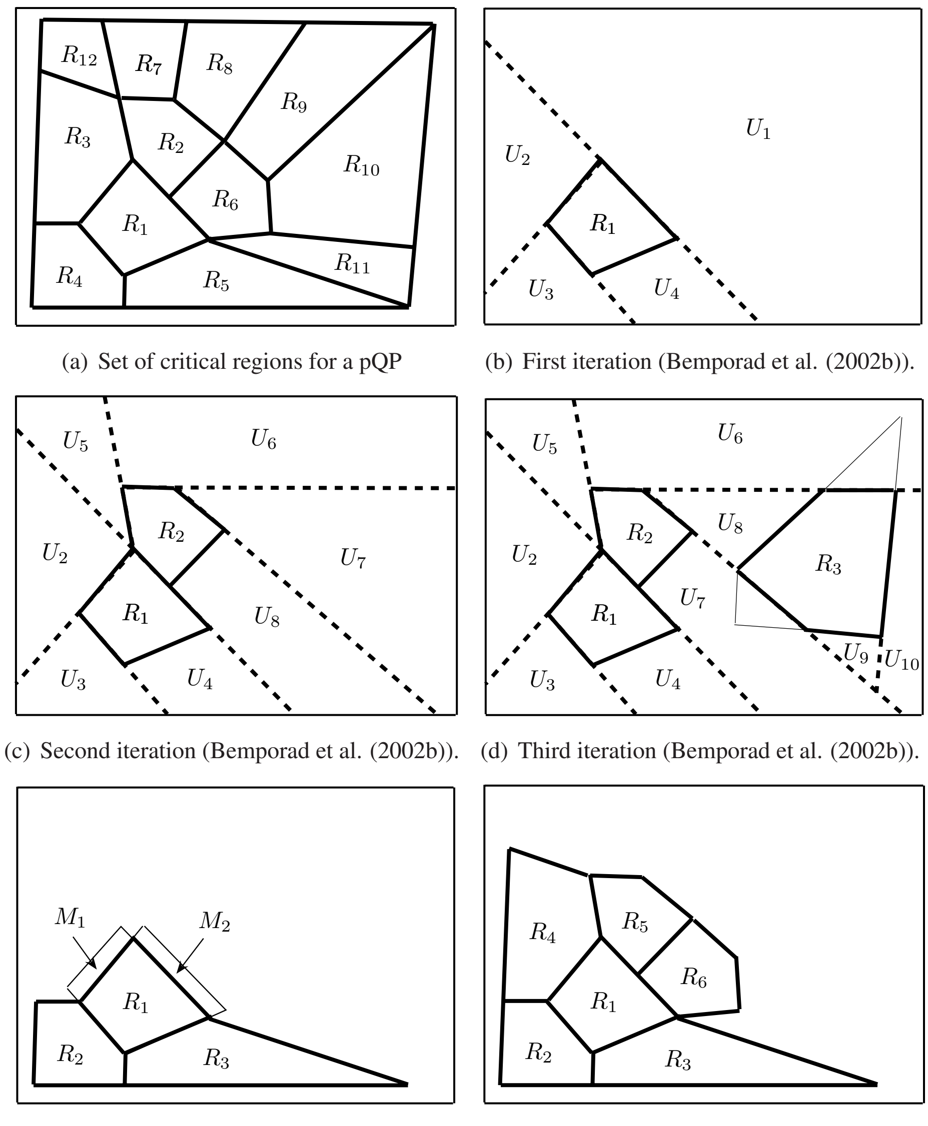
$$M_j := \left\{ \theta \in \Theta \;\middle|\; \begin{array}{rcl} C_i\theta & \leq & d_i,\ \forall i \in \{1, \ldots, J\} \backslash \{j\} \\ C_j\theta & \geq & d_j \\ C_j\theta & \leq & d_j + \varepsilon \end{array} \right\}$$

    Compute the set $\mathcal{C}(M_j)$ by solving the pQP (3.1) inside M_j using the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002).
    for each $\mathcal{A} \in \mathcal{C}(M_j)$ **do**
      if $\dim\,(\mathrm{cl}\,(\Theta_{\mathcal{A}}) \cap U) = s - 1$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathrm{cl}\,(\Theta_{\mathcal{A}})\}$. {Adjacent critical region}
      else
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathrm{cl}\,(\Theta_{\mathcal{A}})\}$.
      end if
    end for
  end if
end if

we have shown a possible first iteration of the proposed method. Note that for two facets of R_1 the conditions in Section 3.2.2 do not hold, and hence, the sets M_1 and M_2 are constructed. After identifying the optimal active sets in M_j, the set of critical regions is as illustrated in Figure 3.3(f).

■

The two key issues we want to illustrate with the above example is that i) for the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) the artificial cuts affect the exploration strategy in parts of the parameter space where the cuts are unnecessary, causing the performance to degrade for large K, and ii) the proposed method discards the artificial partitioning once a set M_j has been fully explored. Since the number of regions intersecting M is expected to be small, the



(a) Set of critical regions for a pQP

(b) First iteration (Bemporad et al. (2002b)).

(c) Second iteration (Bemporad et al. (2002b)).

(d) Third iteration (Bemporad et al. (2002b)).

(e) First iteration. proposed method

(f) Regions after artificial cuts are discarded

Figure 3.3: Illustration of different exploration strategies

algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) is well suited to explore inside M_j.

The computational advantages of the algorithm in (Tøndel, Johansen, and Bemporad 2003a) compared to the one in (Bemporad, Morari, Dua, and Pistikopoulos 2002) is well documented, so the performance of the proposed procedure relies on how often the conditions in Section 3.2.2 do not hold. Numerical results will be given in the next section. Before we prove the correctness of the algorithm, we need a technical lemma, which is proven in (Spjøtvold 2005) and included in Appendix B.

Lemma 3.1 *Given two s-dimensional closed sets, P and S, in $\mathbb{R}^s$, such that*

$$\mathrm{int}(P) \cap \mathrm{int}(S) = \emptyset.$$

A necessary condition for the set $P \cup S$ to be convex, is that

$$\dim(P \cap S) = s - 1.$$

Theorem 3.2 (Correctness of the Algorithm) *Algorithm 3.1 combined with Procedure 3.2 for Step 7 ensures that $\cup_{k=1}^{K} R_k = \Theta$.*

PROOF: Assume that $\mathcal{R}$ is the output of the algorithm and that $\cup_{R\in\mathcal{R}} R \subset \Theta$. Let

$$\mathcal{P} := \{\mathrm{cl}\,(\Theta_{\mathcal{A}}) \mid \dim\,(\Theta_{\mathcal{A}}) = s \text{ for (3.1)}\} \setminus \mathcal{R},$$

and let M_j^R denote the set in Procedure 3.2 associated with the j^{th} facet of $R \in \mathcal{R}$. By the correctness of the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) and the fact that $\dim\left(\mathrm{cl}\,(\Theta_{\mathcal{A}}) \cap M_j^R\right) = s$ if R and $\Theta_{\mathcal{A}}$ are adjacent along the j^{th} facet of R, all full-dimensional critical regions adjacent to R have been identified. Hence, for any pair $(R, P) \in \mathcal{R} \times \mathcal{P}$ we must have $\dim\,(R \cap P) < s - 1$, otherwise P would be a member of $\mathcal{R}$, and consequently, $\dim\,((\cup_{R\in\mathcal{R}} R) \cap (\cup_{P\in\mathcal{P}} P)) < s - 1$. Moreover, we have $\Theta = (\cup_{R\in\mathcal{R}} R) \cup (\cup_{P\in\mathcal{P}} P)$. Hence, by Lemma B.1 a contradiction is reached, since Θ is convex.

□

3.4 Numerical example

In this section we make a quantitative comparison of the following exploration strategies: (i) the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002), and (ii) the proposed algorithm of combining Algorithm 3.1 with Procedure 3.2 for Step 7. The algorithms are tested on an MPC problem for a linear time invariant system

$$z(k+1) = \Phi z(k) + \Gamma u(k),\ z(0) = z_0, \tag{3.3}$$

where $z(k) \in \mathbb{R}^4$ and $u(k) \in \mathbb{R}^2$ are the state and input at time k, respectively, and Φ and Γ are matrices with suitable dimensions. The objective is to minimize the following cost function

$$J(z_0) := \sum_{k=1}^{N} \left(z(k)^T Q z(k) + u(k-1)^T R u(k-1)\right),$$

where $Q = Q^T \geq 0$ and $R = R^T > 0$, subject to the system equation (3.3), state constraints $z \in \mathcal{Z} := \{z \mid \underline{z} \leq z \leq \overline{z}\}$, and input constraints $u \in \mathcal{U} := \{u \mid \underline{u} \leq u \leq \overline{u}\}$. This problem is recast as a pQP as described in (Bemporad, Morari, Dua, and Pistikopoulos 2002) and the algorithms are tested on 80 random instances of $(\Phi, \Gamma, Q, R, \mathcal{Z}, \mathcal{U})$ with a prediction horizon $N \in \{3, 4, 5\}$. For simplicity, all systems are stable, controllable and observable. The solutions have an average of 317 critical regions and Figure 3.4 compares the total number of optimization problems solved by the algorithms. As expected, the computational

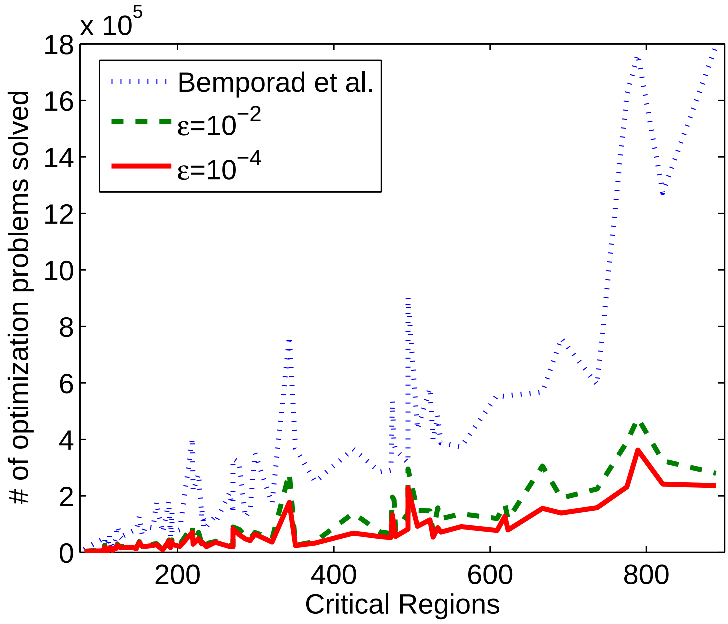


Figure 3.4: Comparison of the number of optimization problems solved by the algorithm.

effort used to find an explicit solution is on average lowest for alternative (ii). This shows that alternative (ii) is preferable also in practice. Note that although the performance of the proposed method relies on the choice of ε, it is not difficult to choose a value such that the proposed method is more efficient than the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002). Also, from Figure 3.4 it is apparent that the difference in the computational effort is expected to grow as the number of critical regions K increases.

3.5 Conclusion

It has been shown by an example that, for strictly convex parametric quadratic programs (and for programs that are convex, but not strictly convex), a critical region may have more than one adjacent critical region for each facet, hence the facet-to-facet property does not hold, in general. This renders some of the recently developed algorithms for this problem class without guarantees that the entire parameter space will be explored. A simple method based on the algorithms in (Bemporad, Morari, Dua, and Pistikopoulos 2002) and (Tøndel, Johansen, and Bemporad 2003a) was proposed such that the completeness of the exploration strategy is guaranteed. Numerical results also indicate that the proposed method is computationally more efficient than the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) in practice.

Part II

Applications of Parametric Programming In Control Theory

Chapter 4

Inf-Sup Control of Discontinuous Piecewise Affine Systems

4.1 Introduction

Gal and Nedoma proposed complete algorithm for solving parametric linear programs (pLP) as early as (Gal and Nedoma 1972) and in the following years the topic was subject of substantial research, see e.g. (Bank, Guddat, Klatte, Kummer, and Tammer 1983; Schechter 1987) and references therein. Parametric programming has had a resurgence of interest recently due to the observation that explicit control laws for some model predictive control problems (Mayne, Rawlings, Rao, and Scokaert 2000) can be obtained by viewing the initial state as a vector of parameters (Bemporad, Morari, Dua, and Pistikopoulos 2002; Bemporad, Borrelli, and Morari 2002; Seron, Goodwin, and Doná 2003). As researchers tried to characterize the solution to more difficult optimal control problems (piecewise affine systems, uncertain systems, non-linear systems etc.), several variations of the dynamic programming approach originally proposed by Bellman (Bellman 1957) has been employed (Witsenhausen 1968; Bertsekas and Rhodes 1973; de la Peña, Alamo, Bemporad, and Camacho 2002; Diehl and Björnberg 2004; Raković, Kerrigan, and Mayne 2004; Borrelli, Baotić, Bemporad, and Morari 2005; Kerrigan and Mayne 2002; Mayne, Rakovć, Vinter, and Kerrigan 2006; Lincoln and Rantzer 2006; Rantzer 2006). The dynamic programming approach for $\inf$–$\sup$ (or more commonly $\min$–$\max$) control was introduced for linear systems as early as (Witsenhausen 1968) and in a more general framework in (Bertsekas and Rhodes 1973). The methods (Witsenhausen 1968; Bertsekas and Rhodes 1973) have served as the foundation for a large number of publications (see e.g. (de la Peña, Alamo, Bemporad, and Camacho 2002; Diehl and Björnberg 2004; Raković, Kerrigan, and Mayne 2004; Borrelli, Baotić, Bemporad, and Morari 2005; Kerrigan and Mayne 2002; Mayne, Rakovć, Vinter, and Kerrigan 2006; Lincoln and Rantzer 2006; Rantzer 2006)) and will also be utilized in this chapter.

Recently, methods for computing explicit control laws for discrete-time piecewise affine (PWA) systems with constraints have been reported in the control literature (Borrelli, Baotić, Bemporad, and Morari 2005; Raković, Kerrigan, and Mayne 2004; Mayne and Raković 2002; Bemporad, Borrelli, and Morari 2000b; Bemporad, Borrelli, and Morari 2000a; Borrelli 2002; Kerrigan and Mayne 2002). There are several reasons why obtaining explicit solutions to this problem class has been of interest; i) Piecewise affine systems arise both naturally (e.g. backlash (Rostalski, Besselmann, Barić, and Morari 2007) and hysteresis) and as approximations to non-linear systems. Piecewise affine models are also equivalent to a large class of hybrid systems (Heemels, Shutter, and Bemporad 2001). ii) An explicit solution to an optimal control problem offers several advantages compared to the on-line counterpart. The required on-line computation time is reduced, rendering optimal control applicable also for fast systems. In addition, the explicit solution makes it possible to off-line verify the correctness of the control law, which is a key point in safety critical applications. iii) Although several authors have addressed the topic of obtaining explicit solutions to optimal control problems for non-linear systems (Johansen 2002; Johansen 2004a; Grancharova and Johansen 2005; Bemporad and Filippi 2006), these methods are often computationally demanding, sub-optimal and may lack stability guarantees. Even for linear systems with quadratic cost functions it is difficult to incorporate disturbances/uncertainties as the $\min$–$\max$ formulation results in a non-convex problem that is not easily divided into sub-problems. iv) Recent results on parametric linear programming (Jones, Kerrigan, and Maciejowski 2007; Spjøtvold, Tøndel, and Johansen 2005a; Jones and Maciejowski 2006; Borrelli, Bemporad, and Morari 2003) and the evaluation of piecewise affine control laws (Christophersen, Kvasnica, Jones, and Morari 2007; Jones, Grieder, and Raković 2006; Tøndel, Johansen, and Bemporad 2003b) are directly applicable to optimal control of discrete-time PWA systems with linear cost functions as these problems can be solved explicitly with a series of pLPs.

In (Lincoln and Rantzer 2006; Rantzer 2006) a relaxed dynamic programming procedure is proposed. The procedure reduces the computational complexity by relaxing the demands for optimality. The existence of a solution is assumed in (Lincoln and Rantzer 2006; Rantzer 2006), which is different from the objective in this chapter; to obtain a sub-optimal solution only when the infimum cannot be attained.

In (Borrelli, Baotić, Bemporad, and Morari 2005) a dynamic programming approach for obtaining explicit control laws to continuous PWA systems not affected by disturbances is presented. Discontinuous PWA systems are briefly mentioned, but the topic is not treated in detail; for instance that an optimizer exists[1] cannot be guaranteed. In addition, the authors represent the domain of the PWA state update equation by closed polyhedra, as described in (Bemporad and Morari 1999), and therefore small gaps are introduced in the domain of the state update equation. Consequently, from a theoretical point of view, the state trajectory may vanish. On

[1]Given an optimization problem, e.g. $J^* := \inf_{x \in X} f(x)$, we say that an optimizer exists if the infimum is attained, i.e. $\exists\, x^* \in X$ such that $J^* = f(x^*)$.

the other hand, one can argue that the control scheme is to be implemented on a microchip or computer and therefore is subject to a finite arithmetic precision. In this chapter we will look at the problem from a theoretical viewpoint and remove the need to introduce small gaps.

An alternative way of describing a discontinuous PWA system is to transform the state update equation into a difference inclusion by performing a regularization, see e.g. (Goebel and Teel 2006). With this system description, the successor state may be set-valued for a given initial state, control input and disturbance. We seek to avoid this situation by still treating the system as a difference equation.

In (Raković, Kerrigan, and Mayne 2004) continuous PWA systems with piecewise affine cost functions subject to state- and input-dependent disturbances are considered. This chapter presents a non-trivial extension of this approach to *discontinuous* PWA systems. Methods for computing explicit control laws for this problem class have not yet been reported in the literature, hence, in this chapter we outline the foundation for a complete computational procedure.

In this chapter we represent the domain of PWA systems by a union of a finite number of open, closed and/or neither open nor closed polyhedra. A solution to the optimal control problem may not exist in this case. However, solutions for which the cost is arbitrarily close to the infimum/supremum are guaranteed to exist. We propose a procedure that obtains a sub-optimal solution to the optimal control problem when the solution does not exist, and the exact solution when it does. This approach does not introduce gaps in the domain of the state update equation, we do not assume that a solution to the optimal control problem exists, and the state update equation is not transformed into a difference inclusion, and thus, the dynamic programming approach is relatively simple from the theoretical point of view. In addition, the proposed procedure allows the degree of sub-optimality to be specified *a priori*.

CHAPTER STRUCTURE: Section 4.2 introduces basic notation and presents an illustrative and motivating example. In Section 4.3 we introduce the basic building block in the chapter, namely how one can obtain sub-optimal solutions to parametric linear programs with strict and non-strict inequality constraints. This building block is then used to obtain sub-optimal solutions to minimization of PWA functions in Section 4.4, to min-max problems in Section 4.5, and finally it is demonstrated in Section 4.6 how these procedures can be used in the dynamic programming approach for the purpose of obtaining explicit sub-optimal solutions to robust optimal control problems for discontinuous PWA systems.

4.2 Preliminaries

4.2.1 Basic Notation and Fundamental Results

For completeness we recall some standard notation and definitions. The *affine hull* of a set S is the intersection of all affine sets containing S, and is denoted $\mathrm{aff}(S)$. The *dimension of a set* $S \subseteq \mathbb{R}^n$ is the dimension of $\mathrm{aff}(S)$, and is denoted $\dim(S)$;

if $\dim(S) = n$, then S is said to be full-dimensional. The *closure* of a set S is denoted $\mathrm{cl}(S)$. The *relative interior* of a set S is the interior relative to $\mathrm{aff}(S)$, i.e.

$$\mathrm{relint}(S) := \{x \in S \mid B(x,r) \cap \mathrm{aff}(S) \subseteq S \text{ for some } r > 0\},$$

where the ball $B(x,r) := \{y \mid \|y - x\| \leq r\}$ and $\|\cdot\|$ is any norm. We denote the orthogonal projection of a set $S \subseteq \mathbb{R}^n \times \mathbb{R}^m$ to $\mathbb{R}^n$ by $\mathrm{Proj}_x S := \{x \in \mathbb{R}^n \mid \exists u \in \mathbb{R}^m \text{ such that } (x,u) \in S\}$.

A *polyhedron* is the intersection of a finite set of open and/or closed halfspaces. Note that our definition of a polyhedron differs from the most common definition in the sense that we do not require the halfspaces to be closed. The reason for this variation is to be able to represent discontinuous piecewise affine functions. A *polygon* is a union of finite number of polyhedra. We will adopt a similar notation to that presented in (Rockafellar and Wets 1998) with regards to the concept of *extended real valued* functions. Thus, a function f is allowed to take values in $\overline{\mathbb{R}} = [-\infty, \infty]$. Recall also that the *infimum* is the greatest lower bound of a set $S \subseteq \mathbb{R}$, defined as a quantity m such that no member of the set is less than m, but if ε is any positive quantity, however small, there is always one member s that is less than $m + \varepsilon$. The infimum of a set $S \subseteq \mathbb{R}$ exists in $\mathbb{R}$ if and only if S is non-empty and bounded from below. Moreover, we introduce the notation $\inf_C f := \inf_{x \in C} f(x) := \inf\{f(x) \mid x \in C\}$ and $\sup_C f := \sup_{x \in C} f(x) := \sup\{f(x) \mid x \in C\}$. By convention we have $\inf_\emptyset f = \infty$ and $\sup_\emptyset f = -\infty$, and hence,

$$\arg\min_{x \in C} f(x) := \begin{cases} \{x \in C \mid f(x) = \inf_C f\} & \text{if } \inf_C f \neq \infty, \\ \emptyset & \text{if } \inf_C f = \infty, \end{cases}$$

and

$$\arg\max_{x \in C} f(x) := \begin{cases} \{x \in C \mid f(x) = \sup_C f\} & \text{if } \sup_C f \neq -\infty, \\ \emptyset & \text{if } \sup_C f = -\infty. \end{cases}$$

We say that the infimum (supremum) of f over C is *attained* if $\arg\min_{x \in C} f(x) \neq \emptyset$ ($\arg\max_{x \in C} f(x) \neq \emptyset$). For a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$, the *domain* of f is defined as the set

$$\mathrm{dom}\,(f) := \{x \in \mathbb{R}^n \mid -\infty < f(x) < \infty\}.$$

Whenever we refer to a function f or mapping F having a certain property, we implicitly mean that the property holds only on the domain of f or F, e.g. if we say that f is continuous, it is continuous at every $x \in \mathrm{dom}\,(f)$.

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is *piecewise affine* (PWA) on its domain if $\mathrm{dom}\,(f)$ is the union of finitely many polyhedra, relative to each of which $f(\cdot)$ is affine. If $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$, then 2^Y is the *power set* (set of all subsets) of Y and a set-valued map is defined as $F : X \to 2^Y$. For notational simplicity, we use double arrows to specify that a mapping is set-valued, i.e. set-valued maps are specified as $F : X \rightrightarrows Y$. We say that the set-valued map $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is *piecewise polyhedral* if the graph of F, defined as $\mathrm{gph}\,(F) := \{(x, u) \mid u \in F(x)\}$, is a polygon. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is a *selection* of the set-valued map $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ if $f(x) \in F(x)$ for all $x \in \mathrm{dom}\,(F)$. If $F : X \rightrightarrows Y$ is a mapping, then the restriction of F to the domain D is written $F|_D : D \rightrightarrows Y$.

Throughout we will use the superscript * to distinguish between optimizers and decision variables, e.g. for the problem $\min_x f(x)$, x is the decision variable and x^* denotes an optimizer. We also let *such that* be abbreviated *s.t.*

Given the optimization problem

$$J^* := \inf_{u \in U} f(u),$$

we denote by $\varepsilon\text{-}\arg\min_{u\in U} f(u)$ the set of values of $u \in U$ for which $f(u) \leq J^* + \varepsilon$, that is,

$$\varepsilon\text{-}\arg\min_{u\in U} f(u) := \{u \in U \mid f(u) \leq J^* + \varepsilon\}.$$

The following observation follows directly from the definition of the infimum of a set, but for completeness we include a proof.

Lemma 4.1 *Assume that $J^* > -\infty$, that $U \subseteq \mathbb{R}^n$ is a non-empty set and $f : \mathbb{R}^n \to \mathbb{R}$, then $\varepsilon\text{-}\arg\min_{u\in U} f(u) \neq \emptyset$ for all $\varepsilon > 0$.*

PROOF: That the set U is non-empty implies that $F := \{f(u) \mid u \in U\}$ is non-empty. The infimum of F exists in $\mathbb{R}$ if and only F is non-empty and bounded from below, which is true by assumption ($J^* > -\infty$). Since J^* is the infimum of the set F, then from the definition of the infimum of a set we have that there exists $\bar{f} \in F$ such that $\bar{f} < J^* + \varepsilon$, $\forall \varepsilon > 0$. Since $\bar{f}$ is a member of F there exists $u \in U$ such that $f(u) = \bar{f}$.

□

In the sequel we will, for the problem of minimizing (maximizing) a function f over a set C, assume that $\inf_C f$ ($\sup_C f$) is bounded, and that the set C over which the optimization is performed, is non-empty.

4.2.2 Example

In this section we present a motivating example where we show that for a simple PWA function the infimum cannot be attained. The reader may interpret x as the initial state and u as the control input in the following example to see that there may be a problem to compute a control law that attains the infimum. It should also be noted that minimization of PWA functions are sub-problems in the dynamic programming recursion that may be utilized to compute explicit solutions to optimal control problems.

Example 4.1

Consider the following function:

$$f(x,u) := \begin{cases} f_1(x,u) := 2 & \text{if} \quad (x,u) \in P_1 \\ f_2(x,u) := -u + x & \text{if} \quad (x,u) \in P_2 \\ f_3(x,u) := -2 & \text{if} \quad (x,u) \in P_3 \end{cases} \tag{4.1a}$$

$$P_1 := \{(x,u) \in \mathbb{R} \times \mathbb{R} \mid u \geq 0, -5 \leq x \leq 5\}, \tag{4.1b}$$

$$P_2 := \{(x,u) \in \mathbb{R} \times \mathbb{R} \mid u < 0, -5 \leq x \leq 5\}, \tag{4.1c}$$

$$P_3 := \{(x,u) \in \mathbb{R} \times \mathbb{R} \mid 5 < x \leq 7\}, \tag{4.1d}$$

which is to be minimized with respect to u for all values of $x \in \mathcal{X} := [-5, 7]$, that is,

$$J^*(x) := \inf_u f(x,u), \quad \forall x \in \mathcal{X}.$$

Figure 4.1(a) illustrates $f(0,u)$ for $u \in [-7, 7]$; clearly the infimum $J^*(0) = 0$ cannot be attained, that is, $\nexists u^*$ such that $f(0,u^*) = J^*(0) = 0$. In fact, the infimum cannot be attained for any $x \in [-5, 2)$. The value function $J^*(\cdot)$ is depicted in Figure 4.1(b) and it is indicated where a solution does not exist. Two strategies that might be natural to consider in order to overcome this problem is to either treat $f(\cdot)$ as set-valued, or to introduce small gaps in the domain of $f(\cdot)$, such that each affine function is defined over a closed polyhedron. If we treat $f(\cdot)$ as a set-valued map by removing the strict inequalities:

$$\bar{f}(x,u) := \{f_i(x,u) \mid (x,u) \in \mathrm{cl}\,(P_i)\}, \tag{4.2}$$

then the naive approach of choosing the minimum of $\bar{f}(\cdot)$ will change the value function compared to original the problem, see Figure 4.2(a). In addition, we see that even though $\bar{f}(x,\cdot)$ attains its minimum for all $x \in \mathcal{X}$, this is not case for $f(x,\cdot)$.

Introducing small gaps in the domain of the function, that is,

$$\tilde{f}(x,u) : = \begin{cases} 2 & \text{if} \quad (x,u) \in \tilde{P}_1 \\ -u + x & \text{if} \quad (x,u) \in \tilde{P}_2 \\ -2 & \text{if} \quad (x,u) \in \tilde{P}_3 \end{cases} \tag{4.3a}$$

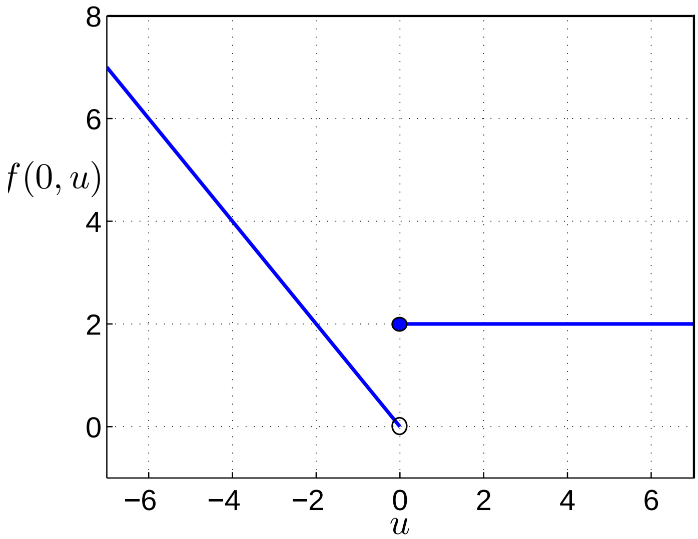
$$\tilde{P}_1 := \{(x,u) \in \mathbb{R} \times \mathbb{R} \mid u \geq 0, -5 \leq x \leq 5\}, \tag{4.3b}$$

$$\tilde{P}_2 := \{(x,u) \in \mathbb{R} \times \mathbb{R} \mid u \leq -\delta, -5 \leq x \leq 5\}, \tag{4.3c}$$

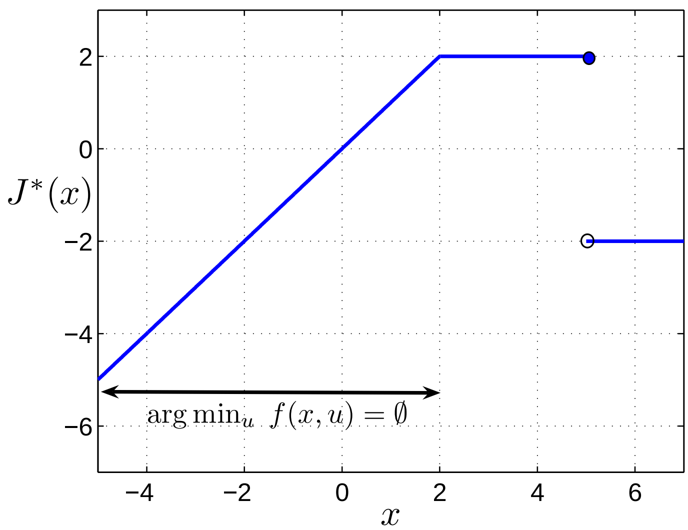
$$\tilde{P}_3 := \{(x,u) \in \mathbb{R} \times \mathbb{R} \mid 5 + \delta \leq x \leq 7\}, \tag{4.3d}$$

where $\delta > 0$, will result in that $\tilde{f}(x,\cdot)$ is undefined for some $x \in \mathcal{X}$, see Figure 4.2(b).

■



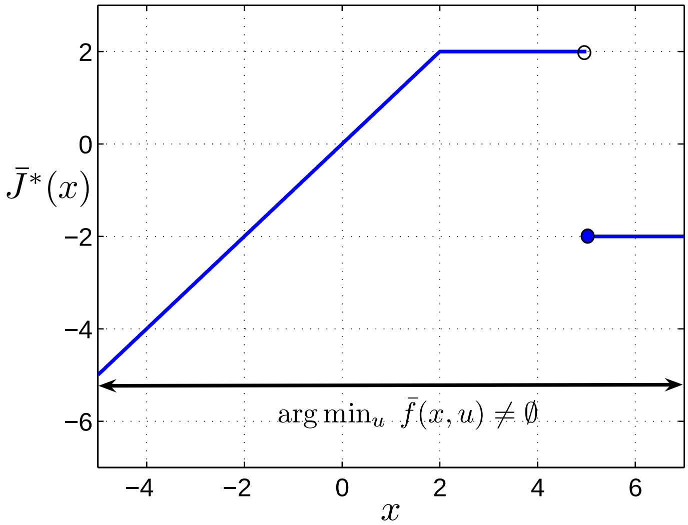
(a) Illustration of $f(0, \cdot)$. At the discontinuity the filled circle indicates the value of $f(0, \cdot)$, i.e. $f(0, 0) = 2$.



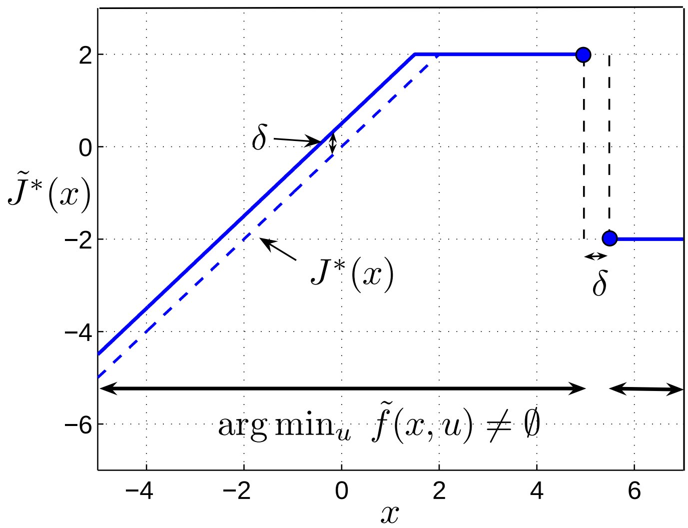
(b) Illustration of $J^*(\cdot)$. At the discontinuity the filled circle indicates the value of $J^*(\cdot)$, i.e. $J^*(5) = 2$. For $x \in [-5, 2)$ we have $\arg\min_u f(x, u) = \emptyset$.

Figure 4.1: Exact solutions for minimization of (4.1a).

We see from the above example that, even for a very simple PWA function, the infimum cannot be attained. In the sequel, we will propose a procedure that guarantees that for all x for which the infimum is bounded, we obtain a u^* such that $f(x, u^*) \leq J^*(x) + \varepsilon$, where the scalar $\varepsilon > 0$ can be specified *a priori*.



(a) Illustration of $\bar{J}^*(x) := \inf_u \bar{f}(x,u)$. It is concluded that $\exists u^*$ s.t $\bar{J}^*(x) = \bar{f}(x,u^*)$ for all $x \in [-5,2)$, however, $\nexists u^*$ s.t $J^*(x) = f(x,u^*)$, cf. Figure 4.1(b). Moreover, $\bar{J}^*(5) = -2 \neq J^*(5) = 2$.



(b) Illustration of $\tilde{J}^*(x) := \inf_u \tilde{f}(x,u)$. The dotted line depicts $J^*(\cdot)$. The solid line depicts $\tilde{J}^*(\cdot)$. We see that in the interval $5 < x < 5+\delta$ the value function is undefined. In the interval $5+\delta \leq x \leq 7$, $\tilde{J}^*(\cdot)$ coincides with $J^*(\cdot)$.

Figure 4.2: Alternative approaches for the solution of (4.1a).

4.3 ε-Optimal Solutions to Parametric Linear Programs with Strict and Non-strict Inequality Constraints

In this section we consider the problem of finding ε-optimal solutions to parametric linear programs with strict and non-strict inequalities. We propose a procedure that will be repeatedly used in subsequent sections for the purpose of obtaining sub-optimal solutions to optimal control problems with piecewise affine cost and polygonic constraints.

Consider the problem

$$\mathbb{P}(x): \quad J^*(x) := \inf_u \left\{ c^T u \mid (x,u) \in \mathcal{Z} \right\}, \tag{4.4a}$$

$$\mathcal{Z} := \left\{ (x,u) \in \mathbb{R}^n \times \mathbb{R}^m \;\middle|\; \begin{array}{lcl} Au + Bx & \leq & d \\ Eu + Fx & < & g \end{array} \right\}, \tag{4.4b}$$

which is to be solved for all $x \in \mathcal{X}$, where $\mathcal{X}$ is defined as:

$$\mathcal{X} := \{x \in \mathbb{R}^n \mid \exists u \text{ s.t. } (x,u) \in \mathcal{Z}\} \cap \{x \in \mathbb{R}^n \mid J^*(x) > -\infty\} = \mathrm{dom}\,(J^*)\,, \tag{4.5}$$

and c, A, B, d, E, F and g are matrices with suitable dimensions.

The constraint set $\mathcal{Z}$ defines the set-valued map $U : \mathcal{X} \rightrightarrows \mathbb{R}^m$ given by

$$U(x) := \{u \mid (x,u) \in \mathcal{Z}\} \tag{4.6}$$

and hence (4.4) can be written as

$$J^*(x) = \inf_u \left\{ c^T u \mid u \in U(x) \right\}.$$

We make the following standing assumption:

Assumption 4.1 *For any parametric optimization problem that can be expressed as $z^*(\theta) := \inf_y \{f(\theta,y) \mid (\theta,y) \in \mathcal{Y}\}$, $\forall \theta \in \Theta$, the sets $\mathcal{Y}$ and*

$$\{\theta \mid \exists y \text{ s.t. } (\theta,y) \in \mathcal{Y}\} \cap \{\theta \in \Theta \mid z^*(\theta) > -\infty\}$$

are non-empty.

Returning to our original problem $\mathbb{P}(\cdot)$, the above assumption implies that $\mathcal{Z} \neq \emptyset$, $\mathcal{X} \neq \emptyset$ and that $\mathcal{X}$ is polyhedral.

For pLPs with only non-strict inequalities the following is well-known (Gal and Nedoma 1972; Bank, Guddat, Klatte, Kummer, and Tammer 1983; Dantzig, Folkman, and Shapiro 1967; Borrelli, Bemporad, and Morari 2003):

Theorem 4.1 (Solution properties for pLPs) *Consider the pLP*

$$\hat{J}^*(x) := \min_u \left\{ c^T u \mid (x,u) \in \mathrm{cl}\,(\mathcal{Z}) \right\}, \tag{4.7}$$

which is to be solved for all values of $x \in \widehat{\mathcal{X}}$, where

$$\begin{aligned} \widehat{\mathcal{X}} &:= \{x \in \mathbb{R}^n \mid \exists u \text{ s.t. } (x,u) \in \mathrm{cl}\,(\mathcal{Z})\} \cap \left\{ x \in \mathbb{R}^n \;\middle|\; \hat{J}^*(x) > -\infty \right\} \\ &= \mathrm{dom}(\hat{J}^*), \end{aligned}$$

and $\mathcal{Z}$ is defined in (4.4b).

$i)$ *There exists a continuous and PWA function* $u^* : \widehat{\mathcal{X}} \to \mathbb{R}^m$ *that satisfies*

$$u^*(x) \in \arg\min_u \left\{ c^T u \mid (x,u) \in \mathrm{cl}\,(\mathcal{Z}) \right\},\ \forall x \in \widehat{\mathcal{X}}.$$

$ii)$ *The value function* $\hat{J}^* : \widehat{\mathcal{X}} \to \mathbb{R}$ *is continuous, convex and piecewise affine.*

Remark 4.1 *Note that according to the most common definitions of a continuous point to set map (Berge 1963; Hausdorff 1957; Aubin and Frankowska 1990), the mapping* $x \mapsto \arg\min_u \left\{ c^T u \mid (x,u) \in \mathrm{cl}\,(\mathcal{Z}) \right\}$ *is continuous, and several continuous selections are available; for instance minimum norm, Steiner point, or any extremal selector. For computational approaches, see (Spjøtvold, Tøndel, and Johansen 2007; Jones, Kerrigan, and Maciejowski 2007; Jones 2005)*

Before we present a theorem that provides a procedure for obtaining ε-optimal solutions for parametric linear programs with strict and non-strict inequalities, we recall a fundamental result for support functions to convex sets, formulated as a lemma for clarity of presentation (Rockafellar 1972, page 112):

Lemma 4.2 *If* $S \subseteq \mathbb{R}^m$ *is a convex set and* $c \in \mathbb{R}^m$ *is given, then*

$$\inf_u \left\{ c^T u \mid u \in S \right\} = \inf_u \left\{ c^T u \mid u \in \mathrm{cl}\,(S) \right\} = \inf_u \left\{ c^T u \mid u \in \mathrm{relint}\,(S) \right\}.$$

We cannot immediately use the above lemma in a parametric setting, so we provide the following corollary:

Corollary 4.1 *Consider* (4.4) *and* (4.7)*. Since* $\mathcal{Z}$ *is polyhedral we have that*

$$J^*(x) = \hat{J}^*(x) \quad \textit{for all} \quad x \in \mathcal{X} = \mathrm{dom}\,(J^*)\,.$$

PROOF:
Using Lemma 4.2 we have

$$\inf_u \left\{ c^T u \mid u \in U(x) \right\} = \inf_u \left\{ c^T u \mid u \in \mathrm{cl}\,(U(x)) \right\}, \quad \forall x \in \mathrm{dom}\,(J^*)\,.$$

We now need to show that

$$\inf_u \left\{ c^T u \mid u \in \mathrm{cl}\,(U(x)) \right\} = \inf_u \left\{ c^T u \mid (x,u) \in \mathrm{cl}\,(\mathcal{Z}) \right\}, \quad \forall x \in \mathrm{dom}\,(J^*)\,.$$

Defining $\bar{U}(x) := \{u \mid (x,u) \in \mathrm{cl}\,(\mathcal{Z})\}$ *we can write* (4.7) *as*

$$\inf_u \left\{ c^T u \;\middle|\; u \in \bar{U}(x) \right\},$$

and since $\mathcal{Z}$ *is polyhedral it trivially follows that*

$$\mathrm{cl}\,(U(x)) = \left\{ u \;\middle|\; \begin{array}{l} Au \le d - Bx \\ Eu \le g - Fx \end{array} \right\} = \bar{U}(x). \qquad \square$$

A parametric optimization is said to be a *parametric piecewise LP* if the set of parameters for which the infimum is bounded is the union of a finite number of polyhedra, relative to each of which the problem reduces to a pLP. Consider $\mathbb{P}(\cdot)$ and define the parametric piecewise LP ($\hat{J}^*(\cdot)$ is PWA and equal to $J^*(\cdot)$ on $\mathcal{X}$):

$$\mathbb{P}_\varepsilon(x) : V_\varepsilon^*(x) := \min_{(u,t)} \{t \mid (x,u,t) \in \mathcal{Z}_\varepsilon\}, \tag{4.8a}$$

$$\mathcal{Z}_\varepsilon := \left\{ (x,u,t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \;\middle|\; \begin{array}{rcl} Au + Bx & \leq & d \\ Eu + Fx & \leq & g + \mathbf{1}t \\ c^T u & \leq & \hat{J}^*(x) + \varepsilon \end{array} \right\}, \tag{4.8b}$$

where $\mathbf{1}$ is a vector of ones and ε is a positive scalar, which is to be solved for all values of $x \in \mathcal{X}_\varepsilon$, where

$$\mathcal{X}_\varepsilon := \widehat{\mathcal{X}} \cap \operatorname{Proj}_x \mathcal{Z}_\varepsilon. \tag{4.9}$$

The following theorem provides a means for obtaining ε-optimal solutions to pLPs with strict and non-strict inequality constraints:

Theorem 4.2 *Consider the optimization problems given in* (4.4), (4.7) *and* (4.8) *and let ε be an arbitrary positive scalar. The following holds:*

$i)$ $\mathcal{X}_\varepsilon = \widehat{\mathcal{X}} = \operatorname{dom}(\hat{J}^*) \supseteq \mathcal{X} = \operatorname{dom}(J^*)$.

$ii)$ *We have that* (4.8) *attains its minimum $\forall x \in \mathcal{X}_\varepsilon$, and that given any $x \in \mathcal{X}$*

$$(u^*(x), t^*(x)) \in \arg\min_{(u,t)} \{t \mid (x,u,t) \in \mathcal{Z}_\varepsilon\} \Rightarrow u^*(x) \in \varepsilon\text{-}\arg\min_{u \in U(x)} c^T u$$

$iii)$ *The function $V_\varepsilon^*(\cdot)$ is continuous and piecewise affine on $\mathcal{X}_\varepsilon$.*

$iv)$ *There exists a minimizer function $u^* : \mathcal{X}_\varepsilon \to \mathbb{R}^m$ that is continuous and PWA.*

PROOF:

$i)$ That $\widehat{\mathcal{X}} \supseteq \mathcal{X}$ follows from $\operatorname{Proj}_x(\operatorname{cl}(\mathcal{Z})) \supseteq \operatorname{Proj}_x \mathcal{Z}$ and

$$\left\{x \mid \hat{J}^*(x) > -\infty\right\} = \{x \mid J^*(x) > -\infty\}$$

(recall that $J^*(x) = \infty$ if $x \notin \mathcal{X}$), hence,

$$\begin{aligned} \widehat{\mathcal{X}} &= \operatorname{Proj}_x(\operatorname{cl}(\mathcal{Z})) \cap \left\{x \mid \hat{J}^*(x) > -\infty\right\} \\ &\supseteq \operatorname{Proj}_x \mathcal{Z} \cap \{x \mid J^*(x) > -\infty\} = \mathcal{X}. \end{aligned}$$

That $\mathcal{X}_\varepsilon = \widehat{\mathcal{X}}$ holds trivially by noting that fixing $t = 0$ and $\varepsilon = 0$ for all $x \in \widehat{\mathcal{X}}$ renders any minimizer $u^*(\cdot)$ of (4.7) feasible also for (4.8).

$ii)$ Given any $x \in \mathcal{X}_\varepsilon$, (4.8) is an LP and consequently attains its minimum if the infimum is bounded, which is true on $\mathcal{X}_\varepsilon$ by construction. Lemma 4.1 and Assumption 4.1 ensures that there always exists ε-optimal solutions, i.e.

$$\forall x \in \mathcal{X} \; \exists \tilde{u} \in \varepsilon\text{- arg} \min_{u \in U(x)} c^T u.$$

Thus, for any $x \in \mathcal{X}$ and for all $\tilde{u} \in \varepsilon\text{- arg} \min_{u \in U(x)} c^T u$ there exists some $\gamma(\tilde{u}) < 0$ such that

$$\emptyset \neq \left\{ u \;\middle|\; \begin{array}{rcl} Au + Bx & \leq & d \\ Eu + Fx & \leq & g + \mathbf{1}\gamma(\tilde{u}) \\ c^T u & \leq & \hat{J}^*(x) + \varepsilon \end{array} \right\} \subseteq U(x),$$

which immediately implies $t^*(x) < 0$ for all $x \in \mathcal{X}$, and consequently $u^*(x) \in \varepsilon\text{- arg} \min_{u \in U(x)} c^T u$ for all $x \in \mathcal{X}$, for all

$$(u^*(x), t^*(x)) \in \arg \min_{(u,t)} \{t \mid (x, u, t) \in \mathcal{Z}_\varepsilon\}.$$

$iii)$ Define a new parameter y and write (4.8) as

$$\bar{V}_\varepsilon^*(x, y) := \min_{(\bar{u}, \bar{t})} \left\{ \bar{t} \;\middle|\; (x, y, \bar{u}, \bar{t}) \in \bar{\mathcal{Z}}_\varepsilon \right\}, \tag{4.10a}$$

$$\bar{\mathcal{Z}}_\varepsilon := \left\{ (x, y, \bar{u}, \bar{t}) \;\middle|\; \begin{array}{rcl} A\bar{u} + Bx & \leq & d \\ E\bar{u} + Fx & \leq & g + \mathbf{1}\bar{t} \\ c^T \bar{u} & \leq & y + \varepsilon \end{array} \right\}. \tag{4.10b}$$

Clearly, the above problem is a pLP, and consequently, $\bar{V}_\varepsilon^*(\cdot, \cdot)$ is continuous and piecewise affine. Defining $y = \hat{J}^*(x)$ we see that $\bar{V}_\varepsilon^*(\cdot, \hat{J}^*(\cdot))$ is a composition of continuous functions and therefore also a continuous function. Moreover, composition of PWA functions is a PWA function. We clearly also have $V_\varepsilon^*(x) = \bar{V}_\varepsilon^*(x, \hat{J}^*(x))$ for all $x \in \mathcal{X}_\varepsilon$.

$iv)$ Following the same argument as in $iii)$ we have that there exists a solution $(\bar{u}^*(\cdot, \cdot), \bar{t}^*(\cdot, \cdot))$ to (4.10) such that $\bar{u}^*(\cdot, \cdot)$ is continuous and PWA on its domain, and hence the same holds for $u^*(\cdot) = \bar{u}^*(\cdot, \hat{J}^*(\cdot))$.

□

The pLP is first solved over the closure of $\mathcal{Z}$ to obtain the function $\hat{J}^*(\cdot)$. Then solving (4.8) ensures that we obtain a function $u^*(\cdot)$ such that $c^T u^*(x) \leq J^*(x) + \varepsilon$ for all $x \in \mathcal{X}$.

One important detail that should be emphasized is that we can restrict any solution (selection) $u^*(\cdot)$ to (4.8) to $\mathcal{X}$, which is possible since $\mathcal{X}_\varepsilon \supseteq \mathcal{X}$. Thus, we let $u^* : \mathcal{X}_\varepsilon \to \mathbb{R}^m$ be redefined to $u^* : \mathcal{X} \to \mathbb{R}^m$. The restriction of the domain is a *key point* in the procedure because in subsequent sections we want to apply this procedure to the minimization of discontinuous piecewise affine functions, thus slightly enlarging the domain may introduce arbitrarily large errors if we try to select the minimum of several affine functions (this can be seen by comparing Figure 4.1(b) and 4.2(a) and noting that at $x = 5$ the value functions are different). In the sequel, $(u^*_\varepsilon(\cdot), t^*_\varepsilon(\cdot))$ will denote a continuous and optimal selection for (4.8), whose domain is restricted to $\mathcal{X}$.

4.4 ε-Optimal solutions for PWA functions

In the previous section we proposed a procedure for obtaining ε-optimal solutions to pLPs with strict and non-strict inequalities. In this section the procedure is repeatedly applied for the purpose of finding ε-optimal solutions to minimization of PWA functions over polygonic sets.

Consider the problem of minimizing $f(x, \cdot)$, where f is piecewise affine. We will represent f in the following manner:

$$f(x, u) = f_i(x, u) \quad \text{if} \quad (x, u) \in P_i \subseteq \mathbb{R}^n \times \mathbb{R}^m,$$

where $i \in \{1, 2, \ldots, I\}$, each f_i is affine and each P_i is a polyhedron, thus the domain of f is the polygon $\mathcal{P} := \operatorname{dom}(f) = \cup_{i=1}^{I} P_i$. Note that this implies that for each pair $(i, j) \in \{1, 2, \ldots, I\} \times \{1, 2, \ldots, I\}$ for which $P_i \cap P_j \neq \emptyset$ we have $f_i(x, u) = f_j(x, u), \forall (x, u) \in P_i \cap P_j$.

Consider the following optimization problem:

$$J^*(x) := \inf_u \{f(x, u) \mid (x, u) \in \mathcal{P}\}. \tag{4.11}$$

We can clearly represent (4.11) as

$$J^*(x) := \min_{i \in \{1,2\ldots,I\}} \inf_u \{f_i(x, u) \mid (x, u) \in P_i\}. \tag{4.12}$$

Observing that $J^{i*}(x) := \inf_u \{f_i(x, u) \mid (x, u) \in P_i\}$ is a pLP with strict and non-strict inequalities for each $[i \in \{1, 2, \ldots, I\}$ we let

$$\left\{(u^*_{i,\varepsilon}(\cdot), t^*_{i,\varepsilon}(\cdot)) \mid i \in \{1, 2, \ldots, I\}\right\}$$

denote a set of continuous selections where each pair $(u^*_{i,\varepsilon}(\cdot), t^*_{i,\varepsilon}(\cdot))$ optimizes the corresponding piecewise pLP defined by (4.8). Recall also from the previous section that the domain of each $u^*_{i,\varepsilon}(\cdot)$ is restricted to the domain of $J^{i*}(\cdot)$, and hence, $\operatorname{dom}\left(f_i(\cdot, u^*_{i,\varepsilon}(\cdot))\right) = \operatorname{dom}\left(J^{i*}\right)$.

Theorem 4.3 *Consider the optimization problem given in* (4.11).

$i)$ *For any* $\varepsilon > 0$, $x \in \operatorname{dom}(J^*)$ *and the problem*

$$J^*_\varepsilon(x) := \min_{i \in \{1,\ldots,I\}} f_i(x, u^*_{i,\varepsilon}(x)), \tag{4.13}$$

we have that

$$j \in \arg \min_{i \in \{1,\dots,I\}} f_i(x, u^*_{i,\varepsilon}(x)) \Rightarrow u^*_{j,\varepsilon}(x) \in \varepsilon\text{-}\arg\min_u \{ f(x,u) \mid (x,u) \in \mathcal{P} \} .$$

$ii)$ $\operatorname{dom}(J^*_\varepsilon) = \operatorname{dom}(J^*)$.

PROOF:

$i)$ Since $\forall x \in \operatorname{dom}(J^{i*})$ and $\forall i \in \{1, 2, \dots, I\}$ we have

$$u^*_{i,\varepsilon}(x) \in \varepsilon\text{-}\arg\min_u \{ f_i(x,u) \mid (x,u) \in P_i \} ,$$

the assertion trivially holds.

$ii)$ This follows by construction; the domain for each $u^*_{i,\varepsilon}(\cdot)$ is restricted to the domain of J^{i*}, hence the domain of J^*_ε is equal to $\cup_{i=1}^I \operatorname{dom}(J^{i*})$, which is precisely the domain of J^*.

□

We revisit Example 4.1 to illustrate the proposed procedure.

Example 4.2 (Example 4.1 cont.) Consider again the problem of minimizing (4.1a). Although this problem is simple, we formulate the individual pLPs for the purpose of illustrating the procedure. The three pLPs are (where the sets $P_i, i = 1, 2, 3$, are defined in Example 4.1):

$$\begin{aligned} J^{1*}(x) &= \inf_u \{ 2 \mid (x,u) \in P_1 \} , \\ J^{2*}(x) &= \inf_u \{ -u + x \mid (x,u) \in P_2 \} , \\ J^{3*}(x) &= \inf_u \{ -2 \mid (x,u) \in P_3 \} . \end{aligned}$$

Clearly, we only need to use the proposed procedure on the second problem. We solve the pLP over the closure of P_2:

$$\bar{J}^{2*}(x) := \min_u \{ -u + x \mid (x,u) \in \operatorname{cl}(P_2) \} . \tag{4.14}$$

Hence, a solution to (4.14) is $u^*_2(x) = 0$ for all $x \in [-5, 5]$, and $\bar{J}^{2*}(x) = x$. We then continue by finding the ε-optimal solution:

$$V^*_\varepsilon(x) = \min_{(u,t)} \{ t \mid (x,u,t) \in \mathcal{Z}_\varepsilon \} , \tag{4.15a}$$

$$\mathcal{Z}_\varepsilon = \left\{ (x,u,t) \;\middle|\; \begin{array}{rcl} -5 \le & x & \le 5 \\ u & \le & t \\ -u + x & \le & x + \varepsilon \end{array} \right\} = \left\{ (x,u,t) \;\middle|\; \begin{array}{rcl} -5 \le & x & \le 5 \\ u & \le & t \\ -u & \le & \varepsilon \end{array} \right\} , \tag{4.15b}$$

and a solution to (4.15) is $u^*(x) = -\varepsilon$ and hence $f_2(x, u^*(x)) = x + \varepsilon$ with $\mathrm{dom}\,(u^*) = \{x \mid -5 \leq x \leq 5\}$. The final step is to construct $J^*_\varepsilon(\cdot)$, which clearly becomes

$$J^*_\varepsilon(x) = \begin{cases} \min\{2, x+\varepsilon\} & \text{if} \quad -5 \leq x \leq 5 \\ -2 & \text{if} \quad\;\; 5 < x \leq 7 \end{cases}$$

and one selection $u^*_\varepsilon(\cdot)$ that fulfills $f(x, u^*_\varepsilon(x)) = J^*_\varepsilon(x)$ is

$$u^*_\varepsilon(x) = \begin{cases} -\varepsilon & \text{if} \quad -5 \leq x < 2-\varepsilon \\ 0 & \text{if} \quad\;\; 2-\varepsilon \leq x \leq 7 \end{cases}$$

The value function $J^*_\varepsilon(\cdot)$ is depicted in Figure 4.3. It is clear that we have obtained a feasible $u^*_\varepsilon(\cdot)$ such that $f(x, u^*_\varepsilon(x)) \leq J^*(x) + \varepsilon$ for all $x \in \mathrm{dom}\,(J^*)$. In addition, it is worth pointing out that by utilizing the procedure of introducing small gaps in the domain of f, as described in Example 4.1, it is not straightforward to compute an error bound, especially if the procedure is repeatedly used in a dynamic programming recursion, as will be done in Section 4.6.2.

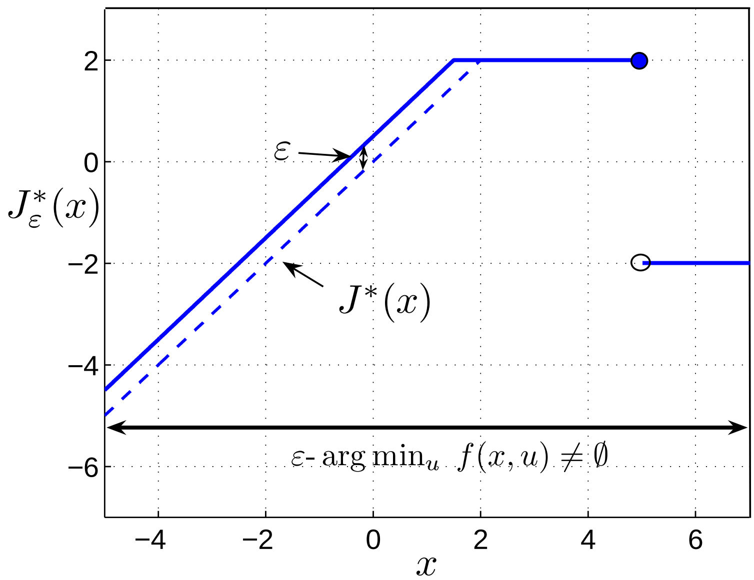


Figure 4.3: The solid line depicts $J^*_\varepsilon(\cdot)$. and the dotted line depicts $J^*(\cdot)$. Clearly the error is less than ε everywhere and no gaps are introduced in the domain of the function, cf. Figures 4.2(a) and 4.2(b).

■

4.5 ε-optimal solutions to inf-sup of PWA functions

It is apparent from the two preceding sections that a pLP with strict and non-strict inequalities can be viewed as a sub-problem of minimizing a PWA function over polygonic constraints. In this section we extend the approach to $\inf$ - $\sup$ problems and now minimization of PWA functions become our sub-problems.

Consider the problem

$$J^*(x) := \inf_{u \in \mathcal{U}(x)} \sup_{w \in \mathcal{W}(x,u)} f(x,u,w) \tag{4.16}$$

where again we consider the system where f is PWA and defined on the polygon

$$\mathcal{P} := \cup_{i=1}^{I} P_i, \quad P_i \subseteq \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p, \forall i \in \{1, 2 \ldots, I\}.$$

The set $\mathcal{U}(\cdot)$ is defined by

$$\begin{aligned}\mathcal{Z} &:= \{(x,u) \mid (x,u) \in \mathcal{Y}, (x,u,w) \in \Pi \; \forall w \in \mathcal{W}(x,u)\}, \\ \mathcal{U}(x) &:= \{u \mid (x,u) \in \mathcal{Z}\},\end{aligned}$$

where $\mathcal{Y}$, Π, and $\operatorname{gph}(\mathcal{W})$ are non-empty polygons, and $\mathcal{W}(x,u) \neq \emptyset$ for all $(x,u) \in \mathcal{Y}$. The reader is referred to (Raković, Kerrigan, Mayne, and Lygeros 2006) for details on how to compute $\mathcal{Z}$. Define also the sets

$$\begin{aligned}\mathcal{X} &:= \operatorname{Proj}_x \mathcal{Z} = \{x \mid \exists u \text{ s.t. } (x,u) \in \mathcal{Z}\}, \\ \Gamma &:= \{(x,u,w) \mid w \in \mathcal{W}(x,u)\} = \operatorname{gph}(\mathcal{W}) =: \cup_{j=1}^{M} \Gamma_j.\end{aligned}$$

The problem (4.16) can be divided into a supremum and an infimum problem as

$$V^*(x,u) := \sup_w \{f(x,u,w) \mid (x,u,w) \in \Gamma\}, \quad \forall (x,u) \in \mathcal{Z}, \tag{4.17a}$$

$$J^*(x) = \inf_u \{V^*(x,u) \mid (x,u) \in \mathcal{Z}\}, \quad \forall x \in \mathcal{X}. \tag{4.17b}$$

In this section we view problem (4.16) from a game theoretic point in the sense that we choose u and our adversary chooses w. We are therefore not concerned with attaining a maximizing w, but only a minimizing u. Define the polygon

$$\mathcal{F} := \cup_{h=1}^{H} F_h$$

where the polyhedra $\{F_1, F_2, \ldots, F_H\}$ covers the set $\Gamma \cap \mathcal{P}$, and each polyhedron F_h is a member of the set

$$\{\Gamma_j \cap P_i \mid (i,j) \in \{1, 2, \ldots, I\} \times \{1, 2, \ldots, M\}, \Gamma_j \cap P_i \neq \emptyset\}.$$

Hence, we can restrict our PWA function f to the domain for which Assumption 4.1 is valid by defining:

$$f(x, u.w) = z_h(x,u,w) \quad \text{if} \quad (x,u,w) \in F_h,$$

where $z_h(x,u,w) := f_i(x,u,w)$ if $(x,u,w) \in P_i$.

For each $h \in \{1,2,\dots,H\}$ define the pLPs:

$$\hat{V}_h^*(x,u) := \max_w \{z_h(x,u,w) \mid (x,u,w) \in \mathrm{cl}(F_h)\}, \tag{4.18a}$$

$$V_h^*(x,u) := \sup_w \{z_h(x,u,w) \mid (x,u,w) \in F_h\}. \tag{4.18b}$$

Theorem 4.4 *The following holds for all* $h \in \{1,2\dots,H\}$*:*

$$V_h^*(x,u) = \hat{V}_h^*(x,u), \quad \forall (x,u) \in \mathrm{dom}(V_h^*).$$

Moreover,

$$V^*(x,u) = \max_{h\in\{1,2,\dots,H\}} \hat{V}_h^*|_{\mathrm{dom}(V_h^*)}(x,u).$$

PROOF: The first assertion holds trivially from the fact that (4.18a) is a pLP and from the equality of the supremum and maximum over respectively F_h and $\mathrm{cl}(F_h)$, cf. Lemma 4.2 and Corollary 4.1. Having the first statement established automatically ensures that the second assertion is correct, since, for each $h \in \{1,2,\dots,H\}$, we restrict $\hat{V}_h^*(\cdot)$ to the domain of $V_h^*(\cdot)$.

□

Since we now have an exact expression for $V^*(\cdot)$, we can now apply the procedure from the previous section for the purpose of obtaining an ε-optimal solution to our problem. Recalling that $V^*(\cdot)$ is PWA and defined on a polygon $\mathcal{R} = \cup_{k=1}^K R_k$, that is,

$$V^*(x,u) = V_k^*(x,u) \quad \text{if} \quad (x,u) \in R_k,$$

then $J_\varepsilon^*(\cdot)$ is defined as

$$J_\varepsilon^*(x) := \min_{k\in\{1,2,\dots,K\}} V_k^*(x, u_{k,\varepsilon}^*(x)). \tag{4.19}$$

Theorem 4.5 $J_\varepsilon^*(x) \le J^*(x) + \varepsilon$, $\forall x \in \mathrm{dom}(J^*)$*, and*

$$\varepsilon\text{-}\arg\min_u \{V^*(x,u) \mid (x,u) \in \mathcal{Z}\} \neq \emptyset, \quad \forall x \in \mathrm{dom}(J^*).$$

PROOF: Both statements are confirmed by construction and consequences of Theorems 4.2, 4.3 and 4.4.

□

4.6 ε-optimal solutions to inf-sup optimal control of PWA systems

In this section we use the results of the previous sections to obtain ε-optimal solutions to robust optimal control problems for discontinuous PWA systems subject to state- and input-dependent disturbances. We recall the problem setup from (Kerrigan and Mayne 2002), which was extended in (Raković, Kerrigan, and Mayne 2004) to handle state- and input-dependent disturbances.

4.6.1 Problem setup

Consider the discrete-time system of the form:

$$x^+ = g(x, u, w),$$

where x is the state, x^+ is the successor state, u is the input, $g(\cdot)$ is assumed piecewise affine on the polygon $\mathcal{P}$, and $w \in \mathcal{W}(x, u) \subseteq \mathbb{R}^p$ is a time-varying disturbance. The state and input are subject to constraints $(x, u) \in \mathcal{Y} \subseteq \mathbb{R}^n \times \mathbb{R}^m$, where we assume that $\mathcal{Y}$ is a polygon. The constraints define the set-valued map

$$\mathcal{U}(x) := \{u \mid (x, u) \in \mathcal{Y}\}. \tag{4.20}$$

Let $\pi := \{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ denote a control policy (i.e. $\mu_i : \mathbb{R}^n \to \mathbb{R}^m$) over the horizon N and let $\mathbf{w} := \{w_0, w_1, \ldots, w_{N-1}\}$ denote a sequence of disturbances. Moreover, let $\phi(i; x, \pi, \mathbf{w})$ denote the solution to $x^+ = g(x, u, w)$ at time-step i for the initial state x, control policy π and disturbance sequence $\mathbf{w}$.

The cost is defined as

$$J_N(x, \pi, \mathbf{w}) := J_f(x_N) + \sum_{i=0}^{N-1} \ell(x_i, u_i)$$

where $x_i := \phi(i; x, \pi, \mathbf{w})$ and $u_i := \mu_i(x_i)$, $\forall i$. The stage cost $\ell(\cdot)$ and terminal cost $J_f(\cdot)$ are assumed to be PWA ($p \in \{1, \infty\}$):

$$\begin{aligned} \ell(x, u) &:= \|Qx\|_p + \|Ru\|_p, \\ J_f(x) &:= \|Px\|_p, \end{aligned}$$

where P, Q, and R are suitably defined weighting matrices.

The optimal control problem considered is given by

$$\mathbb{P}_N(x) : \; J_N^*(x) := \inf_{\pi \in \Pi_N(x)} \sup_{\mathbf{w} \in \mathbf{W}(x, \pi)} J_N(x, \pi, \mathbf{w}), \tag{4.21}$$

where the set of admissible disturbance sequences is given by

$$\mathbf{W}(x, \pi) := \{\mathbf{w} \mid w_i \in \mathcal{W}(x_i, u_i), \; i = 0, 1, \ldots, N-1\},$$

and the set of admissible control polices is

$$\Pi_N(x) := \{\pi \mid (x_i, u_i) \in \mathcal{Y}, \; i = 0, 1, \ldots, N-1, \; x_N \in X_f, \; \forall \mathbf{w} \in \mathbf{W}(x, \pi)\},$$

where X_f is some non-empty polygonic target set. In the sequel, we denote by X_N the set of initial states for which there exists an admissible control policy, i.e.

$$X_N := \{x \mid \Pi_N(x) \neq \emptyset\}.$$

In addition we make the following assumptions in order to ensure that $\mathbb{P}_N(x)$ is well defined for all $x \in X_N$:

A1: The function $g : \mathcal{P} \to \mathbb{R}^n$ is PWA on the polygon $\mathcal{P}$.

A2: The sets $\mathcal{Y}$ and X_f are non-empty polygons.

A3: For all $(x, u) \in \mathcal{Y}$, the set $\mathcal{W}(x, u)$ is non-empty.

A4: The graph of $\mathcal{W}$ is a non-empty polygon.

A5: The value function $J_N^*(\cdot)$ is bounded below on X_N.

Thus, in comparison to (Raković, Kerrigan, and Mayne 2004) several assumptions are relaxed (note that we use the definition in (Rockafellar and Wets 1998) for continuity of a set-valued map):

$i)$ We do not assume that $g(\cdot)$ is continuous.

$ii)$ $\mathcal{Y}$ and X_f are not required to have the origin in the interior.

$iii)$ The set-valued map $x \mapsto \mathcal{U}(x)$ is not required to be continuous and bounded on bounded sets.

$iv)$ The set-valued map $(x, u) \mapsto \mathcal{W}(x, u)$ is not required to be continuous.

$v)$ The solution to $\mathbb{P}_N(x)$ is not assumed to exist $\forall x \in X_N$.

It should be noted that in (Raković, Kerrigan, and Mayne 2004) the majority of the assumptions above are made for the purpose of being able to directly apply the topological results in (Raković, Kerrigan, Mayne, and Lygeros 2006).

4.6.2 Sub-optimal solution via dynamic programming.

The procedure for obtaining approximate solutions to (5.1), under assumption A1-A5, will be presented in this section. The results in (Raković, Kerrigan, Mayne, and Lygeros 2006) reveal geometric structure that lead to the possibility to apply a dynamic programming approach to the above mentioned problem. We recall the dynamic programming method from (Raković, Kerrigan, and Mayne 2004).

Dynamic programming provides a recursive procedure for computing sequentially the partial return functions $J_j^*(\cdot)$ (defined in (5.1) with $N = j$), the associated set-valued control laws $\kappa_j(\cdot)$ as well as their domains (here j denotes 'time-to-go' so that $\kappa_j(\cdot) = \mu_{N-j}^*(\cdot)$ if $j \in \{1, \ldots, N\}$ where $\mu_j(\cdot)$ is as defined in the previous section). The domain of $J_j^*(\cdot)$ and $\kappa_j(\cdot)$ is X_j, the set of states that can be robustly steered to the terminal set X_f in j steps or less. Define also

$$g(x, u, \mathcal{W}(x, u)) := \{g(x, u, w) \mid w \in \mathcal{W}(x, u)\}.$$

The solution to $\mathbb{P}_N(x)$ may be obtained as follows. For all $j \in \{1, 2, \ldots\}$, j denotes "time-to-go", and the partial return function $J_j^*(\cdot)$, the set-valued control law $\kappa_j(\cdot)$, and the controllability set X_j are given by:

$$J_j^*(x) = \inf_{u\in\mathcal{U}(x)} \sup_{w\in\mathcal{W}(x,u)} \left\{ \tilde{J}_{j-1}(x,u,w) \,\middle|\, g(x,u,\mathcal{W}(x,u)) \subseteq X_{j-1} \right\}, \ \forall x \in X_j, \tag{4.22a}$$

$$\tilde{J}_{j-1} := \ell(x,u) + J_{j-1}^*(g(x,u,w)), \tag{4.22b}$$

$$\kappa_j(x) = \arg\min_{u\in\mathcal{U}(x)} \left\{ \ell(x,u) + J_{j-1}^*(x^+) \,\middle|\, g(x,u,\mathcal{W}(x,u)) \subseteq X_{j-1} \right\}, \tag{4.22c}$$

$$X_j = \{x \mid \exists u \in \mathcal{U}(x) \ \text{ s.t. } \ g(x,u,\mathcal{W}(x,u)) \subseteq X_{j-1}\}, \tag{4.22d}$$

with boundary conditions

$$J_0^*(x) = J_f(x), \qquad X_0 = X_f. \tag{4.22e}$$

The conditions $g(x,u,\mathcal{W}(x,u)) \subseteq X_{j-1}$ *and* $u \in \mathcal{U}(x)$ in (4.22) may be expressed as

$$(x,u) \in \Sigma_j := \{(x,u) \in \mathcal{Y} \mid g(x,u,w) \in X_{j-1} \quad \forall w \in \mathcal{W}(x,u)\},$$

in which case X_j can be interpreted as the projection of the set Σ_j, i.e.

$$X_j = \{x \mid \exists u \ \text{ s.t. } \ (x,u) \in \Sigma_j\}.$$

The reader is referred to (Raković, Kerrigan, Mayne, and Lygeros 2006) for details on how to compute Σ_j. In order to analyze $\mathbb{P}_N(x)$, it is convenient to introduce the functions $V_j^*(\cdot)$, $j = 1, 2, \ldots, N-1$, defined by

$$V_j^*(x,u) := \sup_{w\in\mathcal{W}(x,u)} J_j^*(g(x,u,w)).$$

Note that we are interested in values of the functions $\{V_j^*(\cdot)\}_{j=1}^{N-1}$ and the sets $\{\Sigma_j\}_{j=1}^{N-1}$. The recursion (4.22a)-(4.22d) may therefore be rewritten as

$$V_{j-1}^*(x,u) = \sup_{w} \left\{ J_{j-1}^*(g(x,u,w)) \mid w \in \mathcal{W}(x,u) \right\} \quad \forall (x,u) \in \Sigma_j, \tag{4.23a}$$

$$J_j^*(x) = \inf_{u} \left\{ \ell(x,u) + V_{j-1}^*(x,u) \mid (x,u) \in \Sigma_j \right\} \quad \forall x \in X_j, \tag{4.23b}$$

$$\kappa_j(x) = \arg\min_{u} \left\{ \ell(x,u) + V_{j-1}^*(x,u) \mid (x,u) \in \Sigma_j \right\} \quad \forall x \in X_j, \tag{4.23c}$$

$$\Sigma_j = \{(x,u) \in \mathcal{Y} \mid g(x,u,\mathcal{W}(x,u)) \subseteq X_{j-1}\}, \tag{4.23d}$$

$$X_j = \{x \mid \exists u \ \text{ s.t. } \ (x,u) \in \Sigma_j\}. \tag{4.23e}$$

For each j we propose to use the approximate procedure presented in the previous sections (Theorems 4.2–4.4) when solving (4.23b) when the set $\kappa_j(\cdot)$ (4.23c) is empty. Thus, for each j for which $\kappa_j(x) = \emptyset$ for some $x \in X_j$, we compute the approximation $\kappa_{j,\varepsilon}(\cdot)$, that is, $\kappa_{j,\varepsilon}(\cdot)$ is a *selection* from the set-valued map $x \mapsto \varepsilon\text{-}\arg\min_u \left\{ \ell(x,u) + V_{j-1}^*(x,u) \mid (x,u) \in \Sigma_j \right\}$.

Two approaches are natural when considering the dynamic programming recursion; the first is the one outlined above, namely using the exact expressions for the functions $\{V_j^*(\cdot)\}_{j=0}^{N-1}$ and $\{J_j^*(\cdot)\}_{j=0}^{N}$ and use Theorems 4.2–4.4 to compute $\{\kappa_{j,\varepsilon}(\cdot)\}_{j=1}^{N}$. The second approach is to use the approximate value function in the dynamic programming recursion:

$$\begin{aligned}
V_{j-1,\varepsilon}^*(x,u) &= \sup_w \left\{ J_{j-1,\varepsilon}^*(g(x,u,w)) \mid w \in \mathcal{W}(x,u) \right\}, \quad \forall (x,u) \in \Sigma_j, \\
\kappa_{j,\varepsilon}(x) &\in \varepsilon\text{-}\arg\min_u \left\{ \ell(x,u) + V_{j-1,\varepsilon}^*(x,u) \mid (x,u) \in \Sigma_j \right\}, \quad \forall x \in X_j, \\
J_{j,\varepsilon}^*(x) &= \ell(x,\kappa_{j,\varepsilon}(x)) + V_{j-1,\varepsilon}^*(x,\kappa_{j,\varepsilon}(x)), \quad \forall x \in X_j.
\end{aligned}$$

With this approach an error bound is easily derived, as demonstrated by the following theorem:

Theorem 4.6 $J_{N,\varepsilon}^*(x) \leq J_N^*(x) + N\varepsilon.$

PROOF:

We verify the induction base by carrying out the first iteration of the dynamic programming recursion:

$$\begin{aligned}
V_0^*(x,u) &= \sup_w \{ J_0^*(g(x,u,w)) \mid w \in \mathcal{W}(x,u) \}, \quad \forall (x,u) \in \Sigma_1, \\
\kappa_{1,\varepsilon}(x) &\in \varepsilon\text{-}\arg\min_u \{ \ell(x,u) + V_0^*(x,u) \mid (x,u) \in \Sigma_1 \}, \quad \forall x \in X_1, \\
J_{1,\varepsilon}^*(x) &= \ell(x,\kappa_{1,\varepsilon}(x)) + V_0^*(x,\kappa_{1,\varepsilon}(x)) \leq J_1^*(x) + \varepsilon.
\end{aligned}$$

Assuming that the bound holds for $N = j$, i.e.

$$J_{j,\varepsilon}^*(x) \leq J_j^*(x) + j\varepsilon,$$

we must verify that the bound also holds for $N = j+1$. We get:

$$\begin{aligned}
V_{j,\varepsilon}^*(x,u) &= \sup_w \left\{ J_{j,\varepsilon}^*(g(x,u,w)) \mid w \in \mathcal{W}(x,u) \right\} \\
&\leq \sup_w \left\{ J_j^*(g(x,u,w)) \mid w \in \mathcal{W}(x,u) \right\} + j\varepsilon \\
&= V_j^*(x,u) + j\varepsilon, \quad \forall (x,u) \in \Sigma_{j+1},
\end{aligned}$$

and

$$\kappa_{j+1,\varepsilon}(x) \in \varepsilon\text{-}\arg\min_u \left\{ \ell(x,u) + V_{j,\varepsilon}^*(x,u) \mid (x,u) \in \Sigma_{j+1} \right\},$$

and since

$$\begin{aligned}
&\inf_u \left\{ \ell(x,u) + J_{j,\varepsilon}^*(x,u) \mid (x,u) \in \Sigma_{j+1} \right\} \\
&\quad \leq \overbrace{\inf_u \left\{ \ell(x,u) + V_j^*(x,u) \mid (x,u) \in \Sigma_{j+1} \right\}}^{J_{j+1}^*(x)} + j\varepsilon
\end{aligned}$$

we get

$$J^*_{j+1,\varepsilon}(x) = \ell(x, \kappa_{j+1,\varepsilon}(x)) + V^*_{j,\varepsilon}(x, \kappa_{j+1,\varepsilon}(x)) \leq J^*_{j+1}(x) + (j+1)\varepsilon. \quad \square$$

With this approach it is clear that the degree of sub-optimality can be specified *a priori* by choosing ε.

4.7 Conclusion

A method for obtaining approximate solutions to robust optimal control of discontinuous PWA systems has been presented. This was achieved by repeatedly applying a procedure, which obtained ε-optimal solutions to pLPs with strict and non-strict inequality constraints, in a dynamic programming approach. It has been demonstrated that ε-optimal solutions exists under mild assumptions, a bound on the total error for the approximate dynamic programming has been given and the degree of sub-optimality can be specified *a priori*.

This chapter considered the problem of computing the solution to a given finite horizon optimal control problem. Though this is an interesting and practically useful problem in itself, an interesting research question is how to modify the problem so that stability may be guaranteed if the solution were to be implemented in a receding horizon fashion, as common in MPC (Mayne, Rawlings, Rao, and Scokaert 2000).

Chapter 5

Utilizing Reachability in Point Location Problems

5.1 Introduction

Recently it has been shown that an optimal control law can be obtained in explicit form as a function of the initial state for several classes of model predictive control problems, see e.g. (Bemporad, Borrelli, and Morari 2002; Bemporad, Morari, Dua, and Pistikopoulos 2002; Seron, Goodwin, and Doná 2003; Johansen 2002; Grieder, Borrelli, Torrisi, and Morari 2004; Grieder, Kvasnica, Baotić, and Morari 2005; Raković, Kerrigan, and Mayne 2004). Once an explicit solution to a model predictive control problem is obtained, the online computation reduces to evaluating the control law. In this chapter we consider MPC problems where an optimal explicit control law is piecewise affine, that is, the state space is partitioned into a set of non-intersecting polyhedra, each associated with an affine function. The explicit version of MPC can be summarized as follows:

1. Measure the current state of the system.
2. Identify the the region of the state space partition that contains the current state (this will be referred to as the point location problem).
3. Apply the control associated with the region identified in step 2) to the plant.
4. Return to step 1).

In this chapter we are concerned with step 2) above. The existing approaches for performing step 2. (Tøndel, Johansen, and Bemporad 2003b; Jones, Grieder, and Raković 2006; Borrelli, Baotić, Bemporad, and Morari 2001) does a search over the entire state space partition at every time instant. We show that under certain assumptions on the MPC problem, this is unnecessary. By utilizing reachability analysis we are able to map an element of the state space partition one step forward in time, and hence, identifying a subset of the state space partition in which our next state is guaranteed to be contained in.

In the context of explicit MPC, we like to highlight three important factors; $i)$ the *worst case* number of arithmetic operations needed to solve the point location problem determines the smallest possible sampling rate for the system, $ii)$ the *average* number of arithmetic operations needed to solve the point location problem for a sequence of states gives an estimate of the energy usage of the device on which the algorithm for the point location problem is implemented, and $iii)$ the number of elements in the state space partition gives an estimate of the storage space required to implement the control scheme. Naturally one cannot, without some *a priori* knowledge of the state at the initialization of the control scheme, reduce the worst case number of arithmetic operations without completely altering the algorithm used to solve the point location problem. The average number, however, can be reduced at the expense of increased off-line processing and required storage space, which is the topic of this chapter.

5.2 Notation, Basic Definitions and Problem Setup

A *polyhedron* is the intersection of a finite number of closed halfspaces. A non-empty set F is a *face* of the polyhedron $P \subset \mathbb{R}^n$ if there exists a hyperplane $\{z \in \mathbb{R}^n \mid a^T z = b\}$, where $a \in \mathbb{R}^n$, $b \in \mathbb{R}$, such that $F = P \cap \{z \in \mathbb{R}^n \mid a^T z = b\}$ and $a^T z \leq b$ for all $z \in P$. We say that a set of finitely many polyhedra $\mathcal{P} := \{P^1, P^2, \ldots, P^I\}$ forms a *polyhedral cover* of a polyhedron P if $\cup_i^I P^i = P$. With some abuse of mathematical rigor, we say that a set of polyhedra $\mathcal{P} := \{P^1, P^2, \ldots, P^I\}$ is a *polyhedral partition* of a polyhedron P if and only if $\mathcal{P}$ is a polyhedral cover of P and the intersection of the relative interiors of any two members of $\mathcal{P}$ is equal to the empty set. If $f : X \to Y$ is a function, then the *restriction* of f to the domain $D \subseteq X$ is written $f|_D : D \to Y$. The convex hull operator is denoted by $\mathrm{conv}(\cdot)$. Given two sets $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^n$, the *Minkowski set addition* is defined as $X \oplus Y := \{x + y \mid x \in X, y \in Y\}$.

Consider the discrete-time system on the form:

$$x^+ = f(x, u, w),$$

where x is the (measured) state, x^+ is the successor state, u is the input, $f(\cdot)$ can take the form $Ax + Bu$, $Ax + Bu + w$, or $Ax + Bu + w$ with $[A\,B] \in \mathbb{M} := \mathrm{conv}\left([A^1\,B^1], [A^2\,B^2], \ldots, [A^J\,B^J]\}\right)$, and $w \in \mathcal{W}$ is a persistent disturbance where $\mathcal{W}$ is a polyhedron. The state and input are subject to constraints $(x, u) \in \mathcal{Y}$ where we assume that $\mathcal{Y}$ is a polyhedron. The constraints define the set-valued map $\mathcal{U}(x) := \{u \mid (x, u) \in \mathcal{Y}\}$. Let $\pi := \{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ denote a control policy over the horizon N and let $\mathbf{w} := \{w_0, w_1, \ldots, w_{N-1}\}$ denote a sequence of disturbances. Moreover, let $\phi(i; x, \pi, \mathbf{w})$ denote the solution to $x^+ = f(x, u, w)$ at iteration i for the initial state x, control policy π and disturbance sequence $\mathbf{w}$. The cost is defined as

$$V_N(x,\pi,\mathbf{w}) := V_f(x_N) + \sum_{i=0}^{N-1} l(x_i,u_i)$$

where $x_i := \phi(i;x,\pi,\mathbf{w})$ and $u_i := \mu_i(x_i)$, $\forall i$. The stage cost $l(\cdot)$ and terminal cost $V_f(\cdot)$ are assumed to be linear or quadratic. The optimal control problem considered is given by

$$\mathbb{P}(x):\ V^*(x) := \min_{\pi\in\Pi_N(x)} \max_{\mathbf{w}\in\mathbf{W}} V_N(x,\pi,\mathbf{w}) \tag{5.1}$$

where the set of admissible disturbance sequences is given by

$$\mathbf{W} := \{\mathbf{w} \mid w_i \in \mathcal{W},\ i=0,1,\ldots,N-1\},$$

and the set of admissible control polices is

$$\Pi_N(x) := \left\{ \pi \;\middle|\; \begin{array}{l} (x_i,u_i)\in\mathcal{Y},\ i=0,1,\ldots,N-1, \\ x_N \in X_f,\ \forall \mathbf{w}\in\mathbf{W} \end{array} \right\}.$$

and X_f is some polyhedral terminal constraint. Note that if $w_i = 0\ \forall i$, $\mathbb{P}(x)$ reduces to a minimization problem. In the sequel, we denote by $\mathcal{X}$ the set of initial states for which a solution to $\mathbb{P}(x)$ exists. We assume throughout this chapter that there exists a solution to $\mathbb{P}(\cdot)$ such that the function $u^* : \mathcal{X} \to \mathbb{R}^m$ defined by $u^*(x) = u_0(x)$ is piecewise affine on a polyhedral partition $\mathcal{R} := \{R^1, R^2, \ldots, R^I\}$ of $\mathcal{X}$, which is the case under some additional assumptions on the combination of $f(\cdot)$ and the norm used in the cost function, see e.g. (Bemporad, Morari, Dua, and Pistikopoulos 2002; Bemporad, Borrelli, and Morari 2002; Raković, Kerrigan, and Mayne 2004; Seron, Goodwin, and Doná 2003; Johansen 2002; Grieder, Borrelli, Torrisi, and Morari 2004; Grieder, Kvasnica, Baotić, and Morari 2005).

5.3 Point Location Problem

The point location problem can be stated as: Given a polyhedral partition $\mathcal{R} := \{R^1, R^2, \ldots, R^I\}$ of the polyhedron R and a point $x \in R$, find $k \in \{1,2,\ldots,I\}$ such that $x \in R^k$.

There exist several approaches for the purpose of solving the point location problem, the next four subsections briefly summarize the most common strategies (the reader is referred to the respective paper for details).

5.3.1 Linear search

Linear search is the most inefficient of the methods we consider, but also the most general approach for solving the point location problem:

1. let $i \leftarrow 1$.

2. if $x \in R^i$, terminate (point location problem solved).

3. let $i \leftarrow i + 1$ and return to 2).

5.3.2 Comparison of value functions

Comparison of value functions improves upon the linear search in that it is not explicitly checked if $x \in R^i$. This approach relies on that $V^*(\cdot)$ is convex, continuous, PWA and defined on a polyhedral partition $\mathcal{R} := \{R^1, R^2, \ldots, R^I\}$ of $\mathcal{X}$. The function $V^*(\cdot)$ is given by

$$V^*(x) := V^*|_{R^i}(x) \quad \text{if} \quad x \in R^i.$$

Thus, given x, an index k such that $x \in R^k$ is determined from

$$k = \arg\max_{i \in \{1,\ldots,I\}} V^*|_{R^i}(x).$$

Clearly, the complexity of this approach is linear in the cardinality of $\mathcal{R}$. The main advantages with this approach is its simplicity; that one only needs to store the value functions and optimizers, and that it requires no additional off-line computational effort.

5.3.3 Binary search tree

In (Tøndel, Johansen, and Bemporad 2003b) the authors propose a binary search tree approach for solving the point location problem. It is only assumed that $u^*(\cdot)$ is piecewise affine on a polyhedral partition $\mathcal{R}$ of $\mathcal{X}$. It is beyond the scope of this chapter to describe in detail how the tree is constructed, but the structure is as follows (see also Appendix C):

1. Each leaf node represents one polyhedral set $R \in \mathcal{R}$ and the associated control law $u^*|_R(\cdot)$.

2. Each node that is not a leaf node represents a hyperplane, $h(x) = a^T x + b$, and points to two children, c_+ and c_-. The leaf nodes in the c_+ (c_-) branch are the leaf nodes whose corresponding polyhedral set intersect

$$\{x \mid h(x) > 0\} \, (\{x \mid h(x) \leq 0\}).$$

Thus, for a given x we start at the root node, and continue with:

1. If current node is a leaf node, terminate (point location solved).

2. If $h(x) > 0$ branch to c_+, otherwise branch to c_-. Return to 1).

Compared to linear search and comparison of value functions the main advantages with the binary search tree is the low search time (the depth of the search tree is estimated to be $1.7\ \log_2 I$, where I is the cardinality of $\mathcal{R}$). In addition, no other assumption than that $u^*(\cdot)$ is piecewise affine on a polyhedral partition $\mathcal{R}$ is placed on the problem, making the method very accessible. The main drawback is that the required storage space may become large and that a large number of polyhedra in $\mathcal{R}$ may render the construction of the search tree computationally intractable.

5.3.4 Logarithmic time approach

Recall that a collection of polyhedra $\mathcal{P} := \{P^1, P^2 \ldots, P^I\}$ is called *complex* if $i)$ every face of a member of $\mathcal{P}$ is itself a member of $\mathcal{P}$ and $ii)$ the intersection of any two members of $\mathcal{P}$ is a face of each of them.

In (Jones, Grieder, and Raković 2006) it is shown that under the assumptions that $V^*(\cdot)$ is convex, continuous, and PWA on a complex $\mathcal{R} := \{R^1, R^2, \ldots, R^I\}$ that forms a polyhedral partition of $\mathcal{X}$, the point location problem can be solved in logarithmic time using a nearest neighbor search (Arya, Mount, Netanyahu, Silverman, and Wu 1998). The procedure works by intersecting a power-diagram (Aurenhammer 1991), which is a weighted Voronoi diagram, with the complex $\mathcal{R}$ and implementing a nearest neighbor search.

The main advantages of this approach compared to the methods mentioned in Sections 5.3.1-5.3.3 is the low worst case number of arithmetic operations needed to find the solution. Compared to the binary search tree the computational effort needed for preprocessing is greatly reduced, however, the requirement that state space partition must form a complex and the restrictions on the value function $V^*(\cdot)$ reduces the applicability of the method compared to the binary search tree.

5.4 Utilizing Reachability Analysis in Point location Problems

Instead of searching through the entire partition $\mathcal{R}$ at each sample one can utilize reachability analysis to reduce the number of polyhedral sets that are candidates to contain the state at the next time instant. Under the assumptions on $f(\cdot)$ and $u^*(\cdot)$, it is possible to compute a convex outer approximation of the one-step forward reach set associated with a set $R \in \mathcal{R}$, that is, a convex outer approximation of the set

$$R^+ := \{f(x, u^*|_R(x), w) \mid x \in R, w \in \mathcal{W}\},$$

where the control law $u^*(\cdot)$ is defined as

$$u^*(x) = u^*|_R(x) := K_R x + k_R \quad \text{if} \quad x \in R,\ R \in \mathcal{R}.$$

Given a set $R \in \mathcal{R}$ we can map the set R one-step forward in time yielding the set R^+. We associate with this reach set the subset $\mathcal{M}(R^+)$ of $\mathcal{R}$ defined by $\{R^i \in \mathcal{R} \mid R^+ \cap R^i \neq \emptyset\}$ and define a polyhedral partition

$$\mathcal{N}(R^+) := \{R^i \cap R^+ \mid R^i \in \mathcal{M}(R^+)\}$$

of R^+. Thus, if our previous state was contained in R, then the point location problem reduces to a search through set $\mathcal{N}(R^+)$. The off-line computation required and the online algorithm are stated in Algorithm 5.1 and 5.2, respectively.

Procedure 5.1 Offline computation.

Input: The system $x^+ = f(x, u, w)$, the constraints set $(\mathcal{Y}, \mathcal{W})$, and the PWA control law $u^*(\cdot)$.

Output: A collection $\mathcal{C}$ of pairs $(R, \mathcal{N}(R^+))$.

1: $\mathcal{C} \leftarrow \emptyset$.
2: **for** all $R \in R$ **do**
3: Compute $R^+ := \{f(x, u^*|_R(x), w) \mid x \in R, w \in \mathcal{W}\}$.
4: Compute $\mathcal{N}(R^+) := \{R^i \cap R^+ \mid R^i \in \mathcal{M}(R^+)\}$.
5: $\mathcal{C} \leftarrow \mathcal{C} \cup \{(R, \mathcal{N}(R^+))\}$.
6: **end for**

Procedure 5.2 Explicit MPC utilizing reachability.

Input: An explicit control law $u^*(\cdot)$ defined on $\mathcal{R}$, and the output $\mathcal{C}$ of Algorithm 5.1.

1: **Initialization:** Given the measured state x, solve the point location problem for $\mathcal{R}$, i.e. determine R such that $x \in R$. Apply $u^*|_R(x)$ to the plant.
2: **while** MPC algorithm is running **do**
3: Measure the state x.
4: Given R, recall $\mathcal{N}(R^+)$ from $\mathcal{C}$ and find $\bar{R} \in \mathcal{N}(R^+)$ such that $x \in \bar{R}$.
5: Apply the control $u^*|_{\bar{R}}(x)$ to the plant.
6: $R \leftarrow \bar{R}$.
7: **end while**

5.4.1 Evaluation of piecewise affine control laws for the regulation problem of deterministic systems

Consider the class of systems where

$$x^+ = f(x, u, 0) = Ax + Bu.$$

The reach set R^+ associated with R is easily found from

$$R^+ := \{Ax + Bu^*|_R(x) \mid x \in R\} = (A + BK_R)R \oplus \{Bk_R\}.$$

We see that if we use the Algorithms 5.1 and 5.2 on this problem, it resembles open loop control in the sense that we have assumed that our system is completely deterministic. To remedy this we consider the case where measurement noise and input corruption are represented by the model (model uncertainty is treated in the next subsection):

$$x^+ = Ax + Bu + w, \ w \in \mathcal{W}.$$

In this case the reach set becomes:

$$\begin{aligned} R^+ :&= \{Ax + Bu^*|_R(x) + w \mid x \in R, w \in \mathcal{W}\} \\ &= (A + BK_R)R \oplus \mathcal{W} \oplus \{Bk_R\}. \end{aligned}$$

5.4.2 Evaluation of piecewise affine control laws for the regulation problem of uncertain systems

Consider the class of systems where

$$\begin{aligned} x^+ = f(x,u,w) &= Ax + Bu + w, \\ \begin{bmatrix} A & B \end{bmatrix} \in \mathbb{M} &:= \operatorname{conv}\left(\begin{bmatrix} A^1 & B^1 \end{bmatrix}, \ldots, \begin{bmatrix} A^J & B^J \end{bmatrix}\right). \end{aligned}$$

The reach set associated with $R \in \mathcal{R}$ is given by

$$\begin{aligned} R^+ :&= \left\{ Ax + Bu^*|_R(x) + w \;\middle|\; \begin{array}{l} x \in R, \ w \in \mathcal{W} \\ \begin{bmatrix} A & B \end{bmatrix} \in \mathbb{M} \end{array} \right\} \\ &= \bigcup_{x \in R} \left\{ Ax + Bu^*|_R(x) + w \;\middle|\; \begin{array}{l} \begin{bmatrix} A & B \end{bmatrix} \in \mathbb{M} \\ w \in \mathcal{W} \end{array} \right\}. \end{aligned}$$

Noting that this set is generally non-convex, we use the convexification of R^+, i.e. $\hat{R}^+ := \operatorname{conv}(R^+)$ or any other suitable convex outer approximation. For each fixed pair $[A \ B] \in \mathbb{M}$ we have that $\{Ax + Bu^*|_R(x) + w \mid x \in R, w \in \mathcal{W}\}$ is a polyhedron, hence $\hat{R}_+$ is also a polyhedron. It is straightforward to use Algorithm 5.1 and 5.2 in the parameter uncertain case if step 3 of Algorithm 5.1 is replaced by: Compute $\hat{R}^+$.

5.4.3 Evaluation of piecewise affine control laws for set-point tracking for deterministic systems

The tracking problem is slightly different than the regulation, and for brevity we only consider the deterministic system on the form

$$x^+ = f(x, u, 0) = Ax + Bu.$$

The objective is to minimize deviation from some desired reference signal s. In this case we assume that $s \in S$, where S is a polyhedron, but that s can change arbitrarily from one sample to the next. The optimal control law is then a function of the set-point and initial state:

$$u^*(x,s) = u^*|_R(x,s) := K_R^1 x + K_R^2 s + k_R \text{ if } (x,s) \in R,$$

which is defined on a polyhedral partition $\mathcal{R}$ of the set $\mathcal{X} \times S$.

The reach set is, in this case, given by:

$$R^+ := R_X^+ \times S, \text{ where}$$
$$R_X^+ := \{Ax + Bu^*|_R(x,s) \mid (x,s) \in R\}.$$

5.5 Numerical Example

Consider a discrete-time double integrator

$$x^+ = Ax + Bu + w = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} T_s^2 \\ T_s \end{bmatrix} u + w,$$

where the sampling interval is $T_s = 0.3$. Further, let the constraints be $-1 \leq x_2 \leq 1$ and $|u| \leq 1$. An explicit control law is defined by solving an optimal control problem for the nominal case, i.e. for $w_i = 0, \ \forall i$. Figure 5.1(a) depicts the deterministic case, i.e. $\mathbf{w} = 0$ and Figures 5.1(b)-5.1(c) for non-zero disturbances. Note that the solution complex $\mathcal{R}$ contains 183 polyhedra, and that in the deterministic case no search is necessary. When disturbance is included, a search over 11 and 29 regions are required for $\|w\|_\infty \leq 0.1$ and $\|w\|_\infty \leq 0.3$, respectively, for a randomly chosen element of $\mathcal{R}$. Note also that for the case where the disturbance is limited to $\|w\|_\infty \leq 0.3$, the reduction is substantial even though, as can be seen from Figure 5.1(c), the disturbance can completely cancel the control action and therefore is unreasonably large.

5.6 Incorporating reachability in algorithms for point location problems

The worst case time to solve the point location problem is not improved by using the reachability approach unless some *a priori* knowledge about the initial state is available. It is, however, obvious that for linear search (Section 5.3.1) and comparison of value functions (Borrelli, Baotić, Bemporad, and Morari 2001) (Section 5.3.2) the average time the microprocessor use to solve the point location problem is reduced when utilizing the reachability approach, and hence, reducing energy consumption or making the processor available for other tasks. This naturally comes at the expense of increased storage and off-line computational effort. For the binary search (Tøndel, Johansen, and Bemporad 2003b) (Section 5.3.3) and logarithmic time (Jones, Grieder, and Raković 2006) (Section 5.3.4) approaches the reduction is not immediate, the algorithms either have to be modified or the extra information available must be encoded into existing data structures. Here we consider only the binary search tree approach (Tøndel, Johansen, and Bemporad 2003b) and we outline three different approaches for utilizing the extra information; mini-trees, lowest start node, and embedded trees.

5.6.1 Mini-trees

By mini-trees we mean building a small search tree for each of the sets in

$$\left\{\mathcal{N}(R^+) \mid R \in \mathcal{R}\right\}.$$

This is the most efficient approach of the three we are considering, however, both storage and off-line computational effort is greatly increased.

5.6.2 Identifying lowest start node

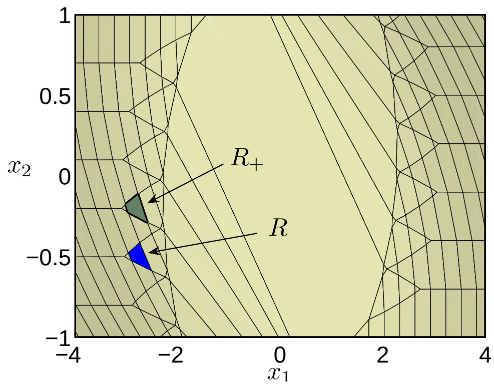
Given a one step forward reach set R^+, one can traverse the original binary search tree with respect to R^+. That is, in each node that is not a leaf node, we check if R^+ is entirely contained on either side of the hyperplane $h(\cdot)$, i.e. if $R^+ \cap \{x \mid h(x) > 0\} = \emptyset$ or $R^+ \cap \{x \mid h(x) \leq 0\} = \emptyset$. We start at the root node and traverse the tree until we reach a node n_l where the associated hyperplane intersects R^+. The node n_l is clearly the node lowest in the tree our search can start from. Thus, with each reach set we associate a node in the binary search tree that acts as the root (or initial) node. This approach requires negligible extra preprocessing and storage, but the average time to solve point location problem is not reduced as much as the with mini-trees or embedded trees.

5.6.3 Embedded trees

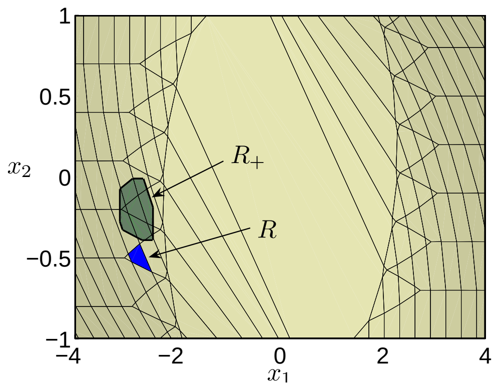
This method is a mix of the two previous approaches. At each node in the search tree a list of all the reach sets is included. Given a reach set and a node, the list indicates whether the node should be evaluated or not. If the node should be evaluated, it is performed like in Section 5.3.3. On the other hand, if there is no need to evaluate the node, the list points to the next node that should be evaluated. This approach requires negligible extra preprocessing, the average time is reduced almost as much as for the mini-trees, but the scheme requires more storage space than the approach of identifying the lowest start node.

5.7 Conclusion

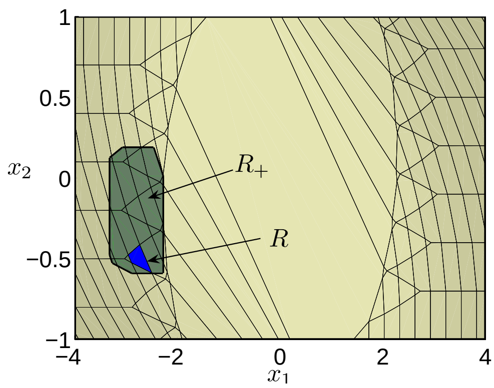
The reachability approach, under certain mild assumptions on the problem data, reduces the online computational effort needed to solve the point location problem. This reduction naturally comes at the expense of increased off-line computational effort and required storage space. Compared to the logarithmic time approach, the proposed method requires more preprocessing and storage, but is less restrictive on the assumptions on the problem data. Thus, the proposed method provides a tradeoff between evaluation speed and storage/preprocessing that can be used for problems for which the logarithmic time approach is not applicable.



(a) Set R and its reach set R^+ in deterministic case.



(b) Set R and its reach set R^+ with $\mathcal{W} = \{w \mid \|w\|_\infty \leq 0.1\}$.



(c) Set R and its reach set R^+ with $\mathcal{W} = \{w \mid \|w\|_\infty \leq 0.3\}$.

Figure 5.1: Illustration of reach sets. Notice that even with large disturbances only a small part of the partition needs to be searched to solve the point location problem.

Part III

Control Allocation

Chapter 6

Control Allocation via Parametric Programming; Thruster-Controlled Floating Platform

6.1 Introduction

The task in control allocation is to determine how to generate a specified generalized force from a redundant set of actuators and control effectors[1] where the associated controls are constrained, see e.g. (Bodson 2002; Luo, Serrani, Yurkovich, Oppenheimer, and Doman 2007; Durham 1993; Durham 1994b; Durham 1994a; Page and Steinberg 2000; Bordignon and Durham 1995b; Bordignon and Durham 1995a; Bodson and Pohlchuck 1998; Buffington, Enns, and Teel 1998; Luo and Doman 2004; Virnig and Bodden 1994; Eberhardt and Ward 1999; Johansen, Fossen, and Tøndel 2005; Johansen, Fuglseth, Tøndel, and Fossen 2007; Lindfors 1993; Johansen 2004b; Lindegaard and Fossen 2003; Johansen, Fossen, and Berge 2003; Sørdalen 1997). The main objective is to obtain the desired generalized force, however, it is also common to incorporate secondary objectives, such as minimizing power consumption, power transients and mechanical tear and wear. Several other factors, such as actuator- and control effector-dynamics (Luo and Doman 2004), can also be incorporated. One way of achieving these secondary goals is to solve a constrained optimization problem online at every sampling instant. A control allocation approach to control synthesis often has the advantage that a high level control law that is independent of actuator and effector configuration can be designed. The approach also utilizes redundancy of the effectors to obtain a fault tolerant scheme. Constrained control allocation can therefore be

[1] We distinguish between control effectors and actuators. An effector is a device that generate generalized forces (e.g. thruster) and an actuator is a device that influences the direction or size of the force (e.g. a motor). Consequently, there can be several actuators associated with one effector.

viewed as a special form of inner loop controller design, where it is common to assume that the inner loop is instantaneous.

Only recently, it has been proposed to solve the constrained optimization problem off-line (Johansen, Fossen, and Tøndel 2005; Johansen, Fuglseth, Tøndel, and Fossen 2007) by utilizing parametric programming techniques (Gal and Nedoma 1972; Bemporad, Morari, Dua, and Pistikopoulos 2002; Tøndel, Johansen, and Bemporad 2003a; Bank, Guddat, Klatte, Kummer, and Tammer 1983). For certain classes of allocation problems the online computational effort then reduces to an evaluation of a piecewise affine function, which can be formulated as a point location problem (Tøndel, Johansen, and Bemporad 2003b; Jones, Grieder, and Raković 2006; Christophersen, Kvasnica, Jones, and Morari 2007; Spjøtvold, Raković, Tøndel, and Johansen 2006). The main advantages of this approach are: i) removing the need for sophisticated optimization software on the processor, ii) the correctness of the solution can be verified off-line, which is a key issue in safety critical applications, iii) the worst case number of arithmetic operations needed to find the solution can easily be computed, iv) the average and worst case number of arithmetic operations needed to find the solution is usually greatly reduced, and v) evaluation of the PWA function can be implemented using fixed point arithmetic. The main drawbacks, on the other hand, are that i) the problem class for which this solution strategy is applicable is limited, and in cases where an exact solution can be found ii) obtaining an explicit solution may be computationally intractable and iii) the storage space required to represent the solution may exceed the available memory. The drawbacks may in particular be apparent when the system has to accommodate reconfiguration due failure situations and/or operation in several modes. However, if we are able to obtain and represent an explicit solution, it is clearly more desirable than utilizing online optimization.

In this chapter we focus on optimal thrust allocation for a scale model of a thruster-controlled floating platform that is commonly used for offshore oil drilling and production. In particular, we seek to obtain an explicit solution to the control allocation problem. The platform has eight rotatable fixed pitch azimuth thrusters and the high level controller specifies surge, sway and yaw forces. The task is to determine the thrust magnitude and azimuth angle for each thruster such that the desired surge, sway and yaw forces are generated. Each thruster can rotate 360 degrees, but the thrust magnitude is limited. In addition, it is necessary to enforce artificial constraints on the azimuth angles to avoid non-linear interaction between the thrusters.

Current non-optimization based approaches to this problem are either conservative in terms of utilizing only a limited fraction of the attainable force set[2] or not optimal in terms of power consumption. One approach is to fix the azimuth thrusters angles and use a pseudo inverse to compute the thrust magnitudes (Fossen 2002; Tyssø and Aga 2006); a method that only utilizes a limited volume of the

[2]The attainable force set is the set of generalized forces that can be generated by the control effectors (azimuth thrusters) while fulfilling the constraints.

attainable force set and may yield singular thruster configurations if thruster failure occurs. Another approach is to use a generalized inverse to compute both azimuth angles and thrust magnitudes (Sørdalen 1997). This approach is also conservative with respect to the attainable force set and not optimal with regards to power consumption. Neither of the approaches efficiently takes into account the constraints on the control inputs other than saturating the controls, and consequently an optimization based approach should be beneficial both with regards to minimizing power consumption and utilizing a larger volume of the attainable force set.

Optimization based approaches, both explicit solutions and online optimization, have been successfully tested on ships and other control allocation problems (Bodson 2002; Luo, Serrani, Yurkovich, Oppenheimer, and Doman 2007; Durham 1993; Durham 1994b; Durham 1994a; Page and Steinberg 2000; Bordignon and Durham 1995b; Bordignon and Durham 1995a; Bodson and Pohlchuck 1998; Buffington, Enns, and Teel 1998; Luo and Doman 2004; Virnig and Bodden 1994; Eberhardt and Ward 1999; Johansen, Fossen, and Tøndel 2005; Johansen, Fuglseth, Tøndel, and Fossen 2007; Lindfors 1993; Johansen 2004b; Lindegaard and Fossen 2003; Johansen, Fossen, and Berge 2003; Sørdalen 1997). However, the platform has 8 control effectors and 16 controls inputs and needs to handle thruster failure conditions, making it questionable whether obtaining an explicit solution is computationally tractable. We seek to illustrate that obtaining an explicit solution to an approximation of the control allocation problem for the platform is computationally tractable, that the explicit solution can evaluated at a high frequency, and that the performance is satisfactory even if rate constraints for the effectors are neglected.

Chapter Structure: Section 6.2 introduces basic notation and nomenclature and in Section 6.3 we recall the fundamentals of constrained control allocation. Explicit solutions to parametric linear and convex quadratic programs are treated in Section 6.4. In Section 6.5 we present the control allocation problem for the platform and experimental results.

6.2 Basic definitions and nomenclature.

For completeness we recall some standard notation and definitions. A *polyhedron* is the intersection of a finite set of open and/or closed halfspaces. For an extended real valued (i.e. allowed to take values in $\overline{\mathbb{R}} := [-\infty, \infty]$) function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$, the *domain* of f is defined as the set

$$\operatorname{dom}(f) := \{x \in \mathbb{R}^n \mid -\infty < f(x) < \infty\}.$$

Whenever we refer to a function f or mapping F having a certain property, we implicitly mean that the property holds only on the domain of f or F, e.g. if we say that f is continuous, it is continuous at every $x \in \operatorname{dom}(f)$. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is *piecewise affine* (PWA) on its domain if $\operatorname{dom}(f)$ is the union of finitely many polyhedra, relative to each of which $f(\cdot)$ is affine. If $X \subseteq \mathbb{R}^n$ and

$Y \subseteq \mathbb{R}^m$, then 2^Y is the *power set* (set of all subsets) of Y and a set-valued map is defined as $F : X \to 2^Y$. For notational simplicity, we use double arrows to specify that a mapping is set-valued, i.e. set-valued maps are specified as $F : X \rightrightarrows Y$. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is a *selection* of the set-valued map $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ if $f(x) \in F(x)$ for all $x \in \mathrm{dom}\,(F)$.

Given two sets $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^n$, the *Minkowski set addition* is defined as

$$X \oplus Y := \{x + y \mid x \in X, y \in Y\}.$$

Given the a sequence of sets $\{X_i\}_{i=a}^b$, we define $\bigoplus_{i=a}^b X_i := X_a \oplus \cdots \oplus X_b$.

Throughout we will use the superscript * to distinguish between optimizers and decision variables, e.g. for the problem $\min_x f(x)$, x is the decision variable and x^* denotes an optimizer.

6.3 Static Constrained Control Allocation.

Consider the equation

$$\bar{\tau} = g(x, u, t), \tag{6.1}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $\bar{\tau} \in \mathbb{R}^{n_\tau}$ is the generalized force, t is time and $u \in \mathbb{R}^{n_u}$ is the control. Assume further that there exists a virtual time-varying feedback controller

$$\tau := k(t, x), \tag{6.2}$$

that is, our desired generalized force is τ while the actual generalized force is $\bar{\tau}$. The task in control allocation is to determine controls u, satisfying some constraints $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$, that generate the generalized force $\bar{\tau}$ that is in some sense closest to the desired τ. When it is possible to obtain $\bar{\tau} = \tau$, it is often an uncountable number of combinations of controls that achieve the desired generalized force. Hence, in this case, secondary objectives such as minimization of power consumption and actuator tear and wear are considered.

It is common to assume a linear (possibly time-varying and state-dependent) relationship between the controls and the generalized forces (Bodson 2002; Johansen, Fossen, and Tøndel 2005; Luo, Serrani, Yurkovich, Oppenheimer, and Doman 2007; Durham 1993; Durham 1994b; Durham 1994a; Page and Steinberg 2000; Bordignon and Durham 1995b; Bordignon and Durham 1995a)

$$\bar{\tau} = B(x, t)u,$$

where $B(\cdot, \cdot) \in \mathbb{R}^{n_\tau \times n_u}$. We will consider an even simpler relationship and in the sequel we write (6.1) as

$$\bar{\tau} = Bu.$$

In this formulation it is sometimes possible to accommodate for actuator and control effector nonlinearities through a nonlinear mapping, as will be done in our application through a quadratic relation between the thruster force and propeller speed.

The simplest version of the constrained control allocation problem is to find a solution (u, s) ($s \in \mathbb{R}^{n_\tau}$ are slack variables) to the system

$$\tau + s = Bu \quad \text{and} \quad u \in \mathcal{U}, \tag{6.3}$$

such that $s = 0$ if there exists $u \in \mathcal{U}$ for which $\tau = \bar{\tau} = Bu$.

Several methods exist for the purpose of solving this problem. The most common approaches are briefly summarized in the following five subsections, but the reader is referred to (Bodson 2002) and references therein for details.

6.3.1 Generalized and cascaded generalized inverses

In the simplest approach to the linear control allocation problem the controls are constrained to $\mathcal{U} := \{u \mid \underline{u} \leq u \leq \overline{u}\}$, where the inequalities are to be interpreted componentwise. The solution procedure called the *generalized inverse* simply assumes that the controls are unconstrained, i.e. the solution becomes:

$$u^* = B^{\#}\tau$$

where $B^{\#}$ is a weighted pseudo-inverse of B given by

$$B^{\#} = H^{-1}B^T(BH^{-1}B^T)^{-1}.$$

u^* solves the problem

$$\min_u u^T H u \quad \text{s.t.} \quad Bu = \tau,$$

where $H > 0$ and B has full rank. If $u^* \notin \mathcal{U}$, then $u = \mathrm{saturate}(u^*)$ is used as the control input. The reader is referred to (Virnig and Bodden 1994; Eberhardt and Ward 1999) for details.

A simple improvement of the generalized inverse is the *cascaded generalized inverse*. The elements $\hat{u}$ of u^* that saturate for the generalized inverse are removed from the equation by letting

$$Bu = \begin{bmatrix} \tilde{B} & \hat{B} \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \hat{u} \end{bmatrix} = \tau \Rightarrow \tilde{u} = \tilde{B}^{+}(\tau - \hat{B}\hat{u}),$$

where $\tilde{B}^{+} = \tilde{B}^T(\tilde{B}\tilde{B}^T)^{-1}$. This process is repeated until either all remaining controls saturate, none saturate or the system is inconsistent. In the latter case the undetermined controls are given by the left pseudo inverse; $\tilde{u} = (\tilde{B}\tilde{B}^T)^{-1}\tilde{B}^T(\tau - \hat{B}\hat{u})$.

6.3.2 Direct allocation.

Direct allocation (Durham 1993; Durham 1994b; Durham 1994a) finds the scalar K such that K is maximized subject to $K\tau = B\hat{u}$ and $\underline{u} \leq \hat{u} \leq \overline{u}$. If $K \leq 1$, we implement $u = \hat{u}$, otherwise $u = \hat{u}/K$. This problem can be written as an LP or solved using geometric arguments when the dimension of τ is small, typically two or three.

6.3.3 Two stage optimization

Two stage optimization (Enns 1998) is divided into a *control deficiency* and a *control sufficiency* stage. In the control deficiency stage the error between the desired and implemented generalized force is minimized, that is,

$$J = \min_u \|\tau - Bu\| \quad \text{s.t.} \quad u \in \mathcal{U} := \{u \mid \underline{u} \leq u \leq \overline{u}\}.$$

If $J = 0$ in the above problem, then the control sufficiency stage is also implemented. Here the objective is to minimize a secondary objective, such as energy minimization. If we let u^* denote the solution obtained from the control deficiency stage, the control sufficiency stage may be to solve the problem

$$\min_u \|u - \hat{u}\| \quad \text{s.t.} \quad Bu = Bu^*,\ u \in \mathcal{U},$$

where $\mathcal{U} := \{u \mid \underline{u} \leq u \leq \overline{u}\}$ and $\hat{u}$ is some desired control vector.

6.3.4 Mixed optimization.

Mixed optimization is an attempt to solve both the control deficiency and sufficiency stage in one go. The problem is given by

$$J = \min_u \|\tau - Bu\| + \varepsilon\|u - \hat{u}\| \quad \text{s.t.} \quad u \in \mathcal{U},$$

where $\mathcal{U} := \{u \mid \underline{u} \leq u \leq \overline{u}\}$ and ε is a scalar defining the relative weighting between the two objectives.

6.3.5 Linear and/or Quadratic Optimization

The formulation that we will utilize in our application is given by:

$$\mathbb{P}(\tau): \ J^*(\tau) := \inf_{u,s} \{J(u,s,\tau) \mid (u,s,\tau) \in \mathcal{Z}\}, \tag{6.4a}$$

$$J(u,s,\tau) := \|Qs\|_l + \|Ru\|_l, \tag{6.4b}$$

$$\mathcal{Z} := \left\{ (u,s,\tau) \;\middle|\; \begin{array}{l} Bu + s = \tau, \\ u \in \mathcal{U} \end{array} \right\}, \tag{6.4c}$$

where $(u, s, \tau) \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_\tau} \times \mathbb{R}^{n_\tau}$, $Q \geq 0$ and $R \geq 0$ are weight matrices, respectively penalizing use of controls and deviation from desired generalized force, $\mathcal{U}$ is the constraint set on the control inputs, and $l \in \{1, 2, \infty\}$[3] denotes the weighting norm.

Remark 6.1 *Please note that it is possible to obtain an exact penalty function (Nocedal and Wright 1999; Kerrigan and Mayne 2002) (in the sense that if there exists $u^* \in \mathcal{U}$ such that $Bu^* = \tau$, then $s^* = 0$) by utilizing mixed norms in the objective function, e.g. $J(u, s, \tau) = \|Qs\|_1 + \|Ru\|_2$. This, however often yields a convex (as opposed to strictly convex) problem, which is slightly more complicated to solve explicitly (Spjøtvold, Tøndel, and Johansen 2007; Tøndel, Johansen, and Bemporad 2003c; Jones and Morari 2006).*

6.4 Explicit Solutions to Linear Constrained Control Allocation via Parametric Programming.

6.4.1 Problem Setup

Under certain assumptions on the control allocation problem (6.4), recent progress in parametric programming allows it to be solved explicitly, typically yielding a piecewise affine solution function (Johansen, Fossen, and Tøndel 2005). In the parametric programming setup we have that $\mathbb{P}(\tau)$ is to be solved for all values of $\tau \in \mathcal{T}$, where

$$\mathcal{T} := \{\tau \mid \exists (u, s) \quad \text{s.t.} \quad (u, s, \tau) \in \mathcal{Z},\ J^*(\tau) > -\infty\} = \operatorname{dom}(J^*).$$

Define the set-valued maps $\mathcal{Y} : \mathcal{T} \rightrightarrows \mathbb{R}^{n_u} \times \mathbb{R}^{n_\tau}$

$$\mathcal{Y}(\tau) := \{(u, s) \mid (u, s, \tau) \in \mathcal{Z}\}$$

and $\mathcal{Y}^* : \mathcal{T} \rightrightarrows \mathbb{R}^{n_u} \times \mathbb{R}^{n_\tau}$

$$\mathcal{Y}^*(\tau) := \arg\min_{(u,s)} \{J(u, s, \tau) \mid (u, s, \tau) \in \mathcal{Z}\}.$$

In the sequel we let $y := [u^T\ s^T]^T$ and let $y^*(\cdot)$ denote a selection of $\mathcal{Y}^*(\cdot)$, that is, $y^*(\tau) \in \mathcal{Y}^*(\tau)$ for all $\tau \in \mathcal{T}$.

6.4.2 Solution via Parametric Programming

One might distinguish between two types of linear allocation problems; $i)$ where the set $\mathcal{U}$ is convex, and $ii)$ when $\mathcal{U}$ is non-convex. However, in this chapter, we consider only the case where $\mathcal{U}$ is convex, closed and polyhedral. This immediately implies that (6.4) is a convex problem and that it attains its minimum for all

[3] $l = 2$ denotes, with some abuse of mathematical rigor, the quadratic norm, that is, $\|Qx\|_2 := x^T Qx$.

$\tau \in \mathcal{T}$. By using linear or quadratic norms in the cost function it is evident that $\mathbb{P}(\cdot)$ is either a parametric linear program (pLP) or parametric quadratic program (pQP). These problems have been subject to a vast amount of research in recent years (Gal and Nedoma 1972; Jones, Kerrigan, and Maciejowski 2007; Borrelli, Bemporad, and Morari 2003; Bemporad, Morari, Dua, and Pistikopoulos 2002; Spjøtvold, Tøndel, and Johansen 2007; Spjøtvold, Tøndel, and Johansen 2005a; Bank, Guddat, Klatte, Kummer, and Tammer 1983; Tøndel, Johansen, and Bemporad 2003a; Tøndel, Johansen, and Bemporad 2003c; Jones and Morari 2006). In parametric programming the goal is to divide the set of parameters of interest into a set of smaller regions such that each region is associated with a function that is optimal for the optimization problem when restricted to its region. For convenience we summarize the solution properties specialized to our problem formulation (Bemporad, Morari, Dua, and Pistikopoulos 2002; Bank, Guddat, Klatte, Kummer, and Tammer 1983; Gal and Nedoma 1972):

Theorem 6.1 (pLPs and pQPs) *Consider problem* (6.4) *and let* $\mathcal{U}$ *be a closed polyhedron.*

(i) If $l \in \{1, 2, \infty\}$*, then* $J^*(\cdot)$ *is continuous and convex on* $\mathcal{T}$.

(ii) If $l \in \{1, 2, \infty\}$*, then there exists a continuous selection* $y^*(\cdot)$ *of* $\mathcal{Y}^*(\cdot)$ *that is piecewise affine on* $\mathcal{T}$*. Moreover, if* $R > 0, Q > 0$ *and* $l = 2$*, then* $y^*(\cdot)$ *is unique.*

(iii) If $l \in \{1, \infty\}$*, then* $J^*(\cdot)$ *is piecewise affine on* $\mathcal{T}$.

(iv) If $l = 2$*, then* $J^*(\cdot)$ *is piecewise quadratic on* $\mathcal{T}$.

Both for pLPs and pQPs the optimizer is PWA and consequently the most common approach for solving the allocation problem where online-optimization is utilized can be substituted with an evaluation of a PWA function (Johansen, Fossen, and Tøndel 2005). There exists several methods for efficient evaluation of a PWA function (Jones, Grieder, and Raković 2006; Tøndel, Johansen, and Bemporad 2003b; Christophersen, Kvasnica, Jones, and Morari 2007), see Appendix C for a brief description of the method in (Tøndel, Johansen, and Bemporad 2003b) that is used in the present chapter.

The importance of the existence of a continuous selection should be emphasized when parametric programming is utilized to obtain explicit solutions to control allocations problems. The most important reason for desiring a continuous mapping from generalized forces to control inputs is to avoid unnecessary tear and wear of the actuators and variations in power consumption. There is also an issue of actually obtaining the desired generalized force; although the mapping is viewed as static this is often an approximation since the actuators are usually affected by rate constraints. See (Spjøtvold, Tøndel, and Johansen 2007; Tøndel, Johansen, and Bemporad 2003c; Jones and Morari 2006) for algorithms that obtain continuous selections to the relevant parametric programs.

6.4.3 Reconfigurable control allocation.

In many applications it is desirable to be able to switch on and off effectors or to change the constraints imposed on the control inputs to an effector. Reasons for this might be handling of actuator/effector failure and different operational modes. The most straightforward way of achieving this is to define additional parameters ϕ, and rewrite (6.4) as

$$J^*(\tau,\phi) := \min_{(u,s)} \{\|Qs\|_l + \|Ru\|_l \mid (u,s,\tau,\phi) \in \mathcal{Z}_\phi\},$$
$$\mathcal{Z}_\phi := \{(u,s,\tau,\phi) \mid Bu + s = \tau, u \in \mathcal{U}(\phi)\}.$$

This approach does not complicate the online optimization problem. In addition, if the parametrization $\mathcal{U}(\cdot)$ is linear, it is possible to solve the problem explicitly (Johansen, Fossen, and Tøndel 2005). However, with parametric programming the complexity of the optimal control $u^*(\cdot,\cdot)$ is often too high for the available memory as solution complexity scales quickly in the number of parameters. It may be also computationally expensive to obtain the explicit solution.

6.5 Case study: Control Allocation for a Thruster Controlled Floating Platform

In this section we present the problem formulation and experimental results for static control allocation for a scale model of a thruster-controlled floating platform, see Figures 6.1 and 6.2. Such platforms are used for offshore oil production, drilling, storage and offloading. The high level controller sending commands to the thrust allocation may be dynamic positioning, joystick control or thruster assisted position mooring control.

Hardware- and software-configuration and relevant physical properties for the experimental setup are given in Appendix D.

We first illustrate how to obtain an explicit solution to the control allocation problem when the high level controller specifies surge, sway and yaw forces and the optimization problem is convex. Secondly, we show how a selected set of thruster or power failure situations can be handled. National and international regulations (IMO 1994) require that the control system is operable after any single point failure, such as loss of a single thruster, single diesel generator or electric switchboard.

6.5.1 System Description

The model takes into account surge-, sway-, and yaw-motions, with the corresponding vessel fixed generalized forces $\tau := [X, Y, N]^T$. Assume that the vessel has a set $\mathcal{P} := \{p_1, p_2, \ldots, p_I\}$ of rotatable thrusters such that each device has two controls; direction and thrust magnitude. The thruster indexed by i is located at $r_i := [l_{i,x} \ l_{i,y} \ l_{i,z}]^T$ relative to the center of rotation in the vessel fixed coordinate system. Assume further that the force T_i from the i^{th} thruster is limited to the x-y plane in the vessel fixed coordinate system. Thruster p_i then produces a force T_i in the direction defined by the angle α_i. The contribution of the i^{th} thruster to the generalized forces acting on the vessel is given by:

$$X_i := T_i \cos \alpha_i, \tag{6.5a}$$
$$Y_i := T_i \sin \alpha_i, \tag{6.5b}$$
$$N_i := T_i \left(l_{i,x} \sin \alpha_i - l_{i,y} \cos \alpha_i \right), \tag{6.5c}$$

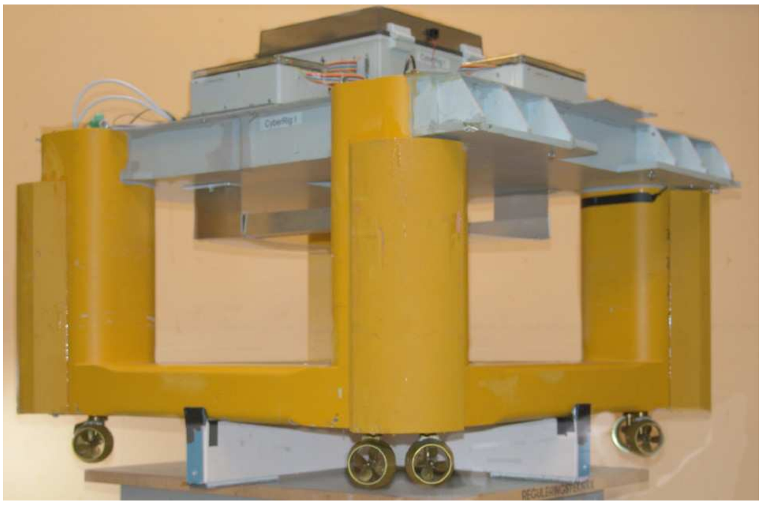


Figure 6.1: CyberRig I: Scale model of a thruster controlled platform.

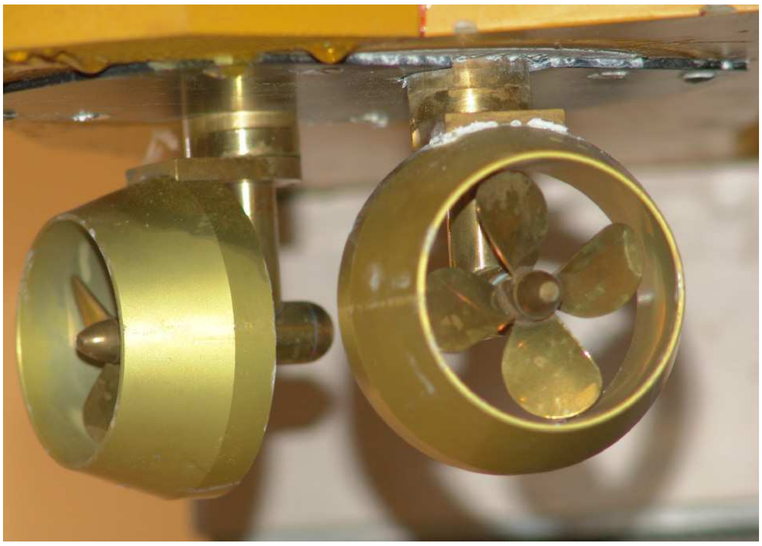


Figure 6.2: Thrusters one of the legs.

such that

$$X = \sum_{i=1}^{I} X_i, \quad Y = \sum_{i=1}^{I} Y_i, \quad \text{and} \quad N = \sum_{i=1}^{I} N_i.$$

In addition we have that each azimuth angle α_i and thrust force T_i are constrained to the sets

$$\begin{aligned} \mathcal{O}_i &:= \left\{(\alpha_i, T_i) \,\middle|\, \underline{\alpha}_i \le \alpha_i \le \overline{\alpha}_i, \underline{T}_i \le T_i \le \overline{T}_i \right\}, \\ &\quad i = \{1, 2, \ldots, I\}, \end{aligned}$$

where $\underline{\alpha}_i$, $\overline{\alpha}_i$, $\underline{T}_i$ and $\overline{T}_i$ are lower and upper bounds on the azimuth angle and thrust force for the i^{th} thruster, respectively. We introduce the concept of *attainable force set*[4]:

Definition 6.1 (Attainable Force Set (AFS)) *The* attainable force set *for a set of control effectors* $\mathcal{P} := \{p_1, p_2, \ldots, p_I\}$ *is given by*

$$\mathcal{T} := \{\tau \in \mathbb{R}^{n_\tau} \mid (\alpha_i, T_i) \in \mathcal{O}_i, \; i \in \{1, 2, \ldots, I\}\}.$$

In other words, the AFS is the set of generalized forces that can be generated by the thrusters while fulfilling the constraints. For offshore vessels the AFS is usually presented as a capability plot illustrating the wind and sea loads the thruster control system is able to counteract (International Marine Contractors Association 2000).

The relationship (6.5) can be written as the non-linear equation

$$\tau = A(\alpha)T,$$

where $\alpha := [\alpha_1, \ldots, \alpha_I]^T$ and $T := [T_1, \ldots, T_I]^T$. To obtain a linear relationship we follow the procedure in (Sørdalen 1997) where the concept of extended thrust is introduced. The extended thrust vector is found by decomposing the individual thrust vectors in the horizontal plane according to: $u_{i,x} := X_i, u_{i,y} := Y_i$ and $u_i := [u_{i,x} \; u_{i,y}]^T \in \mathbb{R}^2$. The generalized thrust vector is then given by the linear equation

$$\tau = Bu,$$

where $u := [u_{1,x} \; u_{1,y} \; u_{2,x} \; u_{2,y} \; \ldots \; u_{I,x} \; u_{I,y}]^T$ and the matrix B is given by

$$B := \begin{bmatrix} 1 & 0 & \ldots & 1 & 0 \\ 0 & 1 & \ldots & 0 & 1 \\ -l_{1,y} & l_{1,x} & \ldots & -l_{I,y} & l_{I,x} \end{bmatrix}.$$

[4] In the aviation literature the attainable force set is referred to as the attainable moment set (Durham 1993)

It is easy to see that the sets $\{\mathcal{O}_i\}_{i=1}^I$ translate into constraints on the controls $\{u_i\}_{i=1}^I$ defined by

$$u_i \in \mathcal{C}_i := \left\{ [u_{i,x}\ u_{i,y}]^T \;\middle|\; \begin{array}{l} u_{i,x} = T_i \cos\alpha_i, \\ u_{i,y} = T_i \sin\alpha_i, \\ (\alpha_i, T_i) \in \mathcal{O}_i \end{array} \right\},$$
$$i = \{1, 2, \ldots, I\}.$$

We will refer to the sets $\{\mathcal{C}_i\}_{i=1}^I$ as attainable thrust regions; the set of surge and sway forces that can be generated by a single thruster.

Definition 6.2 (Attainable Thrust Region (ATR)) *The* attainable thrust region *for a set of I thrusters is given by given by $\mathcal{C} := \bigoplus_{i=1}^I \mathcal{C}_i$.*

Hence, the ATR is the set of surge and sway forces that can be generated by a set of thrusters.

Thruster Model

In this chapter we utilize a conventional quadratic thruster characteristic (Fossen 2002), that is, the thrust force from a given thruster is given by

$$T = K_T \rho D^4 |n| n =: \gamma(K_T, \rho, n), \tag{6.6}$$

where K_T is a strictly positive thrust coefficient where the effect of losses have been accounted for, ρ is the water density, D is the propeller diameter, and n is the propeller speed. Assuming constant water density and thrust coefficient we see that $\gamma(\cdot)$ is reduced to a function only of the propeller speed. Consequently, if for the i^{th} thruster the desired extended thrust vectors are X_i and Y_i, we recover the thrust force T_i and azimuth angle α_i from the relationships (6.5) and the propeller speed from (6.6). Note that it is straightforward to replace (6.6) by a more advanced thruster characteristic.

The flow chart for constrained control allocation for the floating platform can be represented as depicted in Figure 6.3. Our task is to compute some optimal $u^*(\tau)$ when τ is given.

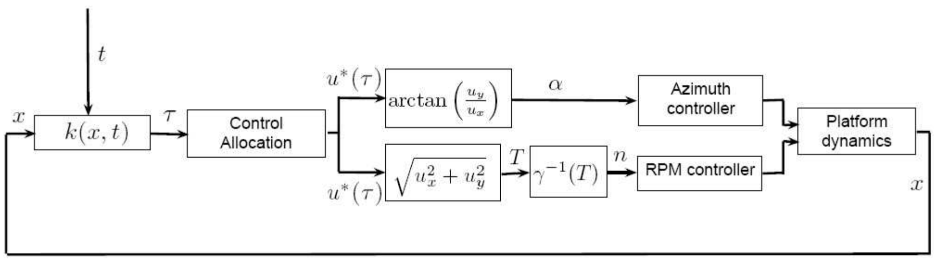
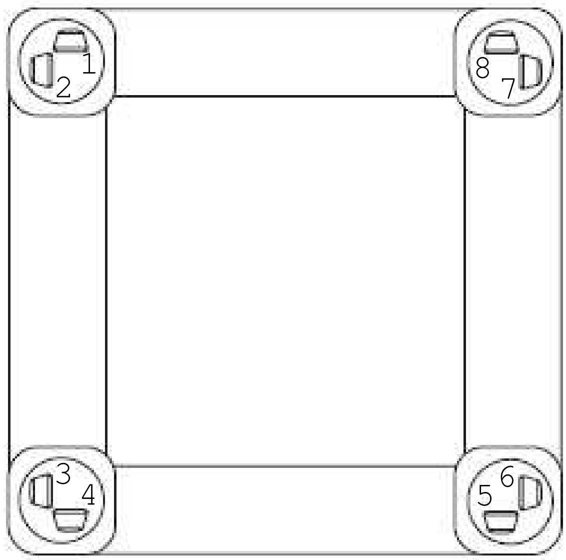


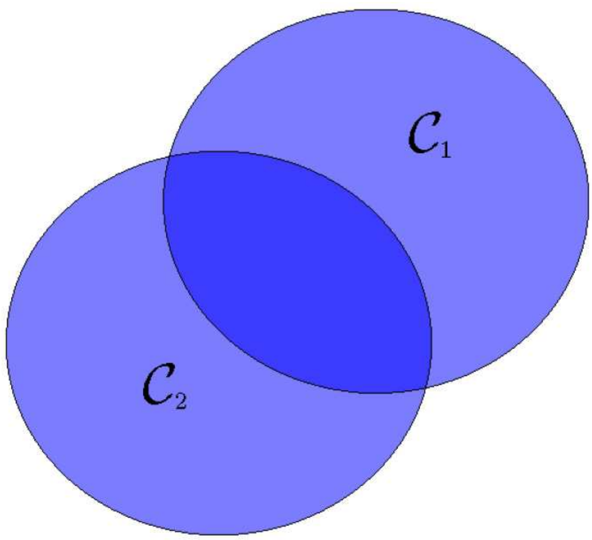
Figure 6.3: Flowchart for control and allocation for CyberRig I.

Thruster configuration

The thruster configuration for the floating platform is depicted in Figure 6.4(a). Two azimuth thrusters are placed on each of the four legs and each thruster can



(a) Thruster configuration for CyberRig I. Thruster 1 and 2 are situated on leg 1, thruster 3 and 4 on leg 2 etc.



(b) Translated ATRs for the two thrusters on leg 1.

Figure 6.4: Thruster configuration and translated ATRs.

rotate 360 degrees. It is not straightforward to obtain the AFS for the vessel due to the following: Considering one leg of the platform, the two thrusters are positioned such that one thruster may affect the flow pattern around the other thruster, resulting in loss of thrust and non-linear behavior such that (6.6) does not hold. In Figure 6.4(b) we have illustrated this interaction by translating the ATRs for the two thrusters on leg one to their physical location on the vessel.

To avoid this interaction, sectors ($\mathcal{S}_i$ and $\mathcal{S}_{i+1}$) are introduced that are mutually exclusive in the sense that if thruster p_i produces a force in direction $\alpha_i \in \mathcal{S}_i$, then p_{i+1} cannot produce a force in direction $\alpha_{i+1} \in \mathcal{S}_{i+1}$, where $i \in \{1, 3, 5, 7\}$. See Figures 6.4(b) and 6.5(a) for an illustration. More precisely,

$$\alpha_i \in \mathcal{S}_i \Rightarrow \alpha_{i+1} \notin \mathcal{S}_{i+1}, \quad i \in \{1, 3, 5, 7\}.$$

The most straightforward approach that may be utilized to meet this constraint is to introduce forbidden sectors such that thruster p_i (p_{i+1}) never produce a force in direction $\alpha_i \in \mathcal{S}_i$ ($\alpha_{i+1} \in \mathcal{S}_{i+1}$), where $i \in \{1, 3, 5, 7\}$. This means that p_i can only produce a force in a restricted ATR, that is, $u_i \in \bar{\mathcal{C}}_i := \mathcal{C}_i \backslash \mathcal{S}_i$. See Figure 6.5(a) for an illustration.

6.5.2 Solution approach

We convexify and approximate the problem by restricting the ATR for each thruster to be an inner polyhedral approximation of a half circle, see Figure 6.5(b). In the sequel, we let inner approximations of the restricted ATRs $\{\bar{\mathcal{C}}_i\}_{i=1}^I$ be denoted by $\{\mathcal{U}_i\}_{i=1}^I$. We then minimize the thrust magnitude for each thruster and the problem becomes:

$$J^*(\tau) := \min_{u,s} \left\{ \frac{1}{2} \left(u^T R u + s^T Q s \right) \;\middle|\; (u, s, \tau) \in \mathcal{Z} \right\}, \tag{6.7a}$$

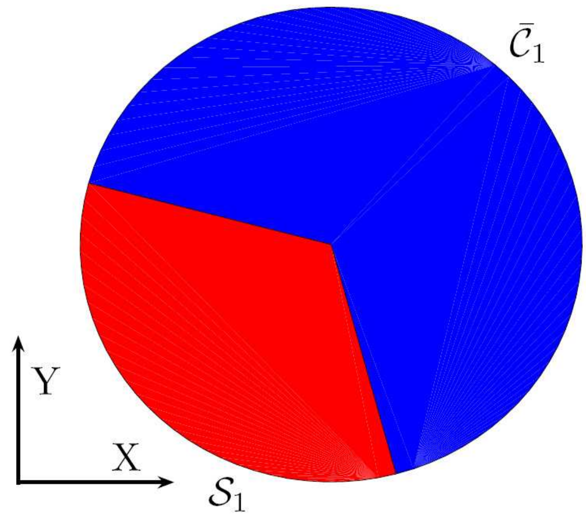
$$\mathcal{Z} := \left\{ (u, s, \tau) \;\middle|\; \begin{array}{l} Bu = \tau + s, \\ u_i \in \mathcal{U}_i, \; i \in \{1, 2, \ldots, 8\} \end{array} \right\}, \tag{6.7b}$$

where $R = I$, $Q = 10^3 \times I$, and $(u, s, \tau) \in \mathbb{R}^{16} \times \mathbb{R}^3 \times \mathbb{R}^3$. This is clearly a convex optimization problem that can be solved by a single pQP.

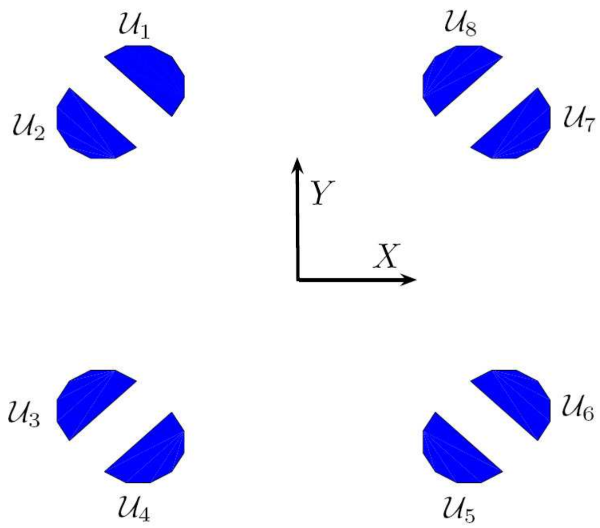
6.5.3 Fault tolerant control allocation

As described in Section 6.4.3, fault tolerant control allocation may be computationally demanding if an explicit solution to the problem is desired. However, for this particular application, the geometry of the problem can be exploited to obtain great reduction in both storage space and required off-line computation.

Consider the case where thruster 1 fails, abbreviated Pf1. The solution to this problem is obtained simply by removing the associated controls u_1 and corresponding constraints from the optimization problem. The question becomes whether solving this problem also give us the solutions to the scenarios where thruster 3, 5, or 7 fail. We argue that this is the case for thruster 3, abbreviated Pf3 (the arguments, with obvious modifications, hold for the other situations as well). Pf3



(a) Restricted ATR $\bar{\mathcal{C}}_1$ for top thruster on the top left leg after forbidden sector has been artificially removed. The sector $\mathcal{S}_1$ is removed to avoid unwanted non-linear interaction with the second thruster on the leg.



(b) Translated and restricted ATRs for all thrusters after forbidden sectors have been removed.

Figure 6.5: Approximation of the ATRs.

would be identical to Pf1 if the vessel fixed coordinate was rotated 90 degrees, however, surge and sway forces for each thruster would be defined relative to the rotated coordinate system. Consequently, we can obtain the solution to Pf3 by the following procedure:

Thruster failure	Rotated 90	Rotated 180	Rotated 270
1	3	5	7
2	4	6	8
1 & 2	3 & 4	5 & 6	7 & 8

Table 6.1: Leftmost column show the failure situations that have been explicitly solved. The three other columns show what failure situations that are equivalent if the vessel fixed coordinated system is rotated.

(i) Rotate the surge and sway components of τ by 90 degrees, i.e

$$\tau^r = \begin{bmatrix} R(90) \begin{bmatrix} X \\ Y \end{bmatrix} \\ N \end{bmatrix}.$$

(ii) Evaluate the solution to Pf1 at τ^r, and denoted the solution $u^r(\tau^r)$.

(iii) Rotate each $u_i^r(\tau^r)$, $i = 2, \ldots, 8$, by -90 degrees to obtain $\bar{u}_i^*(\tau^r)$, $i = 2, \ldots, 8$. Let $\bar{u}_i^*$ be the control input to thruster $i + 2$, $i = 2, \ldots, 6$ and $\bar{u}_7^*$ and $\bar{u}_8^*$ the control inputs to thrusters 1 and 2, respectively.

Note that this procedure only works because the yaw component (N) of the generalized force is not affected by the rotation of the coordinate system due to the geometry of the problem. See Table D.1 for the lengths $l_{(\cdot,\cdot)}$.

In Table 6.1 we have listed cases for which the solution can be found simply by rotation of another solution. Hence, by considering 3 different failure configurations, we obtain solutions for 12 cases.

6.5.4 Experimental Results

Nominal operation

The explicit solution to (6.7) consists of 12522 polyhedral regions. A binary search tree (Tøndel, Johansen, and Bemporad 2003b) (see Appendix C) was then constructed for the purpose of evaluating the PWA function. The worst case depth of the search tree was 24, worst case number of arithmetic operations needed to find the solution was 264, and the tree was stored using 4.218.546 numbers, 468.923 being integers and 3.749.623 being real numbers. On a Dell LATTITUDE laptop with a 1.7 GHz Intel Pentium M CPU running Windows XP and MATLAB 7.0 the PWA function could be evaluated at approximately 200 kHz. As described in Appendix D, the QNX real-time system has a sample frequency of 10 Hz, so the processor was freed for other tasks. In Figure 6.6 we have depicted commanded and measured[5] generalized forces for the case where all desired generalized forces

[5]Only the RPMs and azimuth angles are measured and the *measured generalized forces* are derived from the inverse relationship in Figure 6.3.

was contained in the feasible part of the AFS. In Figure 6.8(a) we have illustrated the restricted ATRs and the measured thrust forces. In Figures 6.10(a), 6.11(a) and 6.12(a) we have depicted the measured azimuth angles, RPMs and extended thrust vectors, respectively. In Figure 6.7 we let the desired generalized forces be infeasible (that is, either not contained in the AFS or in an infeasible part) in large parts of the time-series. We cannot expect to achieve the desired generalized force in this case, and as Figure 6.8(b) shows, the constraints are fulfilled, but that the desired generalized force is not obtained. It is simple to prioritize which component (surge, sway or yaw) that is the most important by setting the weights on the slack variables s. In Figures 6.10(b), 6.11(b) and 6.12(b) we have depicted the measured azimuth angles, RPMs and extended thrust vectors, respectively.

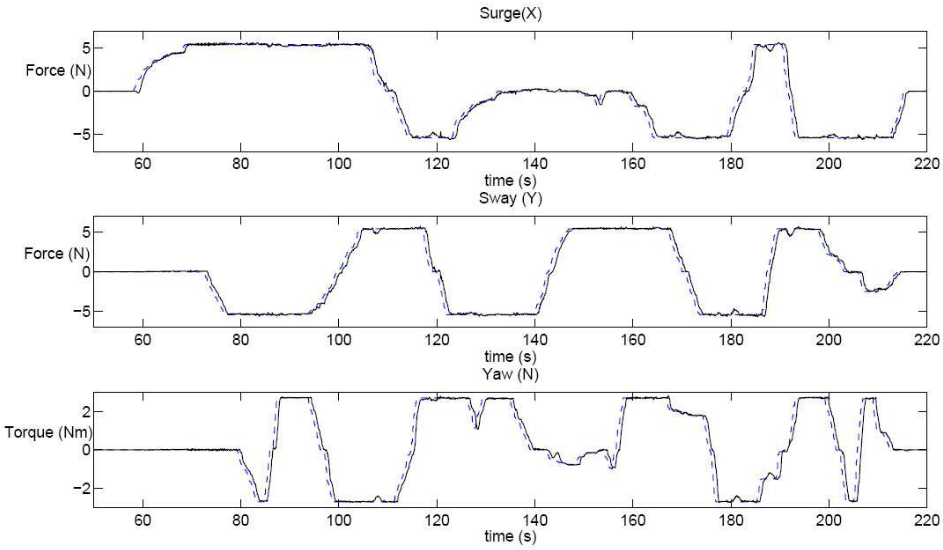


Figure 6.6: Experimental results for CyberRig I; control allocation over convex ATRs and the desired generalized force was always contained in the AFS. The dotted line is the commanded generalized force and the solid line is measured.

Fault tolerant control allocation

In Figure 6.9 we have depicted results for when thruster failure occurs. Assuming the electric power buses are split into four segments, each corresponding to a machine room and switchboard feeding two thrusters in each leg, this covers all critical single point failures; from a single thruster to a whole machine room.

6.6 Further remarks and research

There are several interesting research directions for improving the control allocation for the platform:

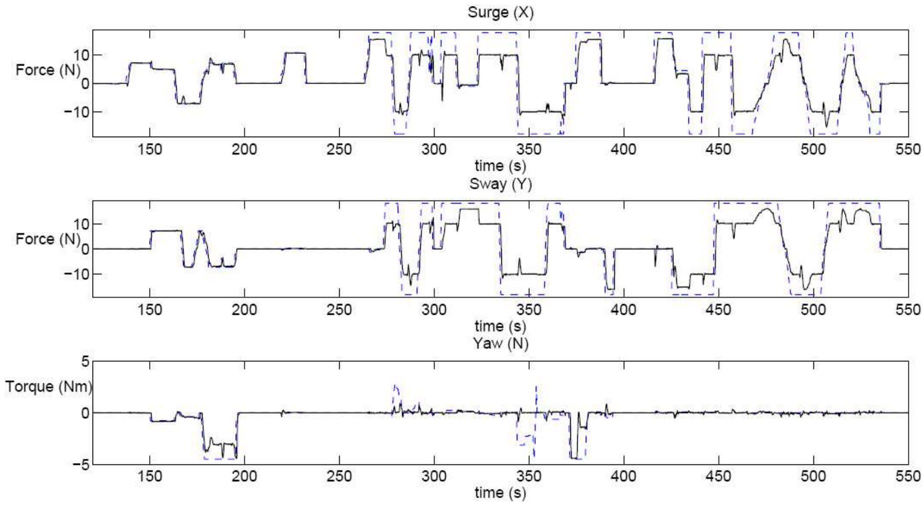


Figure 6.7: Experimental results for CyberRig I; control allocation over convex ATRs and the desired generalized force was not always contained in the AFS. The dotted line is the commanded generalized force and the solid line is measured.

1. It would be an improvement to be able to utilize a larger part or the entire AFS. This would lead to a non-convex and non-linear optimization problem.

2. Rate constraints should be included in the optimization problem. This will greatly complicate the parametric optimization problem as the previous control inputs would enter the problem as parameters yielding a 19 dimensional parameter vector.

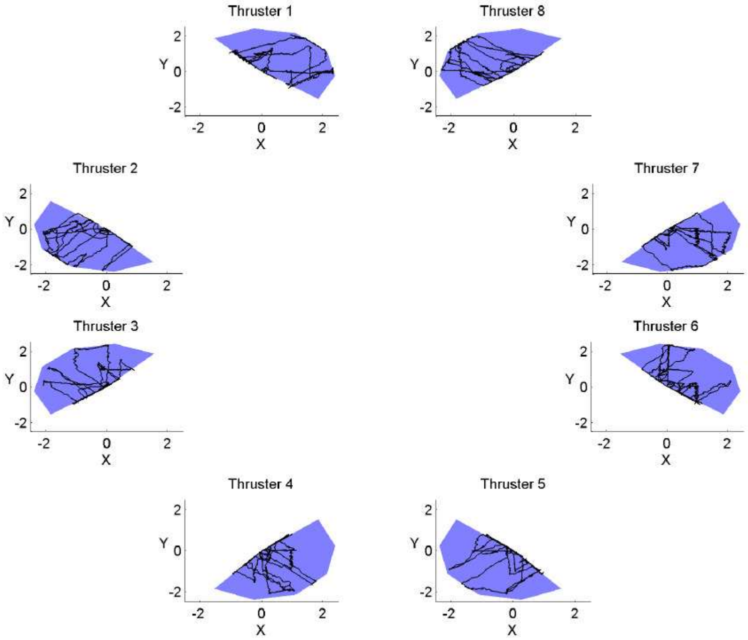
3. A more meaningful cost function should be formulated. As briefly mentioned earlier, rotating a thruster is also power consuming and leads to tear and wear. It would be natural to consider penalizing both change in thrust magnitude and in the azimuth angle in addition to thrust magnitude.

4. As there are no restrictions on the rate of change in the desired generalized force, the problem may obviously be infeasible in some cases (e.g. a thruster can not rotate 180 degrees from one sample to the next). It may then be beneficial to minimize deviation from the desired generalized force over a prediction horizon, taking into account the thruster dynamics, current azimuth angles and RPMs. It is easy to imagine that the optimal azimuth angles and RPMs for the thrusters are far from the current situation, but that the desired generalized force can be generated with a configuration that is close.

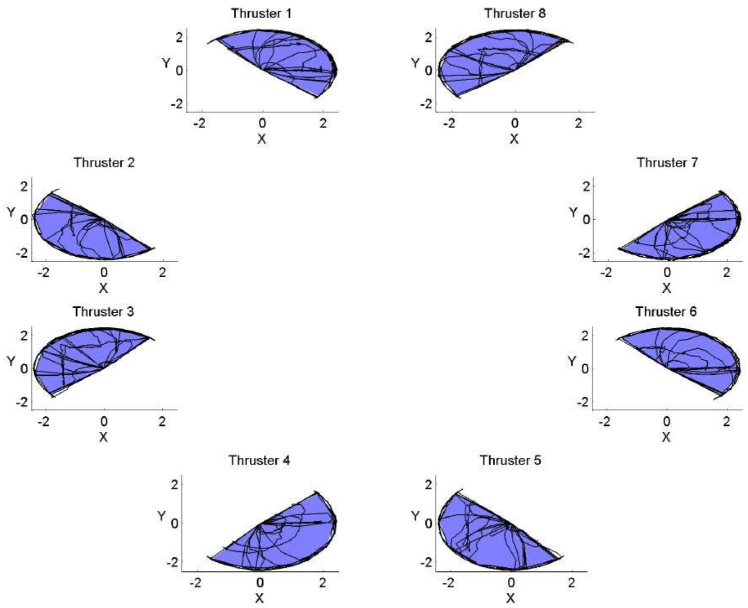
6.7 Conclusion

A convex constrained control allocation problem was formulated for the purpose of mapping desired generalized forces to control inputs. An explicit solution was obtained by utilizing parametric quadratic programming. The PWA solution function was evaluated using a binary search tree and the allocation scheme was implemented on a scale model of a thruster-controlled floating platform.

The PWA function could be evaluated at a frequency of several kHz, well within the sampling rate of 10 Hz, and hence, freeing the computational unit for other tasks. The tracking of the generalized forces was shown to be satisfactory even if rate constraints were not included in the formulation. The method also performed well under single point failure situations.



(a) Measured force from each individual thruster where we have shaded the translated ATRs. All generalized forces are feasible.



(b) Measured force from each individual thruster where we have shaded the translated ATRs. Clearly, the constraints are fulfilled, but the controls are at the boundary of the ATRs for large parts of the time series since the corresponding desired generalized forces were infeasible.

Figure 6.8: Measured thrust forces and the translated ATRs.

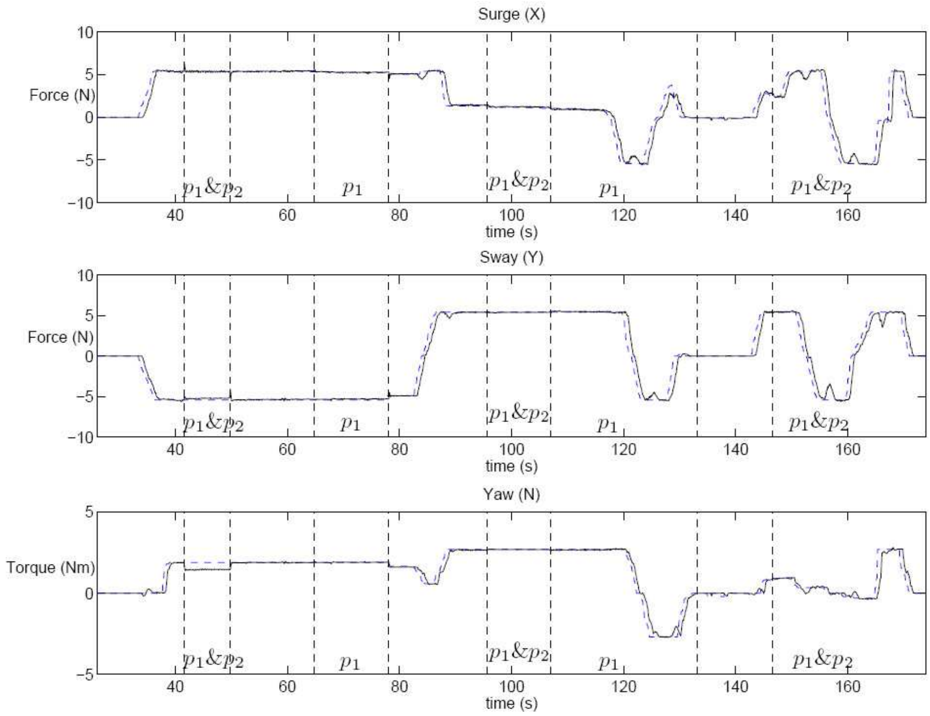
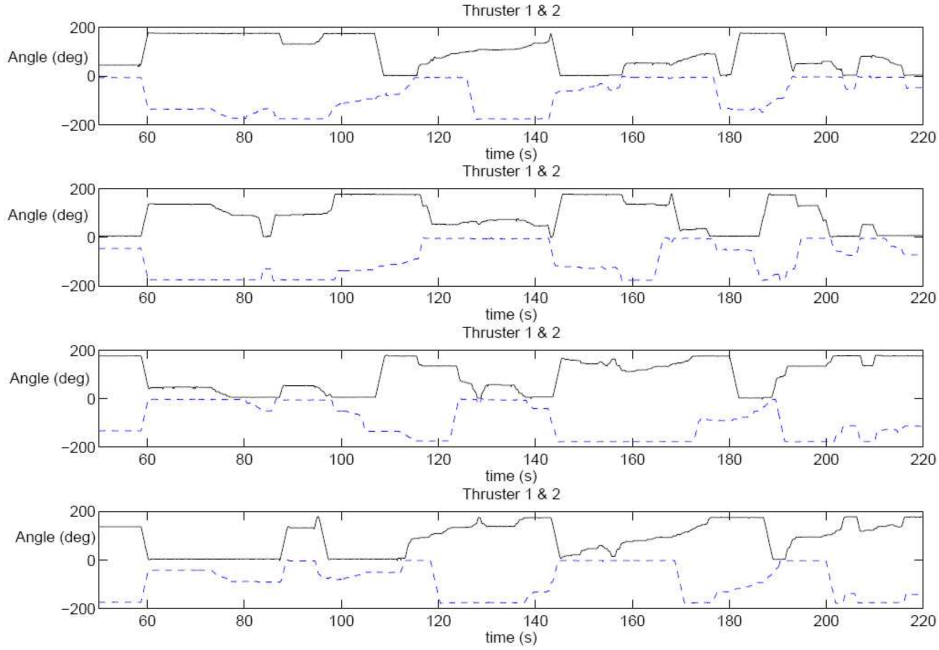
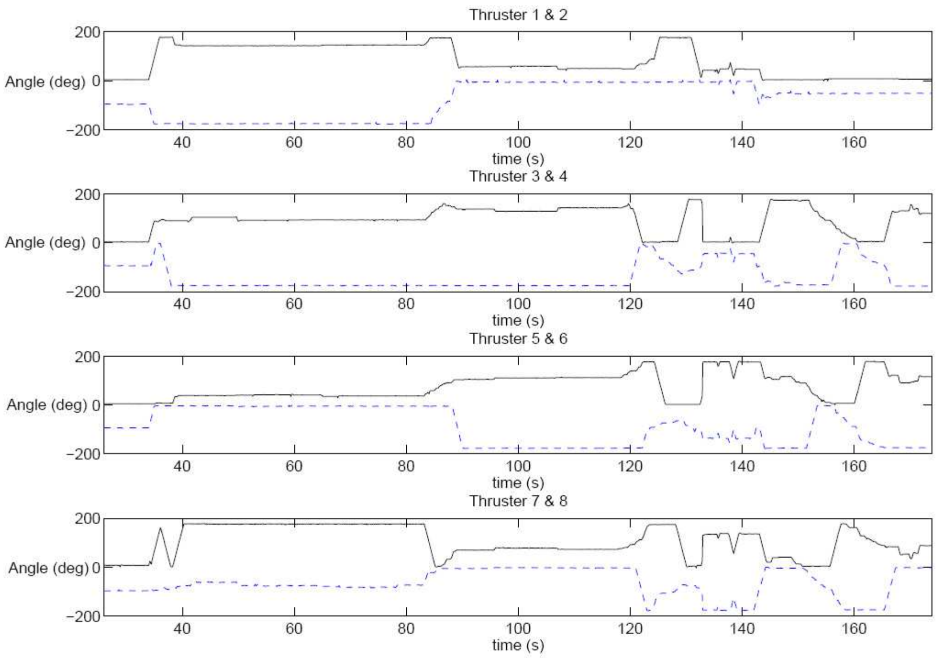


Figure 6.9: Experimental results for CyberRig I; control allocation over convex ATRs. We have indicated time intervals in which either thruster 1 (p_1) or where both thruster 1 and 2 $(p_1\&p_2)$ have failed. The dotted line is the commanded and the solid line is measured.

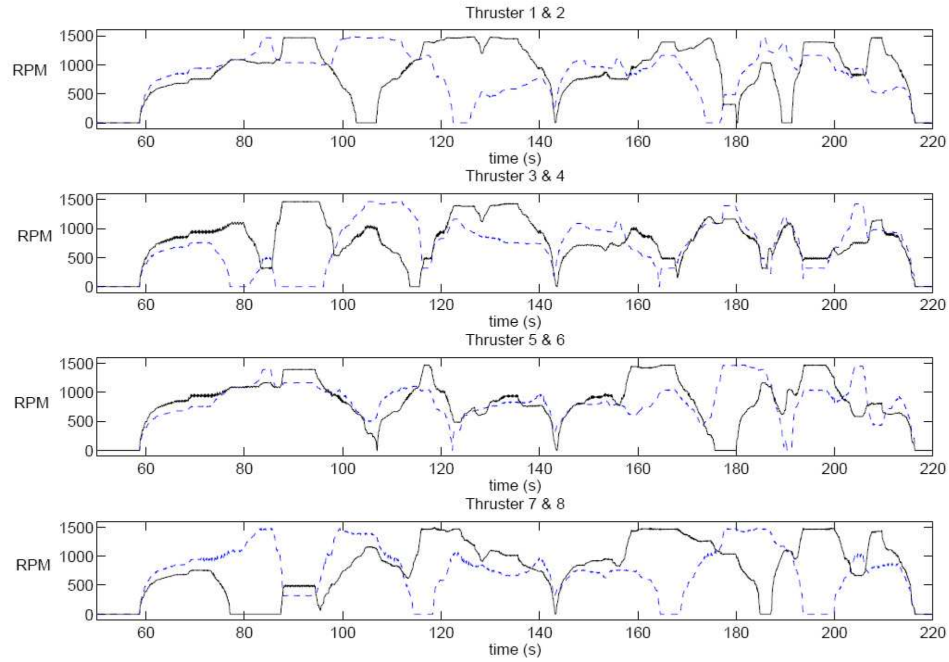


(a) Azimuth angles under nominal operation.

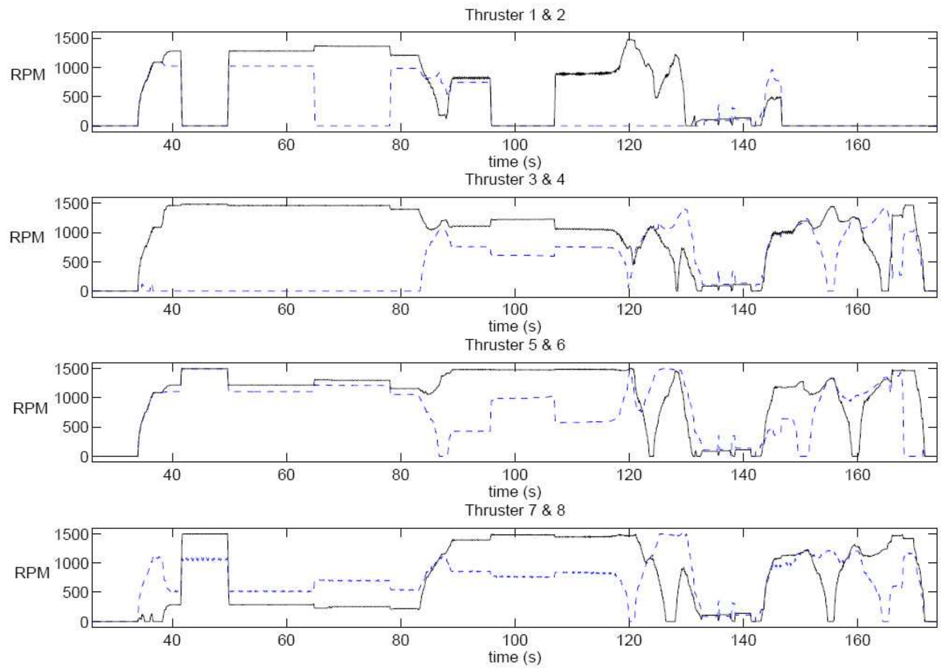


(b) Azimuth angles in failure situations.

Figure 6.10: Dotted lines are azimuth angles for thrusters 1, 3, 5 and 7 and solid lines are for 2, 4, 6 and 8.

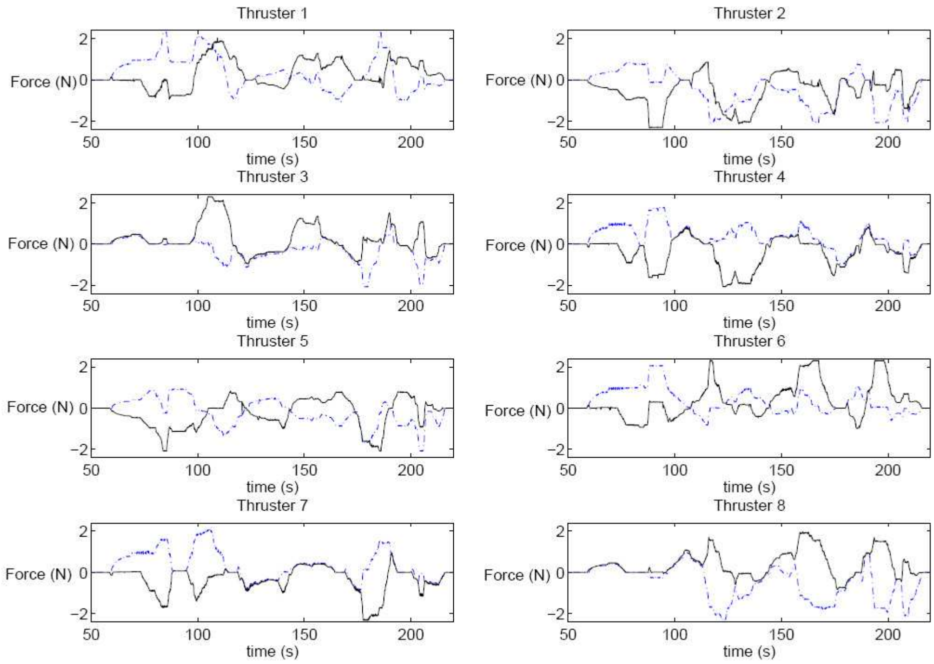


(a) RPMs under nominal operation.

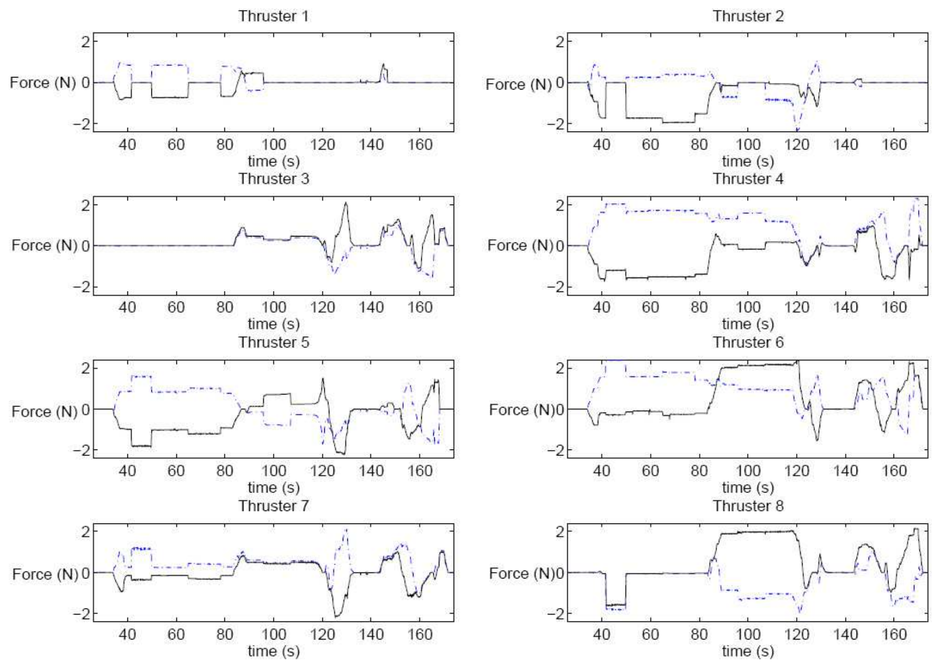


(b) RPMs in failure situations.

Figure 6.11: Dotted lines are RPMs for thrusters 1, 3, 5 and 7 and solid lines are for 2, 4, 6 and 8.



(a) Extended thrust vectors under nominal operation.



(b) Extended thrust vectors in failure situations.

Figure 6.12: Extended thrust vectors. Dotted line is $u_{i,x}$ and solid line is $u_{i,y}$.

Chapter 7

Decomposing Constrained Control Allocation Problems

7.1 Introduction

The task in control allocation is to determine how to generate a specified generalized force from a redundant set of actuators where the associated controls are constrained, see e.g. (Bodson 2002; Bodson and Pohlchuck 1998; Buffington, Enns, and Teel 1998; Johansen, Fossen, and Tøndel 2005; Luo and Doman 2004; Johansen, Fuglseth, Tøndel, and Fossen 2003). The main objective is to obtain the desired generalized force, however, it is also common to incorporate secondary objectives, such as minimizing energy consumption and limiting the rate of change for a control input. Several other factors, such as actuator dynamics (Luo and Doman 2004) and power management, can also be incorporated. One way of achieving these secondary goals is to solve a constrained optimization problem online at every sampling instant.

Only recently, it has, in conformity with the explicit model predictive control approach (Bemporad, Morari, Dua, and Pistikopoulos 2002; Bemporad, Borrelli, and Morari 2002), been proposed to solve the optimization problem offline (Johansen, Fossen, and Tøndel 2005) by utilizing parametric programming techniques (Gal and Nedoma 1972; Bemporad, Morari, Dua, and Pistikopoulos 2002; Dua and Pistikopoulos 2000; Tøndel, Johansen, and Bemporad 2003a; Bank, Guddat, Klatte, Kummer, and Tammer 1983). The online computational effort then reduces to evaluate a piecewise affine function, which can be formulated as a point location problem (Tøndel, Johansen, and Bemporad 2003b; Jones, Grieder, and Raković 2006). The four main advantages of this approach are: i) removing the need for sophisticated optimization software on the microchip/proseccor, ii) the correctness of the solution can be verified off-line, which is a key issue in safety critical applications, iii) the worst case number of arithmetic operations needed to find the solution can easily be computed, and iv) for a large class of problems the average and worst case number of arithmetic operations needed to find the solution is greatly reduced. The main drawbacks, on the other hand, are that $i)$ obtaining an explicit solution may be computationally intractable, $ii)$ the storage space required to represent the solution may exceed the available memory, and $iii)$ in the context of constrained control allocation, the method does not easily allow reconfigurable control without increasing solution complexity.

In this chapter we propose a decomposing strategy for obtaining feasible, sub-optimal solutions to constrained linear control allocation problems. The procedure is motivated by the observation that for practical problems not all the actuators interact directly, suggesting a division of the problem into a set of smaller problems. The actuators are partitioned such that each element of the partition does not interact with the other elements (note that in this chapter, the term *actuators* also include effectors, for example, both the rudder, and the engine that drives it, are labelled actuators). A master- and a set of sub-problems are designed for the purpose of obtaining a feasible, but sub-optimal solution. The decomposing scheme is also extended to yield an optimal solution for a class of allocation problems. In the proposed scheme we can choose to solve some of the problems explicitly and some online, allowing the designer to choose an approach that is best suited for the hardware and software available. Another benefit of the procedure is that reconfigurable control is somewhat more computationally tractable.

7.2 Problem setup

7.2.1 Basic definitions and nomenclature

If $\mathcal{I}$ is an index set, then $|\mathcal{I}|$ denotes the cardinality of $\mathcal{I}$ and $\mathcal{I}_i$ refers to the i^{th} element in $\mathcal{I}$. When referring to a set of indices $\mathcal{I}$, we assume that the set is ordered, i.e. for the i^{th} element in $\mathcal{I}$ we have $\mathcal{I}_i < \mathcal{I}_j, \forall j \in \{i+1, \ldots, |\mathcal{I}|\}$. Recall that a partition of a set S is a collection of sub-sets of S such that the sub-sets are mutually disjoint and their union is equal to S. Let $\mathbb{N}_q$ denote the set $\{1, 2, \ldots, q\}$. If $A \in \mathbb{R}^{n\times m}$ is a matrix or column vector, then $A_{(i,*)} \in \mathbb{R}^{1\times m}$ denotes the i^{th} row of A and $A_{(\mathcal{I},*)} \in \mathbb{R}^{|\mathcal{I}|\times m}$ denotes the matrix $[A^T_{(\mathcal{I}_1,*)}, \ldots, A^T_{(\mathcal{I}_{|\mathcal{I}|},*)}]^T$. Similarly, $A_{(*,i)} \in \mathbb{R}^{n\times 1}$ denotes the i^{th} column of A and $A_{(*,\mathcal{I})} \in \mathbb{R}^{n\times|\mathcal{I}|}$ denotes the matrix $[A_{(*,\mathcal{I}_1)}, \ldots, A_{(*,\mathcal{I}_{|\mathcal{I}|})}]$. If A is a column vector, i.e. $A \in \mathbb{R}^{n\times 1}$, then $A_{(\mathcal{I},*)} \in \mathbb{R}^{|\mathcal{I}|\times 1}$ is abbreviated $A_{\mathcal{I}}$. Finally, if $\{\mathcal{J}^i \mid i \in \mathcal{I}\}$ is a partition of the index set $\mathcal{J}$ and $u \in \mathbb{R}^{|\mathcal{J}|}$ is a vector, we define the operator $\mathrm{sort}(\cdot)$ as the operator that maps the set of sub-vectors $\{u_{\mathcal{J}^i} \mid i \in \mathcal{I}\}$ into $\mathbb{R}^{|\mathcal{J}|}$ and restores the original ordering of the vector, i.e. $u = \mathrm{sort}\left(\{u_{\mathcal{J}^i} \mid i \in \mathcal{I}\}\right)$.

Recall that the set of affine combinations of points in a set $S \subset \mathbb{R}^n$ is called the *affine hull* of S, and is denoted $\mathrm{aff}(S)$. The *dimension of a set* $S \subset \mathbb{R}^n$ is the dimension of $\mathrm{aff}(S)$, and is denoted $\dim(S)$; if $\dim(S) = n$, then S is said to be full-dimensional. The *closure* and *interior* of a set S is denoted $\mathrm{cl}(S)$ and $\mathrm{int}(S)$, respectively. A *polyhedron* is the intersection of a finite number of open and/or closed half-spaces. A *polygon* is a finite union of polyhedra. If $F : X \to Y$ is a mapping, then the restriction of F to the domain $D \subseteq X$ is written $F|_D : D \to Y$. If a mapping F is set-valued the notation $F : X \rightrightarrows Y$ specifies this. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be *piecewise affine* (PWA) on $D \subset \mathbb{R}^n$ if D can be represented as a finite union of polyhedra, relative to each of which $f(x)$ is given by an affine expression.

7.2.2 Static linear control allocation

Consider the equation

$$\bar{\tau} = Bu,$$

where $\bar{\tau} \in \mathcal{T} \subseteq \mathbb{R}^r$ are the generalized forces (virtual controls), $u \in \mathbb{R}^m$ are the controls, and the matrix $B \in \mathbb{R}^{r\times m}$ defines the (linear) relationship between the generalized forces and the controls. Assume further that a virtual controller $\tau := k(t, x)$ is given, i.e. τ is our desired generalized force (virtual control). The task in control allocation is to generate the force τ the controller specifies using the available controls $u \in U \subseteq \mathbb{R}^m$, where U is assumed to be full-dimensional and bounded. Since, in general, one cannot assume that it is possible to generate τ when u is constrained to U, slacks s are introduced in order to ensure that a solution is always obtained, i.e. $Bu + s = \tau$. Hence, the linear control allocation problem can be stated as:

$$\mathbb{P}(\tau): \; J^*(\tau) := \inf_{u,s} \left\{ J(u, s, \tau) \mid (u, s, \tau) \in \mathcal{Y} \right\}, \tag{7.1a}$$

$$J(u, s, \tau) := \|Qs\|_l + \|Ru\|_l, \tag{7.1b}$$

$$\mathcal{Y} := \left\{ (u, s, \tau) \;\middle|\; \begin{array}{l} Bu + s = \tau, \\ u \in U \end{array} \right\}, \tag{7.1c}$$

where $Q \in \mathbb{R}^{p\times p}$ and $R \in \mathbb{R}^{m\times m}$ are weight matrices, respectively penalizing use of controls and infeasibility, and $l \in \{1, 2, \infty\}$[1] denotes the weighting norm. We will assume that $\mathbb{P}(\tau)$ attains its minimum $\forall \tau \in \mathcal{T}$, where $\mathcal{T}$ is a full-dimensional polygon (where each polyhedron in $\mathcal{T}$ is also assumed to be full-dimensional). Henceforth, we write the problem as minimization. In the sequel let the set-valued map $\mathcal{Y}^* : \mathbb{R}^r \rightrightarrows \mathbb{R}^m \times \mathbb{R}^r$ be defined by

$$\mathcal{Y}^*(\tau) := \arg\min_{(u,s)} \{J(u, s, \tau) \mid (u, s, \tau) \in \mathcal{Y}\},$$

and let $(u^*, s^*) : \mathbb{R}^r \to \mathbb{R}^m \times \mathbb{R}^r$ denote a single-valued selection of $\mathcal{Y}^*(\cdot)$, i.e. $(u^*(\tau), s^*(\tau)) \in \mathcal{Y}^*(\tau)$ for all $\tau \in \mathcal{T}$.

In the sequel we distinguish between two types of linear allocation problems; $i)$ where the set U is convex, and $ii)$ when U is non-convex. We will also make use of the following assumption:

[1] $l = 2$ denotes, with some abuse of mathematical rigor, the quadratic norm, that is, $\|Qx\|_2 := x^T Qx$.

Assumption 7.1 *Define:*

$$\mathbb{P}_\varepsilon(\tau): \quad J^*_\varepsilon(\tau) := \min_{(u,s)} \{J(u,s,\tau) \mid (u,s,\tau) \in \mathcal{Y}_\varepsilon\},$$
$$\mathcal{Y}_\varepsilon := \left\{ (u,s,\tau) \;\middle|\; \begin{array}{l} Bu + s = \tau, \\ u \in U_\varepsilon \end{array} \right\},$$

and let the set U be full-dimensional and bounded. Given any $\varepsilon > 0$ we assume that there exists a polygon $U_\varepsilon := \cup_{i\in\mathcal{I}} U^i$ that inner approximates U in the sense that $U_\varepsilon \subseteq U$, where $\mathcal{I}$ contains a finite number of elements and each U^i is a full-dimensional polyhedron, and

$$\forall \tau \in \mathcal{T} \; J^*_\varepsilon(\tau) \leq J^*(\tau) + \varepsilon \textit{ and } \arg\min_{(u,s)} \{J(u,s,\tau) \mid (u,s,\tau) \in \mathcal{Y}_\varepsilon\} \neq \emptyset.$$

As a consequence of the above assumption, we will henceforth assume that the set U in (7.1) is a polygon or a polyhedron, which will be clear from the context.

7.2.3 Reconfigurable control allocation

In many applications it is desirable to be able to switch on and off actuators or to change the constraints imposed on the control inputs to an actuator. The most straightforward way of achieving this is to define additional parameters ϕ, and rewrite (7.1) as

$$J^*(\tau,\phi) : = \min_{(u,s)} \{\|Qs\|_l + \|Ru\|_l \mid (u,s,\tau,\phi) \in \mathcal{Y}\},$$
$$\mathcal{Y} : = \{(u,s,\tau,\phi) \mid Bu + s = \tau, u \in U(\phi)\}.$$

This approach does not complicate the online optimization problem. In addition, if the parametrization $U(\cdot)$ is linear, it is possible to solve the problem explicitly (Johansen, Fossen, and Tøndel 2005), however, with this approach the complexity of the optimal control $u^*(\cdot,\cdot)$ is often too high for the available memory, and, in some cases, it may even be computationally intractable to obtain the explicit solution.

7.3 Explicit solutions to control allocation problems

Recently it has been proposed to solve (7.1) explicitly(see e.g. (Johansen, Fossen, and Tøndel 2005; Johansen, Fuglseth, Tøndel, and Fossen 2003)) and thereby avoid online optimization. The next three subsections summarizes the solution properties of parametric linear-, quadratic-, and mixed integer linear programs.

7.3.1 Parametric linear programs

Consider the linear program with parameters on the right hand side of the constraints:

$$J^*(\theta) := \min_x \left\{ c^T x \mid (x,\theta) \in P \right\}, \tag{7.2a}$$

$$P := \left\{ (x,\theta) \mid Ax \leq b + S\theta \right\}, \tag{7.2b}$$

where c, A, b, and S are matrices with suitable dimensions, and (7.2) is to be solved for all values of $\theta \in \Theta \subseteq \mathbb{R}^s$, where Θ is the set of parameters in which the minimum in (7.2) exists.

Theorem 7.1 *Consider* (7.2)*.*

1. *The function $J^* : \Theta \to \mathbb{R}$ is continuous, convex and PWA on closed, full-dimensional, polyhedra.*
2. *There exists an optimizer function $x^* : \Theta \to \mathbb{R}^n$ such that*

$$x^*(\theta) \in \arg\min_x \left\{ c^T x \mid (x,\theta) \in P \right\},$$

that is continuous and PWA on closed, full-dimensional, polyhedra.

Obtaining a continuous selection $x^*(\cdot)$ can be done for instance via lexicographic perturbation of the pLP (Jones, Kerrigan, and Maciejowski 2007) or by choosing the minimum norm solution (Spjøtvold, Tøndel, and Johansen 2005a).

7.3.2 Parametric quadratic programs

Consider the convex quadratic program with parameters on the right hand side of the constraints:

$$J^*(\theta) := \min_x \left\{ \frac{1}{2} x^T H x + c^T x \,\middle|\, (x,\theta) \in P \right\}, \tag{7.3a}$$

$$P := \left\{ (x,\theta) \mid Ax \leq b + S\theta \right\}, \tag{7.3b}$$

where H, c, A, b, and S are matrices with suitable dimensions, and $H = H^T \geq 0$.

Theorem 7.2 *Consider* (7.3)

1. *The function $J^* : \Theta \to \mathbb{R}$ is continuous, convex and piecewise quadratic on closed, full-dimensional, polyhedra.*
2. *There exists an optimizer function $x^* : \Theta \to \mathbb{R}^n$ such that*

$$x^*(\theta) \in \arg\min_x \left\{ \frac{1}{2} x^T H x + c^T x \,\middle|\, (x,\theta) \in P \right\},$$

that is continuous and PWA on closed, full-dimensional, polyhedra.

A continuous selection can be obtained by choosing the minimum norm solution (Spjøtvold, Tøndel, and Johansen 2007). Note that if $H > 0$, the solution $x^*(\cdot)$ to (7.3) is unique, and hence, also continuous.

7.3.3 Parametric mixed-integer linear programs

Consider the mixed integer linear program with parameters on the right hand side of the constraints:

$$J^*(\theta) := \min_{(x,y)} \left\{ c^T x + d^T y \mid (x,y,\theta) \in P \right\}, \tag{7.4a}$$

$$P := \{(x,y,\theta) \in \mathbb{R}^n \times \{0,1\}^p \times \mathbb{R}^r \mid Ax + Dy \leq b + S\theta\} \tag{7.4b}$$

where c, d, A, D, b, and S are matrices with suitable dimensions.

Theorem 7.3 *Consider* (7.3)

1. *The function* $J^* : \Theta \to \mathbb{R}$ *is lower-semicontinuous and PWA.*
2. *There exists optimizer functions* $x^* : \Theta \to \mathbb{R}^n$ *and* $y^* : \Theta \to \{0,1\}^p$ *such that*

$$(x^*(\theta), y^*(\theta)) \in \arg\min_{(x,y)} \left\{ c^T x + d^T y \mid (x,y,\theta) \in P \right\},$$

that are respectively PWA and piecewise constant.

7.3.4 Explicit solution to constrained linear control allocation

If we consider (7.1), then under our assumption on U we have that $\mathcal{T}$ is a polygon. Moreover, if $l \in \{1,\infty\}$ and U is a polyhedron, then (7.1) can be written as a pLP (7.2) by viewing τ as parameters. Similarly if $l = 2$ and U is a polyhedron we have a pQP (7.3). Finally, if U is a polygon, we have that (7.1) is a pMILP ($l \in \{1,\infty\}$) or pMIQP ($l = 2$).

7.4 Decomposing allocation problems

In this section we propose the decomposing scheme for constrained linear control allocation. We first treat the case where U (or its inner approximation) is convex. In the sequel, if $u \in U \subseteq \mathbb{R}^n$ and $\mathcal{I} \subseteq \mathbb{N}_n$ is an index set, then $U_\mathcal{I}$ denotes the set $U_\mathcal{I} := \left\{ u_\mathcal{I} \in \mathbb{R}^{|\mathcal{I}|} \,\middle|\, \exists u_{\mathbb{N}_n \setminus \mathcal{I}} : (u_\mathcal{I}, u_{\mathbb{N}_n \setminus \mathcal{I}}) \in U \right\}$. Moreover, if $\mathcal{J} \subseteq \mathbb{N}_n$ is another index set such that $\mathcal{I} \cap \mathcal{J} = \emptyset$, then with some abuse of notation, $U_\mathcal{I}(u_\mathcal{J})$ denotes the set

$$U_\mathcal{I}(u_\mathcal{J}) := \left\{ u_\mathcal{I} \in \mathbb{R}^{|\mathcal{I}|} \,\middle|\, \begin{array}{l} \exists u_{\mathbb{N}_n \setminus (\mathcal{I} \cup \mathcal{J})} : \\ (u_\mathcal{I}, u_\mathcal{J}, u_{\mathbb{N}_n \setminus (\mathcal{I} \cup \mathcal{J})}) \in U \end{array} \right\}$$

Definition 7.1 (Non-interacting actuators) *Let the controls* u *be constrained to* U*. Given two actuators,* A *and* B*, and corresponding index sets* $\mathcal{A}$ *and* $\mathcal{B}$ *such that* $u_\mathcal{A} \in \mathbb{R}^{|\mathcal{A}|}$ *and* $u_\mathcal{B} \in \mathbb{R}^{|\mathcal{B}|}$ *are the control inputs to actuator* A *and* B*, respectively. The actuators* A *and* B *are said to be* non-interacting *if and only if*

$$U_\mathcal{A}(u_\mathcal{B}) = U_\mathcal{A},\ \forall u_\mathcal{B} \in U_\mathcal{B}, \quad \textit{and}$$
$$U_\mathcal{B}(u_\mathcal{A}) = U_\mathcal{B},\ \forall u_\mathcal{A} \in U_\mathcal{A}.$$

Remark 7.1 *For linear constrained control allocation problems non-interacting actuators means that by changing the control input for actuator A, the constraints on the control inputs to actuator B are unchanged. Note however, that the controls may still be coupled through the linear relationship $Bu = \bar{\tau}$. In addition, we would like to point out that the linear version of the control allocation problem is often an approximation to a non-linear relationship $\bar{\tau} = g(x, u, t)$. If this is the case, then one should add additional restrictions on the interactions between the actuators in the sense that: the contribution to the generalized forces from actuator A is unchanged for all possible contributions from actuator B. This captures non-linear interaction between the actuators, for example, for marine vessels it is not uncommon to loose effect from thruster A if thruster B affects the flow pattern around thruster A.*

Definition 7.2 (Non-interacting actuator partition) *Consider a set of actuators $\mathcal{P} := \{p_i \mid i \in \mathcal{I}\}$ and the partition $\{P_j \mid j \in \mathcal{J}\}$ of $\mathcal{P}$. If for every pair $(p_A, p_B) \in P_k \times P_j$, $\forall k \in \mathcal{J}$ and $\forall j \in \mathcal{J}$, $k \neq j$, (p_A, p_B) are non-interacting actuators, then $\{P_j \mid j \in \mathcal{J}\}$ is said to be a* non-interacting actuator partition *of $\mathcal{P}$.*

In the sequel, let $\{P_j \mid j \in \mathcal{J}\}$ denote a non-interacting actuator partition of $\mathcal{P}$ and $\left\{\mathcal{J}^j \mid j \in \mathcal{J}\right\}$ be the corresponding collection of index sets, i.e. the control inputs to the actuators in P_j are $u_{\mathcal{J}^j}$. It is immediate that we can write $\mathbb{P}(\cdot)$ as

$$
\begin{aligned}
J^*(\tau) &:= \min_{(u,s)} \left\{ \|Qs\|_l + \|Ru\|_l \mid (u, s) \in \mathcal{Y}(\tau) \right\}, \\
\mathcal{Y}(\tau) &:= \left\{ (u, s) \;\middle|\; \begin{array}{l} s + \sum_{j\in\mathcal{J}} B_{(*,\mathcal{J}^j)} u_{\mathcal{J}^j} = \tau \\ u_{\mathcal{J}^j} \in U_{\mathcal{J}^j}, \quad \forall j \in \mathcal{J} \end{array} \right\}.
\end{aligned}
$$

In the next section we re-formulate the above problem to obtain a master- and a set of sub-problems.

7.4.1 Decomposing Constrained Linear Control Allocation over Convex Sets

In this section we propose the method for decomposing the allocation problem. $\mathbb{M}(\cdot)$ will denote the master problem and a sub-problem will be denoted $\mathbb{S}_j(\cdot)$ for $j \in \mathcal{J}$. The master problem is defined as

$$
\mathbb{M}(\tau): \quad V^*(\tau) := \min_{\{s, \tau^1, \ldots, \tau^{|\mathcal{J}|}\}} \|Qs\|_l + \sum_{j\in\mathcal{J}} \|H^j \tau^j\|_l \tag{7.5a}
$$

$$
\text{s.t.} \quad s + \sum_{j\in\mathcal{J}} \tau^j = \tau, \tag{7.5b}
$$

$$
\tau^j \in \mathcal{T}^j \subset \mathbb{R}^r, \;\; \forall j \in \mathcal{J}, \tag{7.5c}
$$

where $H^j = (H^j)^T \geq 0 \in \mathbb{R}^{r\times r}$ are suitably defined weight matrices and

$$
\mathcal{T}^j := \left\{ \tau^j \in \mathcal{T} \subseteq \mathbb{R}^r \;\middle|\; \exists y_{\mathcal{J}^j} : y_{\mathcal{J}^j} \in \mathcal{N}_j(\tau^j) \right\} \tag{7.6a}
$$

$$
\mathcal{N}_j(\tau^j) := U_{\mathcal{J}^j} \cap \left\{ y_{\mathcal{J}^j} \in \mathbb{R}^{|\mathcal{J}^j|} \;\middle|\; B_{(*,\mathcal{J}^j)} y_{\mathcal{J}^j} = \tau^j \right\}. \tag{7.6b}
$$

It is clear that $\mathcal{T}^j$ is the set of all possible generalized forces (virtual controls) that can be generated by the actuators in the j^{th} element of the actuator partition.

For a given $j \in \mathcal{J}$, the j^{th} sub-problem is defined as:

$$\mathbb{S}_j(\tau^j): \quad V_j^*(\tau^j) := \min_{y_{\mathcal{J}^j}} \left\{ \|R_{(\mathcal{J}^j,\mathcal{J}^j)} y_{\mathcal{J}^j}\|_l \;\middle|\; y_{\mathcal{J}^j} \in \mathcal{N}_j(\tau^j) \right\}. \tag{7.7}$$

For notational simplicity we let $\{s^{\mathbb{M}}(\cdot), \tau^1(\cdot), \ldots, \tau^{|\mathcal{J}|}(\cdot)\}$ denote a set of single valued, continuous, selections for $\mathbb{M}(\cdot)$. Moreover, $\{y_{\mathcal{J}^1}(\cdot), \ldots, y_{\mathcal{J}^{|\mathcal{J}|}}(\cdot)\}$ are single valued, continuous, selections for $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$. By a solution to $\mathbb{M}(\cdot)$ and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ we mean the function $y^* : \mathbb{R}^r \to \mathbb{R}^m$ defined as

$$y^*(\tau) := \operatorname{sort}\left(\{y_{\mathcal{J}^1}(\tau^1(\tau)), \ldots, y_{\mathcal{J}^{|\mathcal{J}|}}(\tau^{|\mathcal{J}|}(\tau))\}\right),$$

i.e. $y^*(\cdot)$ has the same dimension and ordering as $u^*(\cdot)$.

Lemma 7.1 *Consider* $\mathbb{M}(\cdot)$, $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ *and* (7.1). *We have that if* $(s^{\mathbb{M}}(\cdot), y^*(\cdot))$ *is a feasible solution to* $\mathbb{M}(\cdot)$ *and* $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$*, then* $(s^{\mathbb{M}}(\cdot), y^*(\cdot))$ *is feasible for* (7.1). *Moreover, if* $R_{(\mathcal{J}^j,\mathcal{J}^j)} = R^T_{(\mathcal{J}^j,\mathcal{J}^j)} \geq 0$ *for all* $j \in \mathcal{J}$*, then we have*

1. *if* $l = 2$*, then* $V^* : \mathbb{R}^r \to \mathbb{R}$ *and each* $V_j^* : \mathbb{R}^r \to \mathbb{R}$*,* $j \in \mathcal{J}$ *are piecewise quadratic, convex, and continuous.*

2. *if* $l \in \{1, \infty\}$*, then* $V^* : \mathbb{R}^r \to \mathbb{R}$ *and each* $V_j^* : \mathbb{R}^r \to \mathbb{R}$*,* $j \in \mathcal{J}$ *are PWA, convex, and continuous.*

PROOF: *The feasible sets are equal by construction. The properties of* $V^*(\cdot)$ *and each* $V_j^*(\cdot)$ *follows from noting that the problems are pQPs for* $l = 2$ *and pLPs for* $l \in \{1, \infty\}$ *(Theorems 7.1 and 7.2).*

□

How to choose the weight matrices H^j, $j \in \mathcal{J}$ such that the solution function $(s^{\mathbb{M}}(\cdot), y^*(\cdot))$ is not only feasible for (7.1), but also as close to optimal as possible is non-trivial, however, we will not elaborate on this, since exact solutions to (7.1) can be obtained by imposing a natural assumption on R, which is stated below. In Section 7.5 we will show by example that if the problem has certain symmetry properties, the weight matrices $\{H^j \mid j \in \mathcal{J}\}$ are easy to choose.

Assumption 7.2 *Consider* (7.1). *For the weighting matrix* R*, set of actuators* $\mathcal{P} := \{p_1, \ldots, p_I\}$*, and non-interacting actuator partition* $\{P_j \mid j \in \mathcal{J}\}$ *of* $\mathcal{P}$*, we assume that* $R_{(\mathcal{J}^i,\mathcal{J}^j)} = R_{(\mathcal{J}^j,\mathcal{J}^i)} = 0$ *if* $i \neq j$*. Moreover, we assume that for each* $j \in \mathcal{J}$ *we have that* $R_{(\mathcal{J}^j,\mathcal{J}^j)} = R^T_{(\mathcal{J}^j,\mathcal{J}^j)} \geq 0$.

Lemma 7.2 *Assumption 7.2 has the consequence that for* $l \in \{1, 2\}$ *we have:*

$$\|Ru\|_l = \sum_{j \in \mathcal{J}} \|R_{(\mathcal{J}^j,\mathcal{J}^j)} u_{\mathcal{J}^j}\|_l, \tag{7.8}$$

and for $l = \infty$ we have

$$\|Ru\|_\infty = \max_{j\in\mathcal{J}}\{\|R_{(\mathcal{J}^j,\mathcal{J}^j)}u_{\mathcal{J}^j}\|_\infty\}$$

PROOF: *For the quadratic norm we have*

$$\begin{aligned}\|Ru\|_2 := u^T Ru = &\begin{bmatrix}u_{\mathcal{J}^1}^T & \dots & u_{\mathcal{J}^{|\mathcal{J}|}}^T\end{bmatrix}\\ &\operatorname{diag}\left(R_{(\mathcal{J}^1,\mathcal{J}^1)},\dots,R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})}\right)\begin{bmatrix}u_{\mathcal{J}^1}^T & \dots & u_{\mathcal{J}^{|\mathcal{J}|}}^T\end{bmatrix}^T\\ = &\|R_{(\mathcal{J}^1,\mathcal{J}^1)}u_{\mathcal{J}^j}\|_2 + \dots + \|R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})}u_{\mathcal{J}^{|\mathcal{J}|}}\|_2,\end{aligned}$$

and for $l = 1$ we recall that if $i \notin \mathcal{J}^j$ then $R_{(i,\mathcal{J}^j)}u_{\mathcal{J}^j} = 0$, and hence

$$\begin{aligned}\|Ru\|_1 &= \sum_{p\in\mathbb{N}_m}\left|\sum_{q\in\mathbb{N}_m}R_{(p,q)}u_q\right|\\ &= \sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{J}^j}\left|\sum_{q\in\mathcal{J}^j}R_{(p,q)}u_q\right| = \sum_{j\in\mathcal{J}}\|R_{(\mathcal{J}^j,\mathcal{J}^j)}u_{\mathcal{J}^j}\|_1.\end{aligned}$$

For $l = \infty$, Assumption 7.2 clearly leads to

$$\begin{aligned}\|Ru\|_\infty &= \max\left\{\left|\sum_{p\in\mathbb{N}_m}R_{(1,p)}u_p\right|,\dots,\left|\sum_{p\in\mathbb{N}_m}R_{(m,p)}u_p\right|\right\}\\ &= \max\left\{\|R_{(\mathcal{J}^1,\mathcal{J}^1)}u_{\mathcal{J}^1}\|_\infty,\dots,\|R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})}u_{\mathcal{J}^{|\mathcal{J}|}}\|_\infty\right\}.\end{aligned}$$

□

In the following two theorems we are only concerned with the set $\mathcal{T}^* \subseteq \mathcal{T}$ in which $s^*(\tau) = 0$ i.e. we have also assumed that $\|Qs\|_l$ is an exact penalty function for (7.1) (Johansen, Fossen, and Tøndel 2005).

Theorem 7.4 *Consider $\mathbb{M}(\cdot)$, $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ and* (7.1)*, and let $l \in \{1, 2\}$. By changing the master problem to*

$$\mathbb{M}_e(\tau): \quad V^*(\tau) := \min_{\{t,\tau^1,\dots,\tau^{|\mathcal{J}|}\}} t \tag{7.9a}$$

$$\text{s.t.} \quad t \geq \sum_{j\in\mathcal{J}} V_j^*(\tau^j) \tag{7.9b}$$

$$\sum_{j\in\mathcal{J}}\tau^j = \tau, \tag{7.9c}$$

$$\tau^j \in \mathcal{T}^j, \ \ \forall j \in \mathcal{J}, \tag{7.9d}$$

we have that $J^*(\tau) = V^*(\tau)$ for all $\tau \in \mathcal{T}^*$, $(y^*(\cdot), 0) \in \mathcal{Y}^*(\cdot)$ and $\mathbb{M}_e(\cdot)$ is a convex optimization problem.

PROOF: *Convexity of $\mathbb{M}_e(\cdot)$ follows easily by noting that all constraints are linear except* (7.9b)*, which is also convex due to Theorem 7.1 and 7.2. Note first that by construction $y(\cdot) = u^*(\cdot)$ is feasible for* (7.9) *and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ and that $J(u^*(\tau), 0, \tau) = J^*(\tau)$, $\forall \tau \in \mathcal{T}^*$. We also have*

$$\begin{aligned} V^*(\tau) = t^*(\tau) &\geq \sum_{j\in\mathcal{J}} V_j^*(\tau^j) = \sum_{j\in\mathcal{J}} \|R_{(\mathcal{J}^j,\mathcal{J}^j)} y_{\mathcal{J}^j}(\tau^j)\|_l \\ &= \|Ry^*(\tau)\|_l = J(y^*(\tau), 0, \tau). \end{aligned}$$

Hence, if $(y^*(\tau), 0) \notin \mathcal{Y}^*(\tau)$ then $J(y^*(\tau), 0, \tau) > J(u^*(\tau), 0, \tau)$, which contradicts optimality since $u^*(\tau)$ is feasible for (7.9) *and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$.*

$\square$

For $l = \infty$ the master problem has to be modified slightly as demonstrated by the following theorem:

Theorem 7.5 *Consider $\mathbb{M}(\cdot)$, $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ and* (7.1)*, and let $l = \infty$. By changing the master problem to*

$$\begin{aligned} \mathbb{M}_e(\tau): \quad V^*(\tau) := &\min_{\{t,\tau^1,\ldots,\tau^{|\mathcal{J}|}\}} t \\ \text{s.t.} \quad & t \geq V_j^*(\tau^j) \quad \forall j \in \mathcal{J} \\ & \sum_{j\in\mathcal{J}} \tau^j = \tau, \\ & \tau^j \in \mathcal{T}^j, \ \ \forall j \in \mathcal{J}, \end{aligned}$$

we have that $J^*(\tau) = V^*(\tau)$ for all $\tau \in \mathcal{T}^*$, $(y^*(\cdot), 0) \in \mathcal{Y}^*(\cdot)$ and $\mathbb{M}_e(\cdot)$ is a convex optimization problem.

PROOF: *The proof is identical to the proof of Theorem 7.4, except*

$$\begin{aligned} V^*(\tau) = t^*(\tau) \geq \max\{&\|R_{(\mathcal{J}^1,\mathcal{J}^1)} y_{\mathcal{J}^1}(\tau^1)\|_\infty, \ldots, \\ &\|R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})} y_{\mathcal{J}^{|\mathcal{J}|}}(\tau^{|\mathcal{J}|})\|_\infty\} = \|Ry^*(\tau)\|_\infty. \end{aligned}$$

$\square$

Remark 7.2 *For $l \in \{1, \infty\}$ it is straightforward to solve $\mathbb{M}_e(\cdot)$ explicitly since $\mathcal{T}^*$ can be expressed as a union of polyhedra, and in each of these $\mathbb{M}_e(\cdot)$ is a pLP. On the other hand, for $l = 2$, there is currently no available algorithm for obtaining an exact, explicit, solution of* (7.9).

7.4.2 Decomposing Constrained Linear Control Allocation over Non-convex Sets

If the set of attainable forces U is a non-convex polygon, $U = \cup_{i\in\mathcal{I}} U^i$, the optimization problem (7.1) is no longer convex. However, the set $\mathcal{Y}$ can be written as

$$\mathcal{Y} := \left\{(u, s, \tau) \;\middle|\; Bu + s = \tau, u \in U^1 \vee \cdots \vee U^{|\mathcal{I}|}\right\},$$

and (7.1) becomes a parametric mixed integer program. In this case the problem can also be decomposed into $\mathbb{M}(\cdot)$ and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ for the purpose of obtaining sub-optimal solutions. The main difference being that the sets $\{\mathcal{T}^j \mid j \in \mathcal{J}\}$ are more computationally demanding to obtain, since $U_{\mathcal{J}^j} = \cup_{i\in\mathcal{I}} U^i_{\mathcal{J}^j}$. In the non-convex case, both the master- and sub-problems are parametric mixed integer programs. For brevity, we do not consider this case in detail.

7.4.3 Reconfigurable control allocation

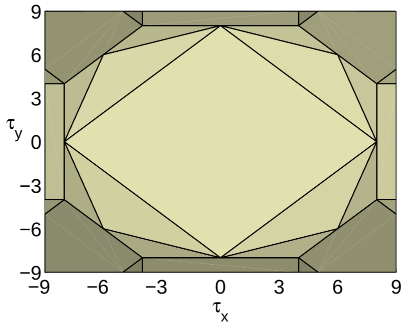
If the master-problem is solved online, we can obtain a tradeoff between the benefits and drawbacks of the explicit solution when the scheme is applied to reconfigurable control allocation. By introducing extra parameters in the allocation problem, as described in Section 7.2.3, the complexity of $u^*(\cdot)$ may increase to the level where the explicit scheme is rendered unusable. By solving the master problem online and the sub-problems explicitly, the control actions from actuator group P_j can be limited simply by changing the constraints on τ^j.

7.5 Numerical Example

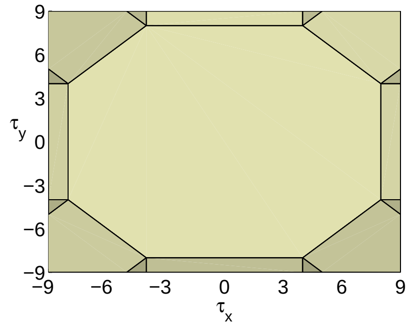
Note that in this section we use slightly different indexing of the variables. Consider the following allocation problem:

$$\min_{\{u,s\}} \left\{ u^T R u + s^T Q s \;\middle|\; \begin{aligned} s_x + \textstyle\sum_{i=1}^4 u_{i,x} &= \tau_x \\ s_y + \textstyle\sum_{i=1}^4 u_{i,y} &= \tau_y \\ |u_{i,x}| + |u_{i,y}| &\le 2, \\ i = 1, 2 & \\ |u_{i,j}| &\le 2, \\ i = 3, 4, j = x, y & \end{aligned} \right\} \tag{7.10}$$

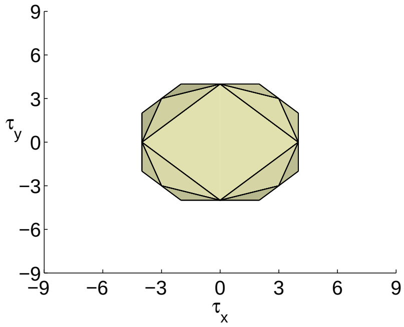
where $R = \mathrm{diag}(1, 1, \dots, 1)$ and $Q = \mathrm{diag}(10^3, 10^3)$. In this problem we have four actuators $\mathcal{P} := \{p_1, \dots, p_4\}$, where the i^{th} actuator has two controls, $u_{i,x} \in \mathbb{R}$ and $u_{i,y} \in \mathbb{R}$, and we have two generalized forces, $\tau_x \in \mathbb{R}$ and $\tau_y \in \mathbb{R}$. Moreover, we define $u := [u_{1,x}, u_{1,y}, \dots, u_{4,x}, u_{4,y}]^T$ and $\tau := [\tau_x, \tau_y]^T$. Looking at the constraints it can be straightforwardly verified that all the actuators are non-interacting. In this example we show two different actuator partitions; first choose the following non-interacting actuator partition $\{P_1, P_2\}$, where $P_1 =$



(a) The set of polyhedra representing the solution $v^*(\cdot) := (s^*(\cdot), u^*(\cdot))$ to (7.10).



(b) The set of polyhedra representing the solution $z^*(\cdot)$ to (7.11) with the actuator partition $P_1 = \{p_1, p_3\}$ and $P_2 = \{p_2, p_4\}$.



(c) Set of polyhedra representing the solutions $u^1(\cdot)$ and $u^2(\cdot)$ to the first sub-problem (7.12) and the second sub-problem.

Figure 7.1: Explicit solutions with the actuator partition $P_1 = \{p_1, p_3\}$ and $P_2 = \{p_2, p_4\}$.

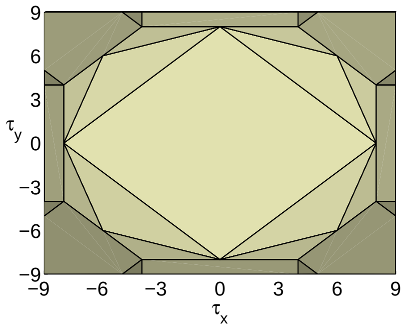
$\{p_1, p_3\}$ and $P_2 = \{p_2, p_4\}$, yielding $u^1 := [u_{1,x}, u_{1,y}, u_{3,x}, u_{3,y}]^T$, and $u^2 := [u_{2,x}, u_{2,y}, u_{4,x}, u_{4,y}]^T$. Following the proposed procedure we get the following master problem

$$\min_{\tau^1,\tau^2,s} \left\{ s^T Q s + \sum_{j=1}^{2} (\tau^j)^T H^j \tau^j \;\middle|\; \begin{array}{l} \tau = s + \tau^1 + \tau^2 \\ \tau^j \in \mathcal{T}^j, j = 1, 2 \end{array} \right\} \tag{7.11}$$

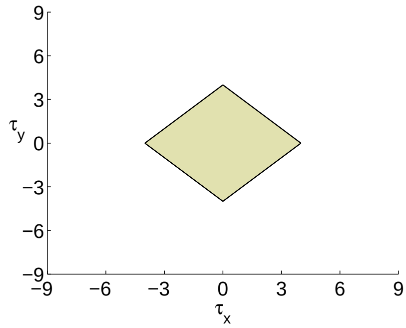
where

$$\mathcal{T}^1 = \left\{ \tau^1 \in \mathcal{T} \;\middle|\; \exists u^1 : \begin{array}{c} B^1 u^1 = \tau^1 \\ |u_{1,x}| + |u_{1,y}| \le 2 \\ |u_{3,i}| \le 2, i = x, y, \end{array} \right\},$$

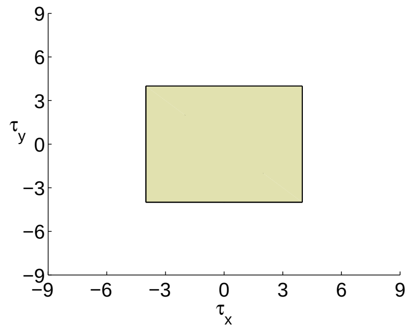
where B^1 consists of the column in B that multiply with u^1, and $\mathcal{T}^2$ is found by replacing the appropriate indices, which yields an identical set, i.e. $\mathcal{T}^1 = \mathcal{T}^2$. More-



(a) The set of polyhedra representing the solution $z^*(\cdot)$ to (7.11) with the actuator partition $P_1 = \{p_1, p_2\}$ and $P_2 = \{p_3, p_4\}$.



(b) Set of polyhedra representing the solution $u^1(\cdot)$ for the first sub-problem, defined by actuator group $P_1 = \{p_1, p_2\}$.



(c) Set of polyhedra representing the solution $u^2(\cdot)$ for the second sub-problem, defined by actuator group $P_2 = \{p_3, p_4\}$.

Figure 7.2: Explicit solutions with the actuator partition $P_1 = \{p_1, p_2\}$ and $P_2 = \{p_3, p_4\}$.

over, since we have a symmetrical problem, we choose $H^1 = H^2 = \text{diag}(1, 1)$. The first sub-problem becomes

$$\min_{u^1} \left\{ (u^1)^T \,\text{diag}(1,1,1,1) u^1 \;\middle|\; \begin{array}{c} u_{1,x} + u_{3,x} = \tau_x^1 \\ u_{1,y} + u_{3,y} = \tau_y^1 \\ |u_{1,x}| + |u_{1,y}| \le 2 \\ |u_{3,i}| \le 2,\ i = x, y \end{array} \right\} \tag{7.12}$$

and the second sub-problem is found by replacing the appropriate indices. Let the function $z^*(\cdot) := [s^*(\cdot)^T \ (\tau^1(\cdot))^T \ (\tau^2(\cdot))^T]^T$ denote the PWA solution to the master problem, and $u^1(\cdot)$ and $u^2(\cdot)$ be the solutions to the two sub-problems. The polyhedra that $u^*(\cdot)$, $z^*(\cdot)$ and $u^1(\cdot)$ are defined on are depicted in Figures 7.1(a)-7.1(c), respectively. Note also that $u^2(\cdot) = u^1(\cdot)$. Figures 7.2(a)-7.2(c) depicts the solutions for the master and two subproblems for the the actuator partition $P_1 = \{p_1, p_2\}$ and $P_1 = \{p_3, p_4\}$. Considering the first actuator partition it is apparent that an explicit solution to the problem can be found by solving two smaller pQPs (the two sub-problems are identical), but more importantly, one can choose to solve either of the problems on-line, allowing a tradeoff between the online computation time and the required storage space. From this example we see that the proposed strategy provides great flexibility. We have the following alternatives for the first actuator partition:

1. Solving master and sub-problems online.
2. Solving the master problem online and one subproblem explicitly, and since the sub-problems are identical this only yields 13 stored polyhedra.
3. Solving the master problem explicitly and one of the sub-problems online.
4. Solving both the master- and sub-problems explicitly.

Obviously, we have similar alternatives for the second actuator partition. Finally, note that for the first actuator partition we have $u^1(\cdot) = u^2(\cdot)$ and that the solution to the original problem also has $u^1 = u^2$ (a strictly convex problem where the constraints and weights on u^1 and u^2 are identical.) Thus, if we choose $H^1 = H^2 = \mathrm{diag}(1,1)$, we have a strictly convex master-problem whose solution is unique ($\tau^1(\cdot) = \tau^2(\cdot)$), hence, $(u^1(\tau^1(\tau)))^T u^1(\tau^1(\tau))) + (u^2(\tau^2(\tau)))^T u^2(\tau^2(\tau)) = (u^*(\tau))^T u^*(\tau)$, i.e. the solution is optimal also for the original problem.

7.6 Decomposition Strategy applied to a Thruster Controlled Floating Platform: Non-convex Formulation

Please note that this section is not self contained in the sense that it extends the case study in Chapter 6. We also adopt the notation from Chapter 6. Consider the thruster-controlled platform from the previous chapter where the virtual controller (typically a dynamic positioning controller) specifies surge, sway and yaw forces. In this section we no longer assume that each approximation $\mathcal{U}_i$ of the restricted ATR $\bar{\mathcal{C}}_i$ is a convex set. In particular we let $\mathcal{U}_i = \mathcal{U}_{i,a} \cup \mathcal{U}_{i,b}$ where $\mathcal{U}_{i,a}$ and $\mathcal{U}_{i,b}$ are convex polyhedra. See Figure 7.3(a) for an illustration. For this case, our allocation problem becomes:

$$J^*(\tau) := \min_{u,s} \left\{ \frac{1}{2}\left(u^T R u + s^T Q s\right) \,\middle|\, (u,s,\tau) \in \mathcal{Z} \right\}, \tag{7.13a}$$

$$\mathcal{Z} := \left\{ (u,s,\tau) \in \mathbb{R}^{16} \times \mathbb{R}^3 \times \mathbb{R}^3 \,\middle|\, \begin{array}{c} Bu = \tau + s, u_i \in \mathcal{U}_{i,a} \vee \mathcal{U}_{i,b} \\ i \in \{1,2,\ldots,8\} \end{array} \right\}, \tag{7.13b}$$

where $R = I$ and $Q = 10^3 \times I$. The approach presented in (Johansen, Fuglseth, Tøndel, and Fossen 2003) to solve a similar problem is to compute an explicit solution to a set of convex problems and compare the solution online. That is, u_i is constrained to either $\mathcal{U}_{i,a}$ or $\mathcal{U}_{i,b}$ and all such combinations are computed. This is clearly not tractable in our case as this will lead to $2^8 = 256$ different combinations if symmetry is not exploited. Currently no algorithm that solves a pMIQP exactly is presented in the literature, hence, we need another approach if we seek an explicit solution.

We now utilize the decomposition strategy, however we seek to have 2 parameters in the sub-problems and 3 in the master problem and therefore introduce the following terms in order to carry out the reformulation:

$$\begin{aligned}
z_{1,x} &:= u_{1,x} + u_{2,x},\\
z_{1,y} &:= u_{1,y} + u_{2,y},\\
z_{2,x} &:= u_{3,x} + u_{4,x},\\
z_{2,y} &:= u_{3,y} + u_{4,y},\\
&\vdots\\
z_{4,x} &:= u_{7,x} + u_{8,x},\\
z_{4,y} &:= u_{7,y} + u_{8,y},
\end{aligned}$$

and let $z_i := [z_{i,x}\ z_{i,y}]^T$ and $z := [z_1^T\ \ldots\ z_4^T]^T$ (z_i are the surge and sway forces from the i^{th} leg). If we were only considering surge and sway forces, we could now write $Bu = \tau$ as

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} z = \tau.$$

The yaw component is, as we recall from (6.5), defined by

$$N := -l_{1,y}u_{1,x} + l_{1,x}u_{1,y} - l_{2,y}u_{2,x} + l_{2,x}u_{2,y}\ \ldots\ - l_{8,y}u_{8,x} + l_{8,x}u_{8,y},$$

and consequently not as easy to express in terms of z_is. Consider first leg one and its contribution $\tau_{1,N} = N_1 + N_2$ to the yaw moment:

$$\tau_{1,N} := -l_{1,y}u_{1,x} + l_{1,x}u_{1,y} - l_{2,y}u_{2,x} + l_{2,x}u_{2,y}. \tag{7.14}$$

Noting that $|l_{1,x}| = |l_{2,y}|$ and $|l_{2,x}| = |l_{1,y}|$, we see that the radiuses from the origin of the vessel-fixed coordinate system to the two thrusters are equal. Clearly, there must then exists an angle β such that if we rotate the vector $r_1 = [l_{1,x}\ l_{1,y}]^T$ by β and $r_2 = [l_{2,x}\ l_{2,y}]^T$ by $-\beta$, the rotated vectors r_1^r and r_2^r would coincide, i.e. $r_1^r = r_2^r =: [l_{1,x}^r\ l_{1,y}^r]^T$. We can now rewrite (7.14) as two equations:

$$\begin{aligned}
\tau_{1,N} &= -l_{1,y}^r z_{1,x} + l_{1,x}^r z_{1,y}, &\text{(7.15a)}\\
-l_{1,y}^r z_{1,x} + l_{1,x}^r z_{1,y} &= -l_{1,y}u_{1,x} + l_{1,x}u_{1,y} - l_{2,y}u_{2,x} + l_{2,x}u_{2,y}. &\text{(7.15b)}
\end{aligned}$$

We have rewritten (7.14) in order to enforce (7.15a) in the master problem and (7.15b) in the sub-problem.

Assume temporarily that the set of surge and sway forces that can be generated by the thrusters on the i^{th} leg is given by $\mathcal{T}_i$ (we will return to the computation of these sets). The master problem is then defined as:

$$\begin{aligned}
\mathbb{M}(\tau): \quad V^*(\tau) &:= \min_{z,s} \left\{ \frac{1}{2}\left(z^T H z + s^T Q s\right) \,\middle|\, (z,s,\tau) \in \mathcal{Z} \right\} \\
\mathcal{Z} &:= \left\{ (z,s,\tau) \in \mathbb{R}^8 \times \mathbb{R}^3 \times \mathbb{R}^3 \,\middle|\, \begin{array}{c} \hat{B}z = \tau + s, z_i \in \mathcal{T}_i \\ i \in \{1,2,\ldots,4\} \end{array} \right\}, \\
\hat{B} &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -l^r_{1,y} & l^r_{1,x} & -l^r_{2,y} & l^r_{2,x} & -l^r_{3,y} & l^r_{3,x} & -l^r_{4,y} & l^r_{4,x} \end{bmatrix},
\end{aligned}$$

where $H = H^T > 0$ is a suitably defined weight matrix.

The sub-problem for leg 1 is then defined as follows:

$$\begin{aligned}
\mathbb{S}_1(z_1): \quad J^*(z_1) &:= \min_{u} \left\{ \frac{1}{2}\left(u^T R u\right) \mid (u, z_1) \in \mathcal{P} \right\} \\
\mathcal{P} &:= \left\{ (u, z_1) \in \mathbb{R}^4 \times \mathbb{R}^2 \,\middle|\, \begin{array}{c} \bar{B}u = Wz_1, u_i \in \mathcal{U}_{i,a} \vee \mathcal{U}_{i,b}, \\ i \in \{1,2\} \end{array} \right\}, \\
\bar{B} &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -l_{1,y} & l_{1,x} & -l_{2,y} & l_{2,x} \end{bmatrix}, \\
W &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -l^r_{1,y} & l^r_{1,x} \end{bmatrix},
\end{aligned}$$

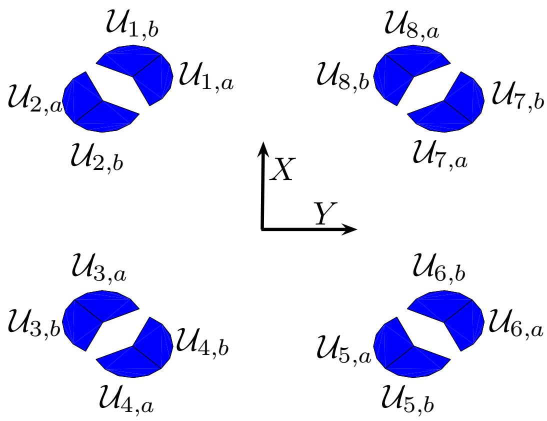
where $u_i := [u_{i,x}\ u_{i,y}]^T$, $u := [u_{1,x}\ u_{1,y}\ u_{2,x}\ u_{2,y}]^T$ and $R = I$. Define similarly the problems $\mathbb{S}_2(z_2)$ (with $u := [u_{3,x}\ u_{3,y}\ u_{4,x}\ u_{4,y}]^T$ and $u_i \in \mathcal{U}_{i,a} \vee \mathcal{U}_{i,b},\ i \in \{3,4\}$) to $\mathbb{S}_4(z_4)$.

The problem $\mathbb{S}_1(\cdot)$ is much simpler to solve than the original problem as only four combinations must be considered.

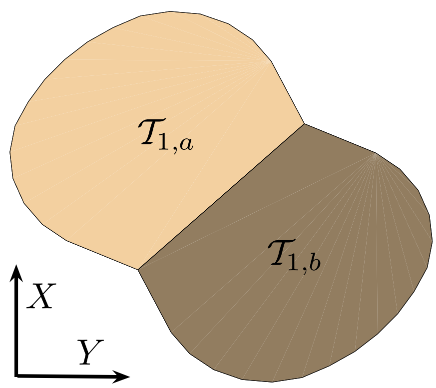
We now consider how to obtain $\mathcal{T}_i$. Clearly, the set $\mathcal{T}_1$ is a non-convex set given by

$$\begin{aligned}
\mathcal{T}_1 &= (\mathcal{U}_{1,b} \oplus \mathcal{U}_{2,a}) \cup (\mathcal{U}_{1,b} \oplus \mathcal{U}_{2,b}) \cup (\mathcal{U}_{1,a} \oplus \mathcal{U}_{2,a}) \cup (\mathcal{U}_{1,a} \oplus \mathcal{U}_{2,b}) \\
&=: \mathcal{T}_{1,a} \cup \mathcal{T}_{1,b},
\end{aligned}$$

which is depicted in Figure 7.3(b). Thus the set $\mathcal{T}_1$ is represented by two convex polyhedra. In Figure 7.4 we have shown all the $\mathcal{T}_i$s. Brute force computations now yield 16 combinations of constraint sets for the master problem and 4 for the subproblems. As in the previous chapter, we can utilize symmetry such that we only consider 4 combinations in the master problem (for example leg 1 and 2) and 4 in the subproblems. Thus, with the decomposition strategy and symmetry the number of problems that must be solved is reduced from 256 to a total of 8. If we in addition approximate each $\mathcal{T}_i$ by a single inner convex polyhedron, a sub-optimal solution could be found by solving only 5 problems.



(a) Approximation of each ATR by a union of two convex polyhedra.



(b) Surge and sway forces that can be generated by thruster 1 and 2 approximated by a union of two convex polyhedra.

Figure 7.3: Non-convex ATRs.

7.7 Conclusion

We have proposed a decomposing strategy for linear constrained control allocation problems. The actuators are partitioned such that a sub-optimal solution can be found be solving a master- and a set of sub-problems. It has also been shown that the decomposing strategy can provide an optimal solution to some classes of allocation problems if the master problem is modified appropriately. The advantages

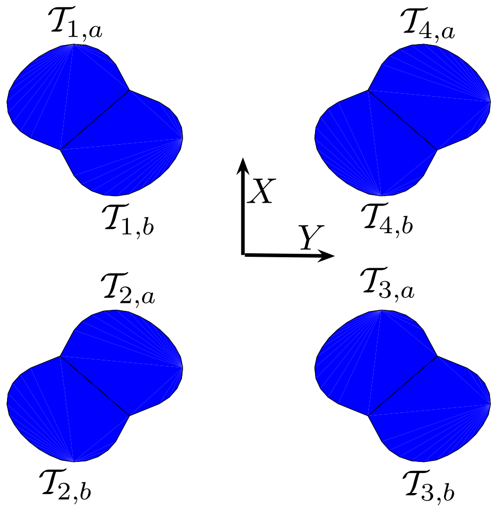


Figure 7.4: The sets $\mathcal{T}_i,\ i = 1, 2, 3, 4$.

with the scheme is that it allows the designer to choose a mix of online optimization and explicit solutions of the allocation problem, providing a tradeoff between the benefits and drawbacks of the explicit approach.

Chapter 8

Conclusions and future research

8.1 Concluding remarks on the chapters

Part one of this thesis presented some new theoretical results in parametric programming. In part two some new ways of utilizing parametric programming was proposed. In part three control allocation was considered and parametric programming was used on a thruster-controlled floating platform and experimental results were reported.

A novel method for obtaining continuous solutions and unique polyhedral representations of solutions to convex parametric quadratic programs was proposed in Chapter 2. The method was based on choosing the minimum norm solution and utilizing the normal cone optimality condition to characterize the parametric region in which the selection remained optimal. The method is applicable to almost all algorithms for convex pQPs (with trivial modifications) as the minimum norm problem can be solved as a separate strictly convex pQP in the parametric regions for which the solution is non-unique.

It was shown in (Spjøtvold 2005) that the facet-to-facet property does not hold for convex pQPs. In Chapter 3 we answered a harder question; does the property hold for strictly convex pQPs? The facet-to-facet property was shown not to hold for this problem, and consequently rendered some algorithms for pQPs without guarantees of correct traversal of the parameter space. A simple procedure based on combining the algorithms in (Tøndel, Johansen, and Bemporad 2003a) and (Bemporad, Morari, Dua, and Pistikopoulos 2002) was proposed to remedy this problem. Numerical results indicated that the proposed method was computationally more efficient than the algorithm in (Bemporad, Morari, Dua, and Pistikopoulos 2002) for problems whose solution consisted of a large number of critical regions.

A framework for obtaining explicit solutions to $\inf - \sup$ control of constrained discontinuous piecewise affine systems affected by state- and input-dependent disturbances was presented in Chapter 4. For this problem class, a solution might not exist, and consequently we proposed a method that obtained an optimal solution when one existed and a sub-optimal solution when one did not. The method utilized parametric piecewise linear programming. The degree of sub-optimality could be specified *a priori* and a bound on the total error was given. It turned out that the total error was linear in the prediction horizon.

The efficiency of explicit MPC depends on the ability to solve the point location problem at a high sampling rate. A method that utilized reachability analysis to reduce the size of the point location problem was presented in Chapter 5. The method intersected the one-step forward reach set with the solution partition in order to reduce the number of candidate regions that could contain the state at the next sampling instant. This approach reduced the computational load on the microchip. If, in addition, one knew with certainty that the state was contained in some strict subset of the state space at the initialization of the control scheme, the worst case search time could also be reduced. The reduction of the point location problem came at the expense of increased off-line processing and required storage space.

Constrained control allocation was one of the applications of parametric programming that was considered.

To demonstrate the usefulness of parametric programming as well as partially (see future research) solving a challenging problem, parametric programming was utilized to find an explicit solution to a control allocation problem for a thruster-controller floating platform in Chapter 6. The solution was implemented on a scale-model of a thruster-controlled platform and the experimental results were reported. The performance of the explicit allocation was satisfactory with regards to obtaining the desired generalized force. There is still work to be done to improve the allocation scheme further, which we will address in Section 8.2.

A decomposition approach was presented in Chapter 7 for the purpose of providing the engineer with a flexible design tool. The master- and sub-problems were constructed such that a mix of online optimization and explicit solutions could be utilized. The possibility of a tradeoff between online optimization and explicit solutions may in some cases be very beneficial, as for instance if the entire problem is too complex for a complete explicit solution. In some cases the flexibility may also allow added features in the control allocation problem, such as taking into account actuator and effector dynamics at a low level.

In general, it seems that the field of parametric programming has matured with the resurgence of interest caused by (Bemporad, Morari, Dua, and Pistikopoulos 2002). Challenges that are left include numerical stability and execution speed of the algorithms. It is questionable if parametric programming is extendable to more difficult problems than those already ”solved”. This is based on the fact that it is difficult to design robust algorithms even for parametric mixed-integer linear programs (see Section 8.2 for further discussion). Solving parametric non-linear programs exactly is rarely possible and obtaining approximations are often numerically difficult and computationally very demanding. Nevertheless, as demonstrated the last few years, parametric programming can be a powerful tool in many areas within control and the author suspects that there is a substantial number of areas not yet discovered.

8.2 Future Research Directions

Although this thesis did not give a complete overview of the field of parametric programming, we mention some research directions that may be investigated in the future.

Parametric Mixed-Integer Linear Programming

Parametric mixed-integer linear programs (pMILP) may seem like a relatively simple but computationally demanding task. A pMILP can be solved by solving a finite number of pLPs (total enumeration of the integer variables) and comparing the solutions. It has proven to be a difficult task to design a more efficient algorithm that does not require comparisons of the pLP solutions to find the optimal solution. There exists some algorithms for obtaining upper bounds on the solutions to pMILPs (Dua and Pistikopoulos 2000; Acevedo and Pistikopoulos 1997), however, these are numerically non-robust and also computationally very demanding. There are some problems with pMILPs that should be emphasized:

- Even though the solution is PWA, the restrictions are defined on open, closed or neither open nor closed polyhedra. In addition, the polyhedra may be lower-dimensional. If x_i^* is defined on the polyhedron P_i, we can have the situation where x_j^* is defined on P_j where P_j is a strict subset of P_i. Naturally, the same might be true for P_j and so on. Thus, the exploration strategy is not straightforward to design and representing the solution may also be problematic.

- The set of parameters that render the solution feasible and bounded may be disconnected and/or lower-dimensional. Obviously, this complicates the exploration strategy.

- The optimal integer solutions may be non-unique. This causes problems for integer-cut strategies where one searches for improved integer solutions (Dua and Pistikopoulos 2000).

The problems above are not trivial to overcome, however, the author suspects that designing an algorithm, which is more efficient than total enumeration of the integer solutions and finds an exact solution to pMILPs, is possible. One of the first steps should be to enforce uniqueness of the integer solution, perhaps utilizing some perturbation strategy analogous to the one in (Jones, Kerrigan, and Maciejowski 2007). The author also suspects that the exploration strategy in (Dua and Pistikopoulos 2000) is too numerically sensitive to yield an efficient algorithm, so designing an algorithm that searches through some kind of basis solutions is perhaps the approach that should be taken.

Control Allocation

The control allocation problem for the thruster-controlled platform presented in the thesis is challenging and there are several improvements that should be investigated. First of all rate constraints should be incorporated. Secondly, the cost function should better reflect actual energy consumption and mechanical tear and wear. These features are not simple to incorporate in the allocation problem as it is inherently non-convex and non-linear. Mixed-integer programming may be utilized to deal with the non-convex feasible set, however, the non-convex cost function may cause multiple optima and require a non-linear programming solver. It is also desirable to explicitly take into account the actuator dynamics in a predictive fashion. The author believes that one may be able to use a predictive approach with a mix of online optimization and explicit solutions.

Bibliography

Acevedo, J. and E. N. Pistikopoulos (1997). A multiparametric programming approach for linear process engineering problems under uncertainty. *Ind. Eng. Chem. Res. 36*, 717–728.

Arya, S., D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu (1998). An optimal algorithm for approximate nearest neighbour searching fixed dimensions. *Journal of the ACM 45*, 891–923.

Aubin, J. P. and H. Frankowska (1990). *Set-Valued Analysis*. Boston: Birkhäuser.

Aurenhammer, F. (1991). Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys 23*, 345–405.

Bank, B., J. Guddat, D. Klatte, B. Kummer, and K. Tammer (1983). *Non-linear Parametric Optimization*. Berlin: Birkhäuser.

Baotić, M. (2002). An efficient algorithm for multi-parametric quadratic programming. Technical Report AUT02-05, ETH Zürich, Institut für Automatik, Physikstrasse 3, CH-8092, Switzerland.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Bemporad, A., F. Borrelli, and M. Morari (2000a). Optimal Controllers for Hybrid Systems: Stability and Piecewise Linear Explicit Form. In *Proc. 39th IEEE Conf. on Decision and Control*, Sydney, pp. 1810–815.

Bemporad, A., F. Borrelli, and M. Morari (2000b). Piecewise linear optimal controllers for hybrid systems. In *Proc. American Contr. Conf.*, Chicago, IL, pp. 1190–1194.

Bemporad, A., F. Borrelli, and M. Morari (2002). Model predictive control based on linear programming - The explicit solution. *IEEE Trans. on Automatic Control 47*, 1974–1985.

Bemporad, A. and C. Filippi (2003). Suboptimal explicit RHC via approximate multiparametric quadratic programming. *Journal of Optimization Theory and Applications 117*, 9–38.

Bemporad, A. and C. Filippi (2006). An algorithm for approximate multi-parametric convex programming. *Computational Optimization and Applications 35*, 87–108.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica 35*, 407–427.

Bemporad, A., M. Morari, V. Dua, and E. N. Pistikopoulos (2002). The explicit linear quadratic regulator for constrained systems. *Automatica 38*, 3–20.

Berge, C. (1963). *Topological Spaces*. London: Oliver and Boyd Ltd.

Berkelaar, A., K. Roos, and T. Terlaky (1997). The optimal set and optimal partition approach to linear and quadratic programming. In T. Gal and H. Greenberg (Eds.), *Advances in Sensitivity Analysis and Parametric Programming*, Chapter 6, pp. 6.1–6.45. Boston, MA: Kluwer Academic Publishers.

Bertsekas, D. P., A. Nedic, and A. E. Ozdaglar (2003). *Convex Analysis and Optimization*. Nashua, NH: Athena Scientific.

Bertsekas, D. P. and I. B. Rhodes (1973). Sufficiently informative functions and the minimax feedback control of uncertain dynamic systems. *IEEE Transactions on Automatic Control 18*, 117–124.

Best, M. J. and N. Chakravarti (1990). Stability of linearly constrained convex quadratic programs. *Journal of Optimization Theory and Applications 64*, 43–53.

Best, M. J. and B. Ding (1972). On the continuity of the minimum in parametric quadratic programs. *Journal of Optimization Theory and Applications 86*, 245–250.

Bodson, M. (2002). Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics 25*, 703–711.

Bodson, M. and W. A. Pohlchuck (1998). Command limiting in reconfigurable flight control. *Journal of Guidance, Control, and Dynamics 21*, 639–646.

Böhm, V. (1975). On the continuity of the optimal policy set for linear programs. *SIAM Journal on Applied Mathematics 28*, 303–306.

Bordignon, K. A. and W. C. Durham (1995a). Closed-form solutions to constrained control allocation problems. *J. Guidance, Control and Dynamics 18*, 1000–1007.

Bordignon, K. A. and W. C. Durham (1995b). Null-space augmented solutions to constrained control allocation problems. In *Proc. AIAA Guidance, Navigation and Control Conference*, Baltimore, MD, pp. 328–333.

Borrelli, F. (2002). *Discrete Time Constrained Optimal Control*. Ph. D. thesis, Swiss Federal Institute of Technology, Zurich, Switzerland.

Borrelli, F., M. Baotić, A. Bemporad, and M. Morari (2001). Efficient on-line computation of constrained optimal control laws. In *Proc. 40th IEEE Conf. on Decision and Control*, Orlando, FL, pp. 1187–1192.

Borrelli, F., M. Baotić, A. Bemporad, and M. Morari (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica 41*, 1709–1721.

Borrelli, F., A. Bemporad, and M. Morari (2003). A geometric algorithm for multi-parametric linear programming. *Journal of Optimization Theory and Applications 118*, 515–540.

Buffington, J. M., D. F. Enns, and A. R. Teel (1998). Control allocation and zero dynamics. *Journal of Guidance, Control, and Dynamics 21*, 458–464.

Christophersen, F. J., M. Kvasnica, C. N. Jones, and M. Morari (2007). Efficient evaluation of piecewise control laws defined over a large number of polyhedra. In *Proc. of the European Control Conference*, Kos, Greece.

Dantzig, G. B., J. Folkman, and N. Z. Shapiro (1967). On the continuity of the minimum set of a continuous function. *Journal of Mathematical Analysis and Applications 17*, 519–548.

de la Peña, D. M., T. Alamo, A. Bemporad, and E. F. Camacho (2002). A dynamic programming approach for determining the explicit solution of MPC controllers. In *Proc. 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas.

Diehl, M. and J. Björnberg (2004). Robust dynamic programming for min-max model predictive control of constrained uncertain systems. *IEEE Trans. Automatic Control 49*, 2253–2257.

Dua, V., N. A. Bozinis, and E. N. Pistikopoulos (2002). A multiparametric programming approach for mixed-integer quadratic engineering problems. *Computers & Chemical Engineering 26*, 715–733.

Dua, V. and E. N. Pistikopoulos (2000). An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of operations research 99*, 123–139.

Durham, W. C. (1993). Constrained control allocation. *J. Guidance, Control and Dynamics 16*, 717–725.

Durham, W. C. (1994a). Attainable moments for the constrained control allocation problem. *J. Guidance, Control and Dynamics 17*, 1371–1373.

Durham, W. C. (1994b). Constrained control allocation: Three-moment problem. *J. Guidance, Control and Dynamics 17*, 330–336.

Eberhardt, R. and D. Ward (1999). Indirect adaptive flight control of a tailless fighter aircraft. In *Proc. AIAA Guidance, Navigation, and Control Conference*, Reston, VA, pp. 466–476.

Enns, D. (1998). Control allocation approaches. In *Proc. AIAA Guidance, Navigation and Control Conference and Exhibit*, Boston, MA, pp. 98–108.

Fiacco, A. V. (1983). *Introduction to sensitivity and stability analysis in nonlinear programming*. Orlando, FL: Academic Press Inc.

Filippi, C. (2004). An algorithm for approximate multi–parametric linear programming. *Journal of Optimization Theory and Applications 120*, 73–95.

Fossen, T. I. (2002). *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim: Marine Cybernetics.

Gal, T. (1980). A "historiogramme" of parametric programming. *Journal of the Operations Research Society 31*, 449–451.

Gal, T. (1995). *Postoptimal Analyses, Parametric Programming, and Related Topics* (Second ed.). Berlin: de Gruyter.

Gal, T. (1997). A historical sketch. In T. Gal and H. Greenberg (Eds.), *Advances in Sensitivity Analysis and Parametric Programming*, Chapter 1, pp. 1.1–1.9. Boston, MA: Kluwer Academic Publishers.

Gal, T. and H. J. Greenberg (Eds.) (1997). *Advances in Sensitivity Analysis and Parametric Programming*. Boston, MA: Kluwer Academic Publishers.

Gal, T. and J. Nedoma (1972). Multiparametric linear programming. *Management Science 18*, 406–442.

Geoffrion, A. M. and G. W. Graves (1977). Parametric and postoptimality analysis in integer linear programming. *Management Science 23*, 453–466.

Goebel, R. and A. R. Teel (2006). Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. *Automatica 42*, 573–587.

Grancharova, A. and T. A. Johansen (2005). Explicit min-max model predictive control of constrained nonlinear systems with model uncertainty. In *Porc. 16th IFAC World Congress*, Prague, Czech Republic.

Grieder, P., F. Borrelli, F. Torrisi, and M. Morari (2004). Computation of the constrained infinite time linear quadratic regulator. *Automatica 40*, 701–708.

Grieder, P., M. Kvasnica, M. Baotić, and M. Morari (2005). Stabilizing low complexity feedback control of constrained piecewise affine systems. *Automatica 41*, 1683–1694.

Hausdorff, F. (1957). *Set Theory*. New York: Chelsea Publishing Company.

Heemels, W. P. M. H., B. D. Shutter, and A. Bemporad (2001). Equivalence of hybrid dynamical models. *Automatica 37*, 1085–1091.

Hogan, W. W. (1973a). The continuity of the perturbation function of a convex program. *Operations Res. 21*, 351–352.

Hogan, W. W. (1973b). Point-to-set maps in mathematical programming. *SIAM Review 15*, 591–603.

IMO (1994, May). IMO MSC circular resolution 645: Guidelines for vessels with dynamic positioning systems. IMO Resolution 645.

International Marine Contractors Association (2000). IMCA M140 specification for DP capability plots.

Johansen, T. A. (2002). On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In *Proc. 41st IEEE Conf. on Decision and Control*, Las Vegas, NV, pp. 2768–2773.

Johansen, T. A. (2004a). Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica 40*, 293–300.

Johansen, T. A. (2004b). Optimizing nonlinear control allocation. In *Proc. 43rd IEEE Conf. on Decision and Control*, Bahamas, pp. 3435–3440.

Johansen, T. A., T. I. Fossen, and S. P. Berge (2003). Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. *IEEE Trans. Control Systems Technology 12*, 211–216.

Johansen, T. A., T. I. Fossen, and P. Tøndel (2005). Efficient optimal constrained control allocation via multi-parametric programming. *AIAA J. Guidance, Control and Dynamics 28*, 506–515.

Johansen, T. A., T. P. Fuglseth, P. Tøndel, and T. I. Fossen (2003). Optimal constrained control allocation in marine surface vessels with rudders. In *Proc. 6th IFAC Conference on Manoeuvring and Control of Marine Crafts*, pp. 215–220.

Johansen, T. A., T. P. Fuglseth, P. Tøndel, and T. I. Fossen (2007). Optimal constrained control allocation in marine surface vessels with rudders. *Control Engineering Practice 15*, 457–464.

Johansen, T. A. and A. Grancharova (2002). Approximate explicit model predictive control implemented via orthogonal search tree partitioning. In *Proc. XV IFAC World Congress*, Barcelona.

Jones, C., E. Kerrigan, and J. M. Maciejowski (2008). On Polyhedral Projection and Parametric Programming. *Journal of Optimization Theory and Applications 137*.

Jones, C. and J. M. Maciejowski (2006). Reverse Search for Parametric Linear Programming. In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, pp. 1504–1509.

Jones, C. N. (2005). *Polyhedral Tools for Control*. Ph. D. thesis, Cambridge University, Cambridge, UK.

Jones, C. N., M. Barić, and M. Morari (2007). Multiparametric linear programming with applications to control. *European Journal of Control 13*, 152–170. Special Issue.

Jones, C. N., P. Grieder, and S. V. Raković (2006). A logarithmic-time solution to the point location problem for parametric linear programming. *Automatica 42*, 2215–2218.

Jones, C. N., E. C. Kerrigan, and J. M. Maciejowski (2004). Equality set projection: A new algorithm for the projection of ploytopes in halfspace representation. Technical Report CUED/F-INFENG/TR.463, Deparment of Engineering,University of Cambridge, Cambridge, UK.

Jones, C. N., E. C. Kerrigan, and J. M. Maciejowski (2007). Lexicographic perturbation for multiparametric linear programming with applications to control. *Automatica 43*, 1808–1816.

Jones, C. N. and M. Morari (2006). Multiparametric linear complementarity problems. In *Proc. IEEE Conference on Decision and Control*, Seville, Spain, pp. 5687–5692.

Kerrigan, E. C. and D. Q. Mayne (2002). Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proc. 41st IEEE Conf. on Decision and Control*, Las Vegas, NV, pp. 1552–1557.

Kvasnica, M., P. Grieder, and M. Baotić (2005). Multi-Parametric Toolbox (MPT). http://control.ee.ethz.ch/~mpt/.

Lincoln, B. and A. Rantzer (2006). Relaxing dynamic programming. *IEEE Transactions on Automatic Control 51*, 1249–1260.

Lindegaard, K.-P. and T. I. Fossen (2003). Fuel efficient control allocation for surface vessels with active rudder usage: Experiments with a model ship. *IEEE Trans. Control Systems Technology 11*, 850–862.

Lindfors, I. (1993). Thrust allocation methods for the dynamic positioning system. In *Proc. 10th International Ship Control Symposium*, Ottawa, pp. 3.93–3.106.

Luc, D. T. and P. H. Dien (1997). Differentiable selection of optimal solutions in parametric linear programming. In *Proc. of the American Mathematical Society*, Volume 125, pp. 883–892.

Luo, Y. and D. B. Doman (2004). Model predictive dynamic control allocation with actuator dynamics. In *Proc. American Control Conference*, Boston, pp. 1695–1700.

Luo, Y., A. Serrani, S. Yurkovich, M. W. Oppenheimer, and D. B. Doman (2007). Model-predictive dynamic control allocation scheme for reentry vehicles. *Journal of Guidance, Control and Dynamics 30*, 100–113.

Mangasarian, O. L. (1979). Uniqueness of solution in linear programming. *Linear Algebra and its Applications 25*, 151–162.

Mayne, D. Q., S. V. Rakovć, R. B. Vinter, and E. C. Kerrigan (2006). Characterization of the solution to a constrained H_∞ optimal control problem. *Automatica 42*, 371–382.

Mayne, D. Q. and S. V. Raković (2002). Optimal control of constrained piecewise affine systems using reverse transformation. In *Proc. 41st IEEE Conference on Decision and Control*, Las Vegas, USA, pp. 1546–1551.

Mayne, D. Q. and S. V. Raković (2003). Optimal control of constrained piecewise affine discrete-time systems. *Computational Optimization and Applications 25*, 167–191.

Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert (2000). Constrained model predictive control: Stability and optimality. *Automatica 36*, 789–814.

Michaels, E. (1956). Continuous selections. *Annals of Mathematics 63*, 361–382.

Nocedal, J. and S. J. Wright (1999). *Numerical Optimization*. New York, USA: Springer.

Page, A. B. and M. L. Steinberg (2000). A closed-loop comparison of control allocation methods. In *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, pp. 1760–1770.

Phu, H. X. and N. D. Yen (2001). On the stability of solutions to quadratic programming problems. *Math. Program. 89*, 385–394.

Raković, S. V., P. Grieder, and C. N. Jones (2004). Computation of Voronoi Diagrams and Delaunay Triangulation via Parametric Linear Programming. Technical Report AUT04-10, Automatic Control Laboratory, Swiss Federal Institute of Technology, ETH-Zentrum, CD8092, Switzerland.

Raković, S. V., E. C. Kerrigan, and D. Q. Mayne (2004). Optimal control of constrained piecewise affine systems with state- and input-dependent disturbances. In *Proc. 16th International Symposium on Mathematical Theory of Networks and Systems*, Katholieke Universiteit Leuven, Belgium.

Raković, S. V., E. C. Kerrigan, D. Q. Mayne, and J. Lygeros (2006). Reachability analysis of discrete-time systems with disturbances. *IEEE Transactions on Automatic Control 51*, 546–561.

Rantzer, A. (2006). Relaxed dynamic programming in switching systems. *Proc. IEE Control Theory and Applications 153*, 567–574.

Rao, C. V., J. B. Rawlings, and J. H. Lee (2001). Constrained linear estimation – a moving horizon approach. *Automatica 37*, 1619–1628.

Rockafellar, R. T. (1972). *Convex Analysis*. Princeton, New Jersey: Princeton Univeristy Press.

Rockafellar, R. T. and R. J.-B. Wets (1998). *Variational Analysis*, Volume 317 of *Grundlehren der mathematischen Wissenschaften*. Berlin: Springer-Verlag.

Rostalski, P., T. Besselmann, M. Barić, and M. Morari (2007). A Hybrid Approach to Modelling, Control and State estimation of Mechanical Systems with Backlash. *International Journal of Control 80*, 1729–1740.

Schechter, M. (1987). Polyhedral functions and multiparametric linear programming. *Journal of Optimization Theory and Applications 53*, 269–280.

Seron, M. M., G. C. Goodwin, and J. A. D. Doná (2003). Characterisation of receding horizon control for constrained linear systems. *Asian Journal of Control 5*, 271–286.

Sørdalen, O. J. (1997). Optimal thrust allocation for marine vessels. *Control Engineering Practice 5*, 1223–1231.

Spjøtvold, J. (2005). Facet-to-facet violation for strictly convex parametric quadratic programs: An example. Technical Report 2005-4-W, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.

Spjøtvold, J. and T. A. Johansen (2007). Constrained control allocation for a thruster–controlled floating platform. *IEEE Trans. Control Systems Technology*. Submitted, July 2007.

Spjøtvold, J., E. C. Kerrigan, C. N. Jones, T. A. Johansen, and P. Tøndel (2004). Conjectures on an algorithm for convex parametric quadratic programs. Technical Report CUED/F-INFENG/TR.496, Department of Engineering, University of Cambridge, Cambridge, UK.

Spjøtvold, J., E. C. Kerrigan, C. N. Jones, P. Tøndel, and T. A. Johansen (2006a). On the facet-to-facet property for convex parametric quadratic programs. *Automatica 42*, 2209–2214.

Spjøtvold, J., E. C. Kerrigan, C. N. Jones, P. Tøndel, and T. A. Johansen (2006b). On the facet-to-facet property for convex parametric quadratic programs and a new exploration strategy. In *Proc. 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan.

Spjøtvold, J., E. C. Kerrigan, S. V. Raković, T. A. Johansen, and D. Q. Mayne (2007a). Inf-sup control of discontinuous piecewise affine systems. *International Journal on Nonlinear and Robust Control*. Submitted, May 2007.

Spjøtvold, J., E. C. Kerrigan, S. V. Raković, T. A. Johansen, and D. Q. Mayne (2007b). Inf-sup control of discontinuous piecewise affine systems. In *European Control Conference*, Kos, Greece.

Spjøtvold, J., S. V. Raković, P. Tøndel, and T. A. Johansen (2006). Utilizing reachability analysis in point location problems. In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, USA.

Spjøtvold, J., P. Tøndel, and T. A. Johansen (2005a). A method for obtaining continuous solutions to multiparametric linear programs. In *Proc. IFAC World Congress on Automatic Control*, Prague.

Spjøtvold, J., P. Tøndel, and T. A. Johansen (2005b). Unique polyhedral representations of continuous selections for convex multiparametric quadratic programs. In *Proc. American Contr. Conf.*, Portland.

Spjøtvold, J., P. Tøndel, and T. A. Johansen (2006). Decomposing linear control allocation problems. In *Proc. 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan.

Spjøtvold, J., P. Tøndel, and T. A. Johansen (2007). A continuous selection and unique polyhedral representation of solutions to convex parametric quadratic programs. *Journal of Optimization Theory and Applications 133*, 177–189.

Tøndel, P., T. A. Johansen, and A. Bemporad (2003a). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica 39*, 489–497.

Tøndel, P., T. A. Johansen, and A. Bemporad (2003b). Computation of piecewise affine control via binary search tree. *Automatica 39*, 945–950.

Tøndel, P., T. A. Johansen, and A. Bemporad (2003c). Further results on multi-parametric quadratic programming. In *Proc. 42nd IEEE Conf. on Decision and Control*, Hawaii, pp. 3173–3178.

Tyssø, J. and A. H. Aga (2006). DP Control system design for cyberrig I. Master's thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.

van der Panne, C. (1975). A node method for multiparametric linear programming. *Management Science 21*, 1014–1020.

Virnig, J. and D. Bodden (1994). Multivariable control allocation and control law conditioning when control effectors limit. In *Proc. AIAA Guidance, Navigation, and Control Conference*, Washington, DC, pp. 572–582.

Witsenhausen, H. (1968). A minimax control problem for sampled linear systems. *IEEE Transactions on Automatic Control 13*, 5–21.

Yu, P. L. and M. Zeleny (1976). Linear multiparametric programming by multicirteria simplex method. *Management Science 23*, 159–170.

Zhang, X. S. and D. G. Liu (1990). A note on the continuity of solutions of parametric linear programming. *Mathematical Programming 47*, 143–153.

Zhao, J. (1997). The lower continuity of optimal solution sets. *Journal of Mathematical Analysis and Applications 207*, 240–250.

Zhuo, X. W., J. A. D. Doná, and M. M. Seron (2005). Explicit solution to constrained linear estimation. In *Proc. IFAC World Congress on Automatic Control*, Prague.

Appendix A

Parametric Linear Programming Results

The parametric linear program (pLP) is as follows:

$$z^*(\theta) := \min_x \left\{ c^T x \mid (x, \theta) \in P \right\}, \tag{A.1a}$$

$$P := \{(x, \theta) \mid Ax \le b + S\theta\}. \tag{A.1b}$$

The dual of (A.1) can be written as (Borrelli, Bemporad, and Morari 2003)

$$v^*(\theta) := \min_\pi \left\{ (b + S\theta)^T \pi \mid (\pi, \theta) \in R \right\}, \tag{A.2a}$$

$$R := \left\{ (\pi, \theta) \,\middle|\, A^T \pi = c, \pi \le 0 \right\}. \tag{A.2b}$$

The primal feasibility, dual feasibility and the complementary slackness conditions for problems (A.1) and (A.2) are

$$Ax \le b + S\theta, \tag{A.3a}$$

$$A^T \pi = c, \quad \pi \le 0, \tag{A.3b}$$

$$(A_i x - b_i - S_i \theta)\pi_i = 0, \quad \forall i \in \{1, \ldots, q\}, \tag{A.3c}$$

respectively.

Let $\mathcal{A}$ be the optimal active set when $\theta = \theta_0$. When both the primal and dual solution to (A.1) are unique, then the value function, the optimizer function and the closure of the critical region are uniquely given by

$$z^*(\theta) = (b + S\theta)^T \pi^*(\theta_0), \tag{A.4a}$$

$$x^*_{\mathcal{A}}(\theta) = A_{\mathcal{A}}^{-1} S_{\mathcal{A}} \theta + A_{\mathcal{A}}^{-1} b_{\mathcal{A}}, \tag{A.4b}$$

$$R_{\mathcal{A}} = \{\theta \in \Theta \mid A_{\mathcal{N}} x^*_{\mathcal{A}}(\theta) \le b_{\mathcal{N}} + S_{\mathcal{N}} \theta\}, \tag{A.4c}$$

respectively, where $\pi^*(\theta_0)$ is the optimal dual solution and $\mathcal{N}$ is the complement of $\mathcal{A}$.

When only the dual solution is non-unique, the optimizer function and critical region are uniquely defined and found by applying Gauss reduction to the system of equalities, $A_{\mathcal{A}}x = b_{\mathcal{A}} + S_{\mathcal{A}}\theta$.

When the primal solution is non-unique a selection must be made, for instance by utilizing the minimum norm method presented in Chapter 2. To use the minimum norm method the optimal active set must be identified, which can be found either by using an interior point solver or utilizing the following lemma:

Lemma A.1 *When the solution $\pi^*(\theta)$ to (A.2) is unique for a given θ, then $\mathcal{A}^*(\theta)$ is uniquely given by*

$$\mathcal{A}^*(\theta) = \{i \in \{1, \dots, q\} \mid \pi_i^*(\theta) < 0\}. \tag{A.5}$$

PROOF: *Let x be an optimal solution to (A.1). Define the sets*

$$\mathcal{K} = \{i \in \{1, \dots, q\} \mid i \in \mathcal{A}(x, \theta), \pi_i^* < 0\},$$

and

$$\mathcal{J} = \{i \in \{1, \dots, q\} \mid i \in \mathcal{A}(x, \theta), \pi_i^* = 0\}.$$

It is obvious that $i \in \mathcal{K} \Rightarrow i \in \mathcal{A}^*$ since the complementarity condition holds for all optimal x. From (Mangasarian 1979) we have that π^* is unique if and only if LICQ holds for $A_{\mathcal{K}}$ and there is at least one feasible solution d to the system

$$A_{\mathcal{K}}d = 0, A_{\mathcal{J}}d < 0. \tag{A.6}$$

Assume first that the primal solution is non-unique. The set of feasible directions at x is given by $\{r \mid A_{\mathcal{A}(x,\theta)}r \leq 0\}$ and consequently $\bar{x} = x + \alpha d$ is feasible for sufficiently small scalar $\alpha > 0$. It is clear that for $\alpha > 0$ the constraints in $\mathcal{J}$ are inactive, so it suffices to show that $\bar{x}$ is optimal:

$$c^T(x + \alpha d) = \pi^{*T}A(x + \alpha d) = \pi^{*T}Ax = c^T x, \tag{A.7}$$

where we have used (A.3b), (A.3c), and (A.6). This implies $i \in \mathcal{J} \Rightarrow i \notin \mathcal{A}^*$. If the primal solution is unique, then we have from (Mangasarian 1979) that

$$A_{\mathcal{K}}d = 0, A_{\mathcal{J}}d \leq 0, \tag{A.8}$$

has no solution $d \neq 0$, hence, the solution to (A.6) is $d = 0$, and consequently $\mathcal{J} = \emptyset$.

Appendix B

Facet-to-Facet Violation Example

B.1 Violation of the facet-to-facet property

We show by an example that the facet-to-facet property does not generally hold for strictly convex pQPs. To show that the violation of the facet-to-facet property is not a consequence of numerical inaccuracies, the solution is analytically verified. Notation used in this appendix is defined in Chapter 3, see also references therein.

Consider the following problem:

$$V^*(\theta) := \min_{x \in \mathbb{R}^3} \frac{1}{2} x^T x \quad \text{s.t.} \quad x \in \mathcal{P}(\theta), \quad \forall \theta \in \Theta, \tag{B.1a}$$

$$\mathcal{P}(\theta) := \left\{ x \in \mathbb{R}^3 \;\middle|\; \begin{array}{ccc} x_1 - x_3 & \leq & -1 + \theta_1 \\ -x_1 - x_3 & \leq & -1 - \theta_1 \\ x_2 - x_3 & \leq & -1 - \theta_2 \\ -x_2 - x_3 & \leq & -1 + \theta_2 \\ \frac{3}{4}x_1 + \frac{16}{25}x_2 - x_3 & \leq & -1 + \theta_1 \\ -\frac{3}{4}x_1 - \frac{16}{25}x_2 - x_3 & \leq & -1 - \theta_1 \end{array} \right\}, \tag{B.1b}$$

$$\Theta := \left\{ \theta \in \mathbb{R}^2 \;\middle|\; -\tfrac{3}{2} \leq \quad \theta_i \quad \leq \tfrac{3}{2}, i = 1, 2 \right\}. \tag{B.1c}$$

The closures of the full-dimensional critical regions obtained by an implementation of the algorithm in (Tøndel, Johansen, and Bemporad 2003c) are depicted in Figure B.1. It is clear that the facet-to-facet property is violated in this example if this is the correct solution, so the remainder of this report is devoted to verify the solution.

To rule out the possibility that numerical inaccuracy is the reason for the violation of the facet-to-facet property, analytical expressions for the closures of some of the relevant critical regions will be derived. We will consider R_2, R_4 and R_7:

- R_2: Solving (B.1) for $\theta = [\frac{6}{5}, -\frac{1}{10}]^T$ we get the optimal active set $\mathcal{A} = \{2, 3, 6\}$.
- R_4: Solving (B.1) for $\theta = [\frac{1}{5}, -\frac{1}{2}]^T$ we get the optimal active set $\mathcal{A} = \{2, 4, 5\}$.

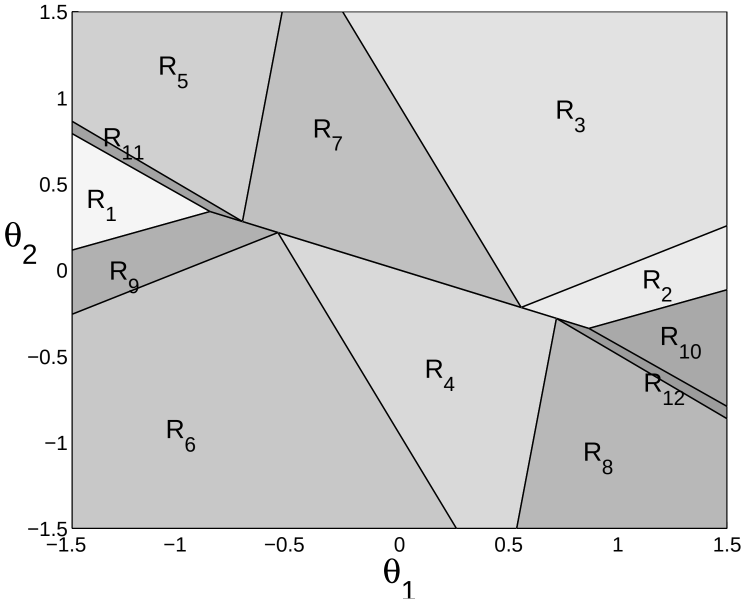


Figure B.1: Facet-to-facet property violated.

- R_7: Solving (B.1) for $\theta = [-\frac{1}{5}, \frac{1}{2}]^T$ we get the optimal active set $\mathcal{A} = \{1, 3, 6\}$.

We rewrite the constraints as:

$$\mathcal{P}(\theta) = \left\{ x \in \mathbb{R}^3 \;\middle|\; \begin{array}{rcl} x_1 - x_3 & \leq & -1 + \theta_1 \\ -x_1 - x_3 & \leq & -1 - \theta_1 \\ x_2 - x_3 & \leq & -1 - \theta_2 \\ -x_2 - x_3 & \leq & -1 + \theta_2 \\ ax_1 + bx_2 - x_3 & \leq & -1 + \theta_1 \\ -ax_1 - bx_2 - x_3 & \leq & -1 - \theta_1 \end{array} \right\}.$$

The computations are given below:

R_2: For the optimal active set $\mathcal{A} = \{2, 3, 6\}$ we have:

$$\begin{aligned} \lambda^*_{\mathcal{A}}(\theta) =& \frac{1}{(b-a+1)^2} \begin{bmatrix} -2a + b + 2a^2 + b^2 & a - b - ab - a^2 - b^2 \\ 2a + b - ab - a^2 - b^2 - 1 & -2a + a^2 + 2b^2 + 1 \\ -2a - b + 2 & a + 2b - 1 \end{bmatrix} \theta \\ &+ \frac{1}{b-a+1} \begin{bmatrix} -a \\ b \\ 1 \end{bmatrix}, \\ x^{\mathcal{A}}(\theta) =& \frac{1}{a-b-1} \begin{bmatrix} -b & b \\ a-1 & -(a-1) \\ a-1 & -b \end{bmatrix} \theta + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned}$$

The equations that ensure feasibility of $x^{\mathcal{A}}(\theta)$ are given by:

$$A_{\mathcal{N}} x^{\mathcal{A}}(\theta) \leq \hat{b}_{\mathcal{N}} + S_{\mathcal{N}}\theta \Rightarrow \frac{2}{a-b-1} \begin{bmatrix} -(a-1) & b \\ -(a-1) & b \\ -(a-1) & b \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

R_4: For the optimal active set $\mathcal{A} = \{2, 4, 5\}$ we have:

$$\begin{aligned} \lambda^*_{\mathcal{A}}(\theta) =& \frac{1}{(a+b+1)^2} \begin{bmatrix} 3b-2a+2a^2+b^2+2 & a+b-ab+a^2+b^2 \\ ab-3b-a^2-b^2+1 & -2a-a^2-2b^2-1 \\ 2a-b-4 & a-2b+1 \end{bmatrix} \theta \\ &+ \frac{1}{a+b+1} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}, \\ x^{\mathcal{A}}(\theta) =& \frac{1}{a+b+1} \begin{bmatrix} b+2 & b \\ 1-a & -a-1 \\ a-1 & -b \end{bmatrix} \theta + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned}$$

The equations that ensure feasibility of $x^{\mathcal{A}}(\theta)$ are given by:

$$A_{\mathcal{N}} x^{\mathcal{A}}(\theta) \leq \hat{b}_{\mathcal{N}} + S_{\mathcal{N}}\theta \Rightarrow \frac{2}{a+b+1} \begin{bmatrix} (1-a) & b \\ (1-a) & b \\ (1-a) & b \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

R_7: For the optimal active set $\mathcal{A} = \{1, 3, 6\}$ we have:

$$\begin{aligned} \lambda^*_{\mathcal{A}}(\theta) =& \frac{1}{(a+b+1)^2} \begin{bmatrix} 2a-3b-2a^2-b^2-2 & ab-a-b-a^2-b^2 \\ 3b-ab+a^2+b^2-1 & 2a+a^2+2b^2+1 \\ b-2a+4 & 2b-a-1 \end{bmatrix} \theta \\ &+ \frac{1}{a+b+1} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}, \\ x^{\mathcal{A}}(\theta) =& \frac{1}{a+b+1} \begin{bmatrix} b+2 & b \\ 1-a & (-a-1) \\ 1-a & b \end{bmatrix} \theta + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned}$$

The equations that ensure feasibility of $x^{\mathcal{A}}(\theta)$ are given by:

$$A_{\mathcal{N}} x^{\mathcal{A}}(\theta) \leq \hat{b}_{\mathcal{N}} + S_{\mathcal{N}}\theta \Rightarrow \frac{2}{a+b+1} \begin{bmatrix} a-1 & -b \\ a-1 & -b \\ a-1 & -b \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

If we examine the equations that ensure feasibility, we se that for R_2 and R_7 they can be written as $(a-1)\theta_1 - b\theta_2 \leq 0$, where we have assumed that $a+b+1 > 0$. For R_4 we se that they can be written as $(1-a)\theta_1 + b\theta_2 \leq 0$. It is clear that

R_2 and R_7 are in the opposite half-space of R_4, because $(a-1)\theta_1 - b\theta_2 \leq 0 \Leftrightarrow (1-a)\theta_1 + b\theta_2 \geq 0$.

$$R_2 = \left\{ \theta \in \Theta \;\middle|\; \begin{array}{ccccc} (a-1)\theta_1 & -b\theta_2 & \leq & 0 \\ (2a-b-2a^2-b^2)\theta_1 & (b-a+ab+a^2+b^2)\theta_2 & \leq & a^2-ab-a \\ (a^2+b^2-2a-b+ab+1)\theta_1 & (2a-a^2-2b^2-1)\theta_2 & \leq & b^2-ab+b \\ (2a+b-2)\theta_1 & (1-a-2b)\theta_2 & \leq & b-a+1 \end{array} \right\}$$

$$R_4 = \left\{ \theta \in \Theta \;\middle|\; \begin{array}{ccccc} (1-a)\theta_1 & b\theta_2 & \leq & 0 \\ (2a-3b-2a^2-b^2-2)\theta_1 & (ab-a-b-a^2-b^2)\theta_2 & \leq & a^2+ab+a \\ (3b-ab+a^2+b^2-1)\theta_1 & (2a+a^2+2b^2+1)\theta_2 & \leq & b^2+ab+b \\ (b-2a+4)\theta_1 & (2b-a-1)\theta_2 & \leq & a+b+1 \end{array} \right\}$$

$$R_7 = \left\{ \theta \in \Theta \;\middle|\; \begin{array}{ccccc} (a-1)\theta_1 & -b\theta_2 & \leq & 0 \\ (3b-2a+2a^2+b^2+2)\theta_1 & (a+b-ab+a^2+b^2)\theta_2 & \leq & a^2+ab+a \\ (ab-3b-a^2-b^2+1)\theta_1 & (-a^2-2b^2-2a-1)\theta_2 & \leq & b^2+ab+b \\ (2a-b-4)\theta_1 & (a-2b+1)\theta_2 & \leq & a+b+1 \end{array} \right\}$$

By letting $a = 3/4$ and $b = 16/25$ we have:
R_2:

$$R_2 = \left\{ \theta \in \Theta \;\middle|\; \begin{bmatrix} -50 & -128 \\ -\frac{3373}{5000} & \frac{13421}{10000} \\ \frac{3121}{10000} & -\frac{8817}{10000} \\ \frac{7}{50} & -\frac{103}{100} \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ -\frac{267}{400} \\ \frac{356}{625} \\ \frac{89}{100} \end{bmatrix} \right\}$$

R_4:

$$R_4 = \left\{ \theta \in \Theta \;\middle|\; \begin{bmatrix} 50 & 128 \\ -\frac{19773}{5000} & -\frac{18821}{10000} \\ \frac{14121}{10000} & \frac{38817}{10000} \\ \frac{157}{50} & -\frac{47}{100} \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ \frac{717}{400} \\ \frac{956}{625} \\ \frac{239}{100} \end{bmatrix} \right\}$$

R_7:

$$R_7 = \left\{ \theta \in \Theta \;\middle|\; \begin{bmatrix} -50 & -128 \\ \frac{19773}{5000} & \frac{18821}{10000} \\ -\frac{14121}{10000} & -\frac{38817}{10000} \\ -\frac{157}{50} & \frac{47}{100} \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ \frac{717}{400} \\ \frac{956}{625} \\ \frac{239}{100} \end{bmatrix} \right\}$$

Figure B.2 depicts the closures of the critical regions from the exact computation and we see that the facet-to-facet property is still violated. To illustrate that we are not missing any small critical regions, we delete the equation that ensures feasibility from the representations of the three closures, see Figure B.3.

To verify that the correct optimal active set has been chosen, we check that $(x^{\mathcal{A}}(\theta), \lambda^*_{\mathcal{A}}(\theta))$ is the unique KKT point for the given θ.

R_2: Let $\theta = [\frac{6}{5} \;\; -\frac{1}{10}]^T$ and consider the feasibility condition:

$$Ax^{\mathcal{A}}(\theta) - \hat{b} - S\theta = \frac{-2}{a-b-1} \begin{bmatrix} -\theta_1 + a\theta_1 - b\theta_2 \\ 0 \\ 0 \\ -\theta_1 + a\theta_1 - b\theta_2 \\ -\theta_1 + a\theta_1 - b\theta_2 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

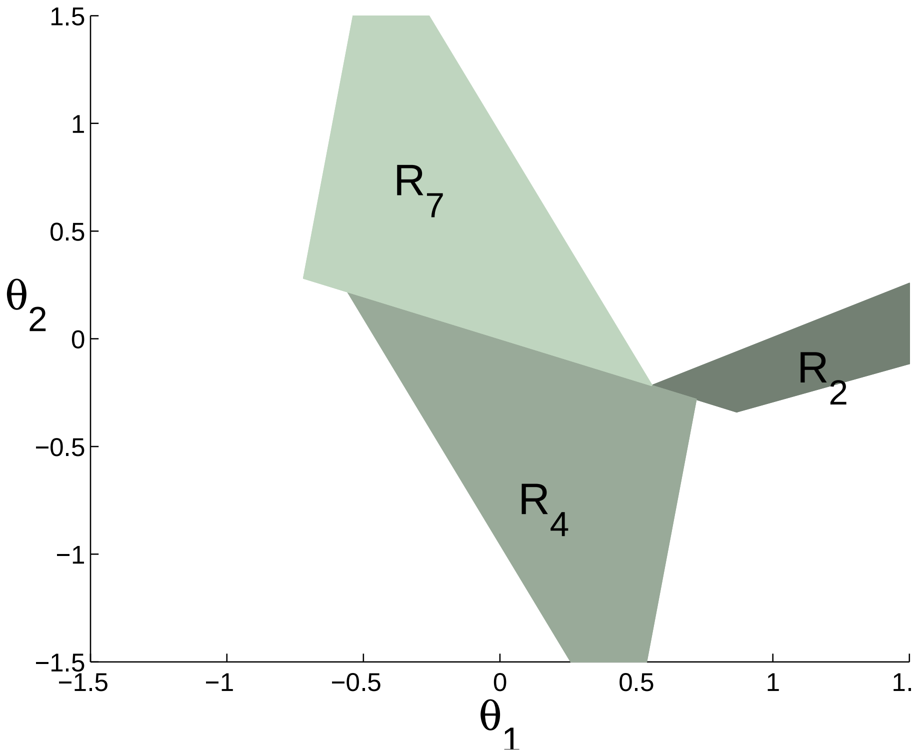


Figure B.2: Closures of critical regions, exact computation.

Assuming that $a - b - 1 < 0$ yields:

$$\begin{bmatrix} -\theta_1 + a\theta_1 - b\theta_2 \\ 0 \\ 0 \\ -\theta_1 + a\theta_1 - b\theta_2 \\ -\theta_1 + a\theta_1 - b\theta_2 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} -0.236 \\ 0 \\ 0 \\ -0.236 \\ -0.236 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

So the feasibility condition holds. Since $\lambda_i^* = 0$ for $i \notin \mathcal{A}$, the complementarity condition also holds. Finally, $\lambda_{\mathcal{A}}^*(\theta) \geq 0$ is verified:

$$\frac{1}{(b-a+1)^2} \begin{bmatrix} -2a+b+2a+b & a-b-ab-a-b \\ 2a+b-ab-a-b-1 & -2a+a+2b+1 \\ -2a-b+2 & a+2b-1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \frac{1}{b-a+1} \begin{bmatrix} -a \\ b \\ 1 \end{bmatrix} \geq 0$$

$$\Rightarrow \begin{bmatrix} -(a+ab+2a\theta_1 - a\theta_2 - b\theta_1 + b\theta_2 + ab\theta_2 - a^2 - 2a^2\theta_1 + a^2\theta_2 - b^2\theta_1 + b^2\theta_2) \\ (b - ab - \theta_1 + \theta_2 + 2a\theta_1 - 2a\theta_2 + b\theta_1 - ab\theta_1 + b^2 - a^2\theta_1 + a^2\theta_2 - b^2\theta_1 + 2b^2\theta_2) \\ (b - a + 2\theta_1 - \theta_2 - 2a\theta_1 + a\theta_2 - b\theta_1 + 2b\theta_2 + 1) \end{bmatrix} \geq 0$$

$$\Rightarrow \begin{bmatrix} 0.27623 \\ 0.10691 \\ 0.619 \end{bmatrix} > \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Since $x^*(\theta)$ is unique, the optimal active set is determined uniquely, and since LICQ holds for $\mathcal{A}$, the KKT-point $(x^{\mathcal{A}}(\theta), \lambda_{\mathcal{A}}^*(\theta))$ is unique.

Table B.1: Optimal active sets and analytically computed non-minimal critical regions. The first $|\mathcal{A}|$ inequalities correspond to $\lambda^*_{\mathcal{A}}(\theta) \geq 0$, where $|\cdot|$ denotes the cardinality.

Region	$\mathcal{A}$	Polyhedron
R_1	$1, 5$	$\begin{bmatrix} \frac{2221}{8817} & 0 \\ \frac{2500}{8817} & 0 \\ \frac{9442}{8817} & 0 \\ \frac{2107}{2939} & 1 \\ \frac{3121}{8817} & -1 \\ \frac{9442}{8817} & 0 \end{bmatrix} \theta \leq \begin{bmatrix} \frac{2221}{8817} \\ \frac{2500}{8817} \\ -\frac{8192}{8817} \\ -\frac{832}{2939} \\ -\frac{5696}{8817} \\ -\frac{8192}{8817} \end{bmatrix}$
R_2	$2, 3, 6$	$\begin{bmatrix} -\frac{6746}{7921} & \frac{13421}{7921} \\ \frac{3121}{7921} & -\frac{8817}{7921} \\ \frac{1400}{7921} & -\frac{10300}{7921} \\ -\frac{50}{89} & -\frac{128}{89} \\ -\frac{50}{89} & -\frac{128}{89} \\ -\frac{50}{89} & -\frac{128}{89} \end{bmatrix} \theta \leq \begin{bmatrix} -\frac{75}{89} \\ \frac{64}{89} \\ \frac{100}{89} \\ 0 \\ 0 \\ 0 \end{bmatrix}$
R_3	$3, 6$	$\begin{bmatrix} \frac{1800}{19073} & -\frac{19721}{38146} \\ -\frac{10000}{19073} & \frac{1800}{19073} \\ -\frac{19773}{19073} & -\frac{18821}{38146} \\ \frac{3373}{19073} & -\frac{13421}{38146} \\ -\frac{16400}{19073} & -\frac{16121}{19073} \\ -\frac{16400}{19073} & -\frac{16121}{19073} \end{bmatrix} \theta \leq \begin{bmatrix} \frac{16121}{38146} \\ \frac{8200}{19073} \\ -\frac{17925}{38146} \\ \frac{6675}{38146} \\ -\frac{5625}{19073} \\ -\frac{5625}{19073} \end{bmatrix}$

Table B.2: Optimal active sets and analytically computed non-minimal critical regions. The first $|\mathcal{A}|$ inequalities correspond to $\lambda^*_{\mathcal{A}}(\theta) \geq 0$, where $|\cdot|$ denotes the cardinality.

Region	$\mathcal{A}$	Polyhedron
R_4	$2,4,5$	$\begin{bmatrix} -\frac{39546}{57121} & -\frac{18821}{57121} \\ \frac{14121}{57121} & \frac{38817}{57121} \\ \frac{31400}{57121} & -\frac{4700}{57121} \\ \frac{50}{239} & \frac{128}{239} \\ \frac{50}{239} & \frac{128}{239} \\ \frac{50}{239} & \frac{128}{239} \end{bmatrix} \theta \leq \begin{bmatrix} \frac{75}{239} \\ \frac{64}{239} \\ \frac{100}{239} \\ 0 \\ 0 \\ 0 \end{bmatrix}$
R_5	$1,3$	$\begin{bmatrix} 2/3 & 1/3 \\ -1/3 & -2/3 \\ 2/3 & -2/3 \\ 2/3 & -2/3 \\ -\frac{19}{50} & -\frac{51}{100} \\ \frac{157}{150} & -\frac{47}{300} \end{bmatrix} \theta \leq \begin{bmatrix} 1/3 \\ 1/3 \\ -2/3 \\ -2/3 \\ \frac{13}{100} \\ -\frac{239}{300} \end{bmatrix}$
R_6	$4,5$	$\begin{bmatrix} -\frac{1800}{19073} & \frac{19721}{38146} \\ \frac{10000}{19073} & -\frac{1800}{19073} \\ -\frac{3373}{19073} & \frac{13421}{38146} \\ \frac{19773}{19073} & \frac{18821}{38146} \\ \frac{16400}{19073} & \frac{16121}{19073} \\ \frac{16400}{19073} & \frac{16121}{19073} \end{bmatrix} \theta \leq \begin{bmatrix} \frac{16121}{38146} \\ \frac{8200}{19073} \\ \frac{6675}{38146} \\ -\frac{17925}{38146} \\ -\frac{5625}{19073} \\ -\frac{5625}{19073} \end{bmatrix}$

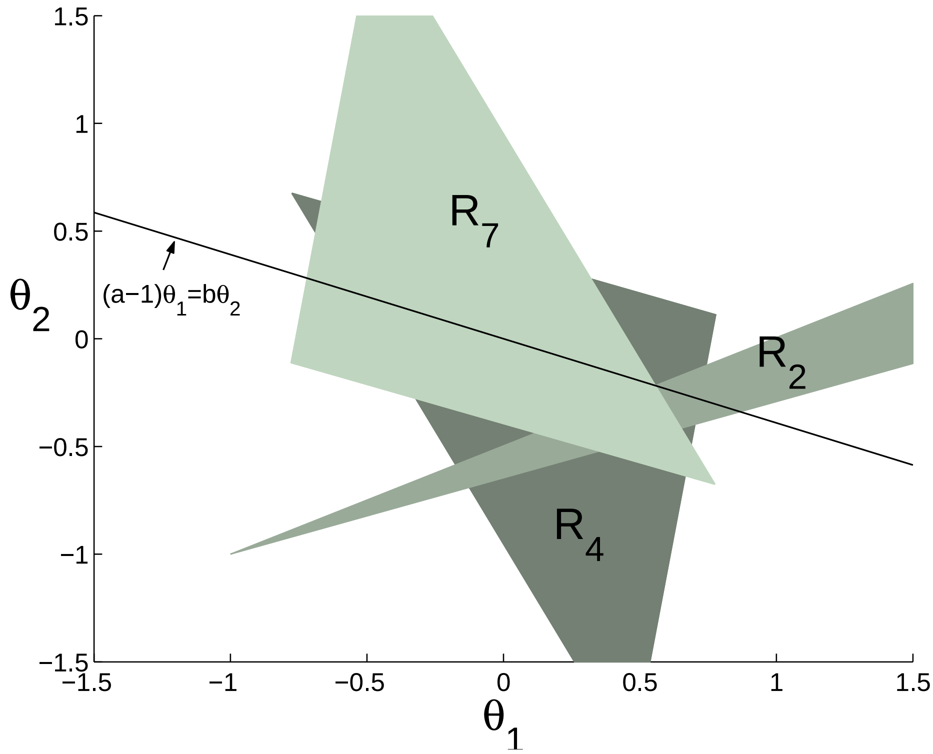


Figure B.3: Closures of critical regions without feasibility constraints.

Without showing the calculations in details, we repeat the analytical computation of the region and the verification of the KKT point for all the critical regions. We will also verify analytically that LICQ holds for all the optimal active sets. In Tables B.1- B.4 the optimal active sets for each critical region and analytically computed expressions for the closure of the critical regions are given. In Table B.5, KKT points calculated from $(x^{\mathcal{A}}(\theta), \lambda^*_{\mathcal{A}}(\theta))$ are given. In Tables B.6 and B.7 it is verified that LICQ holds for all the optimal active sets that define full-dimensional critical regions. In Table B.6 the determinants of $A_{\mathcal{A}}$ are given, hence only the optimal active sets containing three elements are considered. In Table B.7 the Gaussian reductions to "staircase form" are given, i.e. LICQ is proven to hold for the optimal active sets with two elements.

We now consider the lower dimensional critical region given by $\mathcal{A} = \{1, \ldots, 6\}$. Since LICQ is violated for $\mathcal{A}$ the normal cone will be analytically computed. The normal cone is defined as the conic hull of the column vectors of A^T. The rays of the normal cone is depicted in Figure B.4. To obtain the half-space representation of the cone, a systems of 6 equations with 3 unknowns are solved, e.g. for ray 1

Table B.3: Optimal active sets and analytically computed non-minimal critical regions. The first $|\mathcal{A}|$ inequalities correspond to $\lambda^*_{\mathcal{A}}(\theta) \geq 0$, where $|\cdot|$ denotes the cardinality.

Region	$\mathcal{A}$	Polyhedron
R_7	$1,3,6$	$\begin{bmatrix} \frac{39546}{57121} & \frac{18821}{57121} \\ -\frac{14121}{57121} & -\frac{38817}{57121} \\ -\frac{31400}{57121} & \frac{4700}{57121} \\ -\frac{50}{239} & -\frac{128}{239} \\ -\frac{50}{239} & -\frac{128}{239} \\ -\frac{50}{239} & -\frac{128}{239} \end{bmatrix} \theta \leq \begin{bmatrix} \frac{75}{239} \\ \frac{64}{239} \\ \frac{100}{239} \\ 0 \\ 0 \\ 0 \end{bmatrix}$
R_8	$2,4$	$\begin{bmatrix} -2/3 & -1/3 \\ 1/3 & 2/3 \\ -2/3 & 2/3 \\ -2/3 & 2/3 \\ -\frac{157}{150} & \frac{47}{300} \\ \frac{19}{50} & \frac{51}{100} \end{bmatrix} \theta \leq \begin{bmatrix} 1/3 \\ 1/3 \\ -2/3 \\ -2/3 \\ -\frac{239}{300} \\ \frac{13}{100} \end{bmatrix}$
R_9	$1,4,5$	$\begin{bmatrix} \frac{6746}{7921} & -\frac{13421}{7921} \\ -\frac{3121}{7921} & \frac{8817}{7921} \\ -\frac{1400}{7921} & \frac{10300}{7921} \\ \frac{50}{89} & \frac{128}{89} \\ \frac{50}{89} & \frac{128}{89} \\ \frac{50}{89} & \frac{128}{89} \end{bmatrix} \theta \leq \begin{bmatrix} -\frac{75}{89} \\ \frac{64}{89} \\ \frac{100}{89} \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Table B.4: Optimal active sets and analytically computed non-minimal critical regions. The first $|\mathcal{A}|$ inequalities correspond to $\lambda^*_{\mathcal{A}}(\theta) \geq 0$, where $|\cdot|$ denotes the cardinality.

Region	$\mathcal{A}$	Polyhedron
R_{10}	$2, 6$	$\begin{bmatrix} -\frac{2221}{8817} & 0 \\ -\frac{2500}{8817} & 0 \\ -\frac{9442}{8817} & 0 \\ -\frac{3121}{8817} & 1 \\ -\frac{2107}{2939} & -1 \\ -\frac{9442}{8817} & 0 \end{bmatrix} \theta \leq \begin{bmatrix} \frac{2221}{8817} \\ \frac{2500}{8817} \\ -\frac{8192}{8817} \\ -\frac{5696}{8817} \\ -\frac{832}{2939} \\ -\frac{8192}{8817} \end{bmatrix}$
R_{11}	$1, 3, 5$	$\begin{bmatrix} -\frac{2018}{507} & -\frac{2993}{507} \\ -\frac{2107}{507} & -\frac{2939}{507} \\ \frac{3800}{507} & \frac{1700}{169} \\ -\frac{50}{39} & -\frac{128}{39} \\ -\frac{50}{39} & -\frac{128}{39} \\ -\frac{50}{39} & -\frac{128}{39} \end{bmatrix} \theta \leq \begin{bmatrix} \frac{25}{13} \\ \frac{64}{39} \\ -\frac{100}{39} \\ 0 \\ 0 \\ 0 \end{bmatrix}$
R_{12}	$2, 4, 6$	$\begin{bmatrix} \frac{2018}{507} & \frac{2993}{507} \\ \frac{2107}{507} & \frac{2939}{507} \\ -\frac{3800}{507} & -\frac{1700}{169} \\ \frac{50}{39} & \frac{128}{39} \\ \frac{50}{39} & \frac{128}{39} \\ \frac{50}{39} & \frac{128}{39} \end{bmatrix} \theta \leq \begin{bmatrix} \frac{25}{13} \\ \frac{64}{39} \\ -\frac{100}{39} \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Table B.5: A KKT-point is analytically computed from $x^{\mathcal{A}}(\theta)$ and $\lambda^*_{\mathcal{A}}(\theta)$ for a given parameter θ in the interior of each critical region. To verify that the correct optimal active set has been chosen, one can substitute $x^{\mathcal{A}}(\theta)$ and $\lambda^*_{\mathcal{A}}(\theta)$ into the KKT conditions and note that if LICQ holds for $\mathcal{A}$, the KKT point is unique.

	R_1	R_2
θ^T	$\begin{bmatrix} -\frac{3}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{6}{5} & -\frac{1}{10} \end{bmatrix}$
$x^{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} -\frac{10240}{8817} & -\frac{4000}{8817} & \frac{23605}{17634} \end{bmatrix}$	$\begin{bmatrix} \frac{416}{445} & \frac{65}{178} & \frac{563}{445} \end{bmatrix}$
$\lambda^*_{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{11105}{17634} & \frac{6250}{8817} \end{bmatrix}$	$\begin{bmatrix} \frac{27623}{79210} & \frac{10691}{79210} & \frac{6190}{7921} \end{bmatrix}$
	R_3	R_4
θ^T	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$
$x^{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{12300}{19073} & -\frac{5625}{19073} & \frac{32521}{19073} \end{bmatrix}$	$\begin{bmatrix} \frac{100}{239} & \frac{100}{239} & \frac{517}{478} \end{bmatrix}$
$\lambda^*_{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{16121}{19073} & \frac{16400}{19073} \end{bmatrix}$	$\begin{bmatrix} \frac{56575}{114242} & \frac{27644}{57121} & \frac{5850}{57121} \end{bmatrix}$
	R_5	R_6
θ^T	$\begin{bmatrix} -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 \end{bmatrix}$
$x^{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} -\frac{2}{3} & -\frac{2}{3} & \frac{4}{3} \end{bmatrix}$	$\begin{bmatrix} -\frac{12300}{19073} & \frac{5625}{19073} & \frac{32521}{19073} \end{bmatrix}$
$\lambda^*_{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{2}{3} & \frac{2}{3} \end{bmatrix}$	$\begin{bmatrix} \frac{16121}{19073} & \frac{16400}{19073} \end{bmatrix}$
	R_7	R_8
θ^T	$\begin{bmatrix} 0 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \end{bmatrix}$
$x^{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{32}{239} & -\frac{175}{478} & \frac{271}{239} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{3} & \frac{2}{3} & \frac{4}{3} \end{bmatrix}$
$\lambda^*_{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{17029}{114242} & \frac{69409}{114242} & \frac{21550}{57121} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{3} & \frac{2}{3} \end{bmatrix}$
	R_9	R_{10}
θ^T	$\begin{bmatrix} -\frac{6}{5} & -\frac{1}{10} \end{bmatrix}$	$\begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$
$x^{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} -\frac{352}{445} & -\frac{55}{178} & \frac{627}{445} \end{bmatrix}$	$\begin{bmatrix} \frac{10240}{8817} & \frac{4000}{8817} & \frac{23605}{17634} \end{bmatrix}$
$\lambda^*_{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{781}{79210} & \frac{5665}{15842} & \frac{8250}{7921} \end{bmatrix}$	$\begin{bmatrix} \frac{11105}{17634} & \frac{6250}{8817} \end{bmatrix}$
	R_{11}	R_{12}
θ^T	$\begin{bmatrix} -1 & \frac{23}{50} \end{bmatrix}$	$\begin{bmatrix} 1 & -\frac{23}{50} \end{bmatrix}$
$x^{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} -\frac{288}{325} & -\frac{9}{26} & \frac{362}{325} \end{bmatrix}$	$\begin{bmatrix} \frac{288}{325} & \frac{9}{26} & \frac{362}{325} \end{bmatrix}$
$\lambda^*_{\mathcal{A}}(\theta)^T$	$\begin{bmatrix} \frac{5563}{8450} & \frac{3847}{25350} & \frac{154}{507} \end{bmatrix}$	$\begin{bmatrix} \frac{5563}{8450} & \frac{3847}{25350} & \frac{154}{507} \end{bmatrix}$

Table B.6: The determinants of $A_{\mathcal{A}}$ are given to show that LICQ holds for the optimal active sets with three elements.

	R_2	R_4	R_7	R_9	R_{11}	R_{12}
$\det(A_{\mathcal{A}})$	$\frac{89}{100}$	$-\frac{239}{100}$	$-\frac{239}{100}$	$\frac{89}{100}$	$\frac{39}{100}$	$\frac{39}{100}$

Table B.7: The Gaussian reductions of $A_{\mathcal{A}}$ to "staircase form" are given to show that LICQ holds for the optimal active sets with two elements.

	R_1	R_3
$\mathrm{rref}(A_{\mathcal{A}})$	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & \frac{16}{25} & -1/4 \end{bmatrix}$	$\begin{bmatrix} -3/4 & -\frac{16}{25} & -1 \\ 0 & -3/4 & 3/4 \end{bmatrix}$
	R_5	R_6
$\mathrm{rref}(A_{\mathcal{A}})$	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 3/4 & \frac{16}{25} & -1 \\ 0 & -3/4 & -3/4 \end{bmatrix}$
	R_8	R_{10}
$\mathrm{rref}(A_{\mathcal{A}})$	$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & -1 \\ 0 & \frac{16}{25} & 1/4 \end{bmatrix}$

and 4 the plane $ax_1 + bx_2 + x_3 = c$ is given by

$$\begin{aligned} -b-1 &= c, \\ a-1 &= c, \\ 0 &= c. \end{aligned}$$

Thus, we get the inequality $x_1 - x_2 + x_3 \leq 0$. The normal cone is given by:

$$\left\{ x \in \mathbb{R}^3 \;\middle|\; \begin{array}{rcl} x_1 - x_2 + x_3 & \leq & 0 \\ -x_1 + x_2 + x_3 & \leq & 0 \\ -\frac{12}{25}x_1 - x_2 + x_3 & \leq & 0 \\ x_1 + \frac{25}{64}x_2 + x_3 & \leq & 0 \\ \frac{12}{25}x_1 + x_2 + x_3 & \leq & 0 \\ -x_1 - \frac{25}{64}x_2 + x_3 & \leq & 0 \end{array} \right\} =: \left\{ x \in \mathbb{R}^3 \;\middle|\; L_I^{\mathcal{A}} x \leq 0 \right\}$$

We now compute the optimal solution $x^{\mathcal{A}}(\theta)$ by Gaussian reduction:

$$x^{\mathcal{A}}(\theta) = \begin{bmatrix} 0 & -\frac{64}{25} \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The normal cone optimality condition becomes:

$$L_I^{\mathcal{A}} x^{\mathcal{A}}(\theta) \geq 0 \Rightarrow \begin{bmatrix} 0 & -\frac{39}{25} \\ 0 & \frac{39}{25} \\ 0 & \frac{1393}{625} \\ 0 & -\frac{4721}{1600} \\ 0 & -\frac{1393}{625} \\ 0 & \frac{4721}{1600} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \geq - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow -\frac{1600}{4721} \leq \theta_2 \leq \frac{1600}{4721}.$$

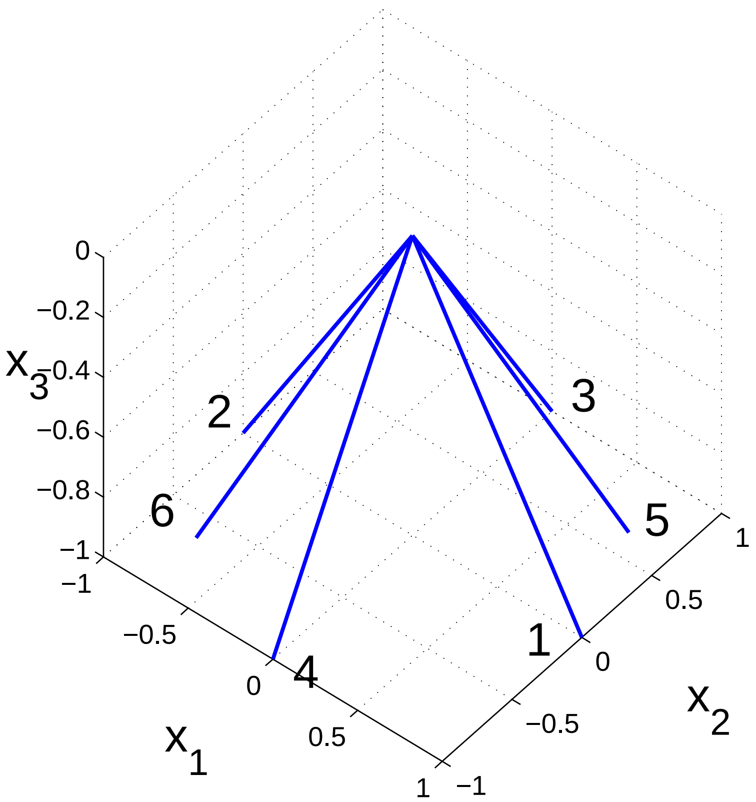


Figure B.4: The normal cone associated with $\mathcal{A} = \{1, \dots, 6\}$ is the conic hull of the rays in the figure. Each ray is normal to a constraint in $\mathcal{P}(\theta)$.

So the closure of the critical region becomes:

$$\mathrm{cl}(\Theta_{\{1,\ldots,6\}}) = \left\{\theta \in \Theta \,\middle|\, \theta_1 = -\frac{64}{25}\theta_2, -\frac{1600}{4721} \le \theta_2 \le \frac{1600}{4721}\right\}.$$

B.2 Maple files

Three Maple files supplement this appendix.

1. VerifyKKT.mv: The file verifies analytically that the KKT conditions hold for a selected parameter in the interior of each critical region.

2. ComputeRegions.mw: The files computes the closures of the critical regions analytically and verifies that LICQ holds for each optimal active set.

3. AllCombinations.mw: The file verifies numerically that every possible combination of active constraints, which has not been reported as an optimal active set in the solution, yields empty or lower dimensional critical regions. To be more specific:

   - For all combinations of constraints for which LICQ holds, the region is computed using the KKT-conditions and the ”solve” command is used to compute a minimal representation of the polyhedron.
   - For all combination of constraints for which LICQ is violated, a Gaussian reduction of $G := A_{\mathcal{A}} - S_{\mathcal{A}} - b_{\mathcal{A}}$ is computed. This shows that several combinations must yield lower-dimensional regions since we get an equality constraint on θ, i.e. $C_i\theta = d_i$.
   - For all combination of constraints for which LICQ is violated and Gaussian reduction does not yield a lower dimensional region, $x^{\mathcal{A}}(\theta)$ is computed and used in the feasibility condition. The ”solve” command is then used to compute the region of feasibility, and it turns out that this is sufficient for showing that the remaining combinations of active constraints yield lower-dimensional critical regions.

Finally, it should be noted that the active sets $\mathcal{A} = \{3\}$ and $\mathcal{A} = \{4\}$ define full-dimensional critical regions if we consider $\Theta := \mathbb{R}^s$, but that the regions do not intersect the polytope we are considering.

B.3 Convexity lemma

If S is a set, then $\mathrm{cone}(S)$ denotes the smallest cone containing S. If S is a set and s_0 is a point, then $S + s_0$ denotes the set $\{s + s_0 \mid s \in S\}$. In the below lemma we use the notation $\mathrm{conch}\{s, S\} := \mathrm{cone}(S - s) + s$.

Lemma B.1 *Given two s-dimensional closed sets, P and S, in $\mathbb{R}^s$, such that* $\mathrm{int}(P) \cap \mathrm{int}(S) = \emptyset$*. A necessary condition for the set $P \cup S$ to be convex, is that*

$$\dim(P \cap S) = s - 1.$$

PROOF: *The union of two disjoint closed sets is clearly non-convex so the intersection must be non-empty. Denote the line segment from a point $s \in S$ to a point $p \in P$ by (s, p). For the union to be convex, we must have $(s, p) \in P \cup S$, $\forall s \in S$ and $\forall p \in P$. Consider a point $s_0 \in \mathrm{int}(S)$ and let $C := \mathrm{conch}\{s_0, P\}$. Due to the s-dimensionality of P, C is clearly s-dimensional. Denote the extreme rays of C by $\{v_1, \ldots, v_n\}$. Clearly, v_i is a direction from s_0 to a point $p_i \in P$. Denote all line segments corresponding to these directions by (s_0, p_i), $i = \{1, \ldots, n\}$. By noting that* $\mathrm{int}(P) \cap \mathrm{int}(S) = \emptyset$ *we conclude that if $(s_0, p_i) \cap P \cap S = \emptyset$, then we have $(s_0, p_i) \notin P \cup S$. So $(s_0, p_i) \cap P \cap S \neq \emptyset$ must hold $\forall i$ for the union to be convex, in other words, all extreme rays of C must run through the intersection. Assume now that* $\dim(P \cap S) = s - 2$*. Take the conic hull of s_0 and $P \cap S$, i.e. $C_{int} := \mathrm{conch}\{s_0, P \cap S\}$, then C_{int} contains all directions from s_0 to $P \cap S$. It is clear that since $P \cap S$ is spanned by $s - 2$ vectors, C_{int} is spanned by $s - 1$ vectors, i.e. $\dim(C_{int}) = s - 1$. Since $\dim(C) > \dim(C_{int})$, $\exists v_l \in C$ such that $v_l \notin C_{int}$, let (s_0, p_l) denote the line segment defined by v_l. It is clear that $(s_0, p_l) \cap P \cap S = \emptyset \Rightarrow (s_0, p_l) \notin P \cup S$. All arguments made are valid for* $\dim(P \cap S) < s - 2$.

$\square$

Appendix C

Binary Search Tree

We recall the procedure for evaluating a piecewise affine function presented in (Tøndel, Johansen, and Bemporad 2003b).

Consider the set of full-dimensional polyhedra $\{X_1, X_2, ..., X_{n_r}\}$ and a corresponding set of affine functions $\{F_1, F_2, ..., F_K\}$. Note that $K \leq n_r$ since several polyhedra can be associated with the same affine function. Let all unique hyperplanes defining the polyhedra $\{X_1, X_2, ..., X_{n_r}\}$ be denoted by $a_j^T x = b_j$ for $j = 1, 2, ..., L$, and define $d_j(x) = a_j^T x - b_j$. Let the index representation $\mathcal{J}$ of a polyhedron denote a combination of indexes combined with the sign of d_j, e.g. $\mathcal{J} = \{1^+, 4^+, 6^-\}$ would mean that $d_1(x) \geq 0$, $d_4(x) \geq 0$ and $d_6(x) \leq 0$. Such an index representation obviously defines a polyhedron; $\mathcal{P}(\mathcal{J})$. We can further define the set of polyhedra corresponding to $\mathcal{J}$ as the index set $\mathcal{I}(\mathcal{J}) = \{i | X_i \cap \mathcal{P}(\mathcal{J})$ is full-dimensional$\}$.

The idea is to construct a binary search tree so that for a given $x \in X$, at each node we will evaluate one affine function $d_j(x)$ and test its sign. Based on the sign we select the left or the right sub-tree. Traversing the tree from the root to a leaf node, one will end up with a leaf node giving a unique affine control law. Each node of the tree will be denoted by N_k. An unexplored non-leaf node N_k will consist of $(\mathcal{I}_k, \mathcal{J}_k)$, where $\mathcal{J}_k$ is the index set (with signs) of hyperplanes obtained by traversing the tree from the root node to N_k and $\mathcal{I}_k = \mathcal{I}(\mathcal{J}_k)$. An explored non-leaf node will contain an index j_k to a hyperplane, while a leaf node will contain an affine control law, F_k. See Figure C.1 for an example of a simple search tree.

The next algorithm is used on-line to traverse the binary tree. In general, the worst-case number of arithmetic operations required to search the tree and evaluate the PWA function is $(2n + 1)D + 2nm$, where D is the depth of the tree, m is the number of variables and n is the number of parameters. At each node there are n multiplications, n additions and 1 comparison. Moreover, $2nm$ operations are required to evaluate the affine state feedback of the leaf node.

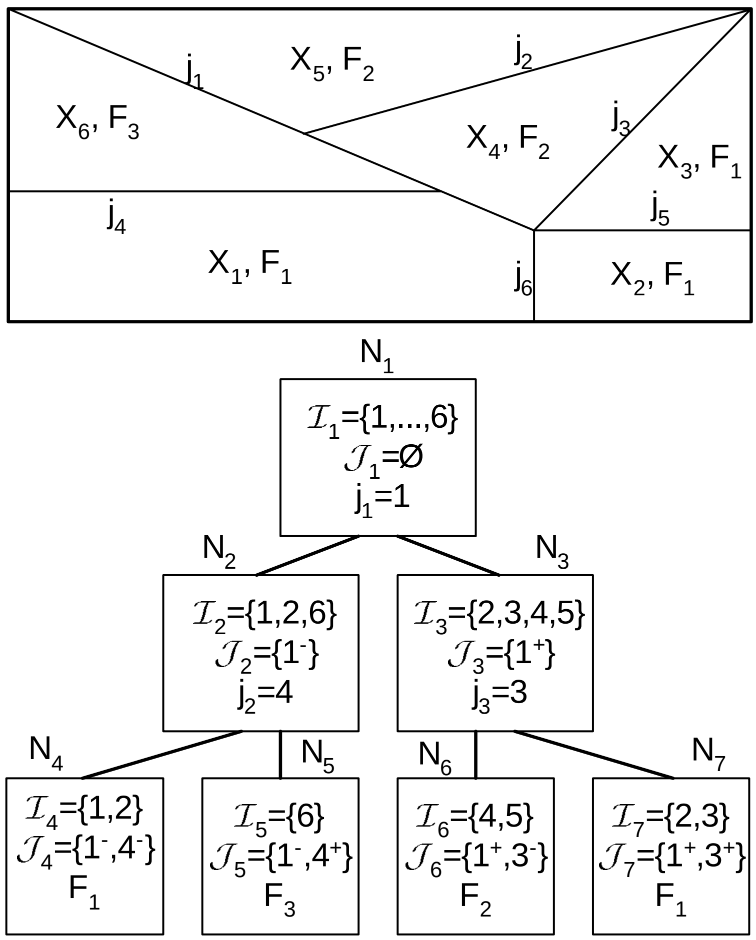


Figure C.1: Search tree generated from partition with $n_r = 6$ and $K = 3$

Procedure C.1 Evaluation of PWA functions via binary search tree

Let the current node N be the root node of the tree.
while N is not a leaf node
Evaluate the hyperplane $d(x) = a^T x - b$ corresponding to node N.
Let N be the child node according to the sign of $d(x)$.
end (while)
Evaluate the function F corresponding to leaf-node N at x.

Appendix D

CyberRig I data

D.1 Physical Properties

The relevant physical properties of CyberRig I are given in Table D.1.

Water density, ρ	1000
Propeller diameter, D	0.05 m
Thrust coefficient, K_T	0.6173
Sampling frequency	10 Hz
$l_{1,x}$	-0.305 m
$l_{1,y}$	0.365 m
$l_{2,x}$	-0.365 m
$l_{2,y}$	0.305 m
$l_{3,x}$	-0.365 m
$l_{3,y}$	-0.305 m
$l_{4,x}$	-0.305 m
$l_{4,y}$	-0.365 m
$l_{5,x}$	0.305 m
$l_{5,y}$	-0.365 m
$l_{6,x}$	0.365 m
$l_{6,y}$	-0.305 m
$l_{7,x}$	0.365 m
$l_{7,y}$	0.305 m
$l_{8,x}$	0.305 m
$l_{8,y}$	0.365 m
Upper limit on thrust magnitude, $\overline{T}_i$	2.4 N
Lower limit on thrust magnitude, $\underline{T}_i$	0 N

Table D.1: Relevant physical data for CyberRig I.

D.2 Hardware

The inboard micro PC on CyberRig I is and IEI Wafer-5822-300 PC/104 compatible board powered by a 300 MHz Pentium compatible CPU with 512 Mb RAM. The micro PC is running a QNX Neutrino Version 6.2 real-time operating system installed on a spearate 50 Gb hard drive. It controls 8 different cards connected to PC/104 bus. The command station running Windows XP is a Dell LATTITUDE D800 laptop PC with a 1.6 GHz Pentium M processor and 512 Mb RAM. All communication between the micro PC and the host PC goes through a LAN.

The azimuth thruster propeller velocity is controlled by four Mesa Electronics 4127 Motor Controller Cards. 4127 is a low cost, LM629 based 2 axis DC servo motor control system implemented on a stackable PC/104 card. The per axis output of the 4127 is an 8 bit sign-magnitude PWM signal that drives a Mesa 7127 dual H-bridge intended for motion control applications.

The rotation of the azimuth thrusters is controlled by three 5936 Stepper Motor Controller PC/104 stackable cards. The 5936 allows a PC/104 based computer system to control three independent SERVEX two phase stepper motor drivers. The orientation of the azimuth is read by a position sensor that outputs an analog voltage in the range 0-5 V to DM6210 PC/104 compatible I/O card.

The case holding the eight cards of the PC/104 stack and CPU is isolated in the middle of the rig and connected to four boxes each containing a dual H-bridge and two stepper motor drives. Each of these boxes are connected to one of the four legs, each containing two stepper motors for azimuth and two RPM controlled propeller motors.

D.3 Software

The low-level software of CyberRig I is written in C and implemented as S-functions in MATLAB/SIMULINK. The SIMULINK models are handled by Opal RT-lab which compiles all code necessary to download the model to the target QNX real-time system. The sampling frequency of the system was chosen to be 10Hz.