Francesco Borrelli

Constrained Optimal Control of Linear and Hybrid Systems

SPIN Springer's internal project number, if known

– Monograph –

February 20, 2003

Springer
Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

To my family.

Acknowledgments

This book is a revised version of my Ph.D. thesis from the automatic control laboratory at ETH-Zurich, written during the years 2001 to 2002. My thesis advisors were Manfred Morari and Alberto Bemporad. Almost all the material presented in this book is extracted from work done jointly with them. I would like to express my deepest gratitude to Alberto and Manfred for sharing with me their ideas and for always being helpful and supportive. My wholehearted gratitude also goes to Professor David Mayne for his careful reading of the book and suggestions on how to improve it. A great thanks goes to Davor Hrovat and Mike Fordor who gave me the opportunity to collaborate with the Ford Research Laboratories in Dearborn, USA.

During my Ph.D. I have been in contact with many extraordinary researchers. Among them I'd like to acknowledge the help of Frank Allgöwer, Rolf Findeisen, Fabio Torrisi, Domenico Mignone, Pascal Grieder, Giancarlo Ferrari Trecate, Eric Kerrigan, Carlo Filippi, Luigi Glielmo and Francesco Vasca. A special thanks goes to Mato Baotic, whose help was essential in speeding up the most recent part of my research.

Thanks also to Gianni, Anna, Agata, Antonio, Gennaro and Peppe for living with me the world beyond science.

This research was supported by the Swiss National Science Foundation, the Federal Office for Education and Science through the Esprit Project 26270 VHS (Verification of Hybrid Systems) and ABB.

Contents

Part II Optimal Control of Linear Systems

Part III Optimal Control of Hybrid Systems

Part IV Applications

Preface

Many practical control problems are dominated by characteristics like

- state, input and operational constraints,
- switches between different operating regimes, and
- the interaction of continuous-time and discrete event systems.

At present no methodology is available to design controllers for such systems in a systematic manner. The most common approach resorts to using tools developed for unconstrained linear systems, patched with a collection of heuristic rules.

This book introduces a design theory for controllers for *constrained* and *switching* dynamical systems. It leads to algorithms which systematically solve control synthesis problems for classes of systems, where there are few, or no tools, currently available.

We will focus on two classes of *discrete-time* dynamical systems: *(i)* constrained linear systems and *(ii)* constrained linear hybrid systems, i.e., systems that include logic states and inputs, whose mode of operation can switch between a finite number of affine systems and where the mode transitions can be triggered by states crossing specific thresholds or by exogenous inputs.

For these two classes of systems we study *optimal control problems* and their state feedback solution. Our approach will make use of *multiparametric programming* and our main objective will be to derive *properties* of the state feedback solution, as well as to obtain *algorithms* to compute it efficiently.

We start by extending the theory of the Linear Quadratic Regulator to linear systems with linear constraints. We consider other norms in the objective function, we solve the robust case with additive and parametric uncertainty and finally extend all these results to hybrid systems. In the concluding part of the book, the applicability of the theory is demonstrated through two experimental case studies: a mechanical laboratory process and a traction control system developed jointly with Ford Motor Company in Michigan.

The book is structured as follows.

The first part of the book is a self-contained introduction to multi-parametric programming. In our framework, parametric programming is the main technique used to study and compute state feedback optimal control laws. In fact, we formulate the finite time optimal control problems as mathematical programs where the input sequence is the optimization vector. Depending on the dynamical model of the system, the nature of the constraints, and the cost function used, a different mathematical program is obtained. The current state of the dynamical system enters the cost function and the constraints as a parameter that affects the solution of the mathematical program. We study the structure of the solution as this parameter changes and we describe algorithms for solving multi-parametric linear, quadratic and mixed integer programs. They constitute the basic tools for computing the state feedback optimal control laws for these more complex systems in the same way as algorithms for solving the Riccati equation are the main tools for computing optimal controllers for linear systems.

In the second part of the book we focus on linear systems with polyhedral constraints on inputs and states. We study *finite time and infinite time* optimal control problems with cost functions based on 2, 1 and ∞ norms. We demonstrate that the solution to all these optimal control problems can be expressed as a *piecewise affine* state feedback law. Moreover, the optimal control law is continuous and the value function is convex and continuous. The results form a natural extension of the theory of the Linear Quadratic Regulator to constrained linear systems. They also have important consequences for the implementation of receding horizon control laws. Precomputing off-line the explicit piecewise affine feedback policy reduces the on-line computation for the receding horizon control law to a function evaluation, therefore avoiding the on-line solution of a mathematical program as is done in *model predictive control*. The evaluation of the piecewise affine optimal feedback policy is carefully studied and we propose algorithms to reduce its storage demands and computational complexity.

We also address the robustness of the optimal control laws. For uncertain linear systems with polyhedral constraints on inputs and states, we develop an approach to compute the state feedback solution to min-max control problems with a linear performance index. Robustness is achieved against additive norm-bounded input disturbances and/or polyhedral parametric uncertainties in the state-space matrices. We show that the robust optimal control law over a finite horizon is a continuous piecewise affine function of the state vector and that the value function is convex and continuous.

In the third part of the book we focus on linear hybrid systems. We give an introduction to the different formalisms used to model hybrid systems focusing on computation-oriented models. We study finite time optimal control problems with cost functions based on 2, 1 and ∞ norms. The optimal control law is shown to be, in general, piecewise affine over non-convex and disconnected sets. Along with the analysis of the solution properties we present algorithms that efficiently compute the optimal control law for all the considered cases.

In the forth and last part of the book the applicability of the theoretical results is demonstrated on a mechanical laboratory process and a traction control system developed jointly with Ford Motor Company in Michigan.

Francesco Borrelli

Notation and Definitions

Notation

Let $M \in \mathbb{R}^{m \times n}$, $Q \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$.

- M' denotes the matrix M transpose.
- $Q \succeq 0$ if and only if (iff) $x'Qx \geq 0 \; \forall x \in \mathbb{R}^n$. $Q \succ 0$ iff $x'Qx > 0 \; \forall x \in \mathbb{R}^n \backslash 0$.
- $\|Qx\|_2$ denotes the squared Euclidean norm of the vector x weighted with the matrix Q, i.e., $\|Qx\|_2 = x'Qx$. $\|x\|$ denotes the squared Euclidean norm of the vector x.
- $\|Mx\|_1$ and $\|Mx\|_\infty$ denote the 1-norm and ∞-norm of the vector Mx, respectively, i.e., $\|Mx\|_1 = |M_1x| + \ldots + |M_mx|$ and $\|Mx\|_\infty = \max\{|M_1x|, \ldots, |M_mx|\}$, where M_i denotes the i-th row of the matrix M.
- Let $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$. The inequality $u \leq v$ will be satisfied iff $u_i \leq v_i$, $\forall i = 1, \ldots, n$ where u_i and v_i denote the i-th component of the vectors u and v, respectively.
- Let $u \in \mathbb{R}^n$ and $c \in \mathbb{R}$. The inequality $u \leq c$ will be satisfied iff $u_i \leq c$, $\forall i = 1, \ldots, n$ where u_i denotes the i-th component of the vector u.
- "CFTOC" will be used as the acronym of "Constrained Finite Time Optimal Control".
  "CROC" as the acronym of "Constrained Robust Optimal Control".
  "OL-CROC" as the acronym of "Open-Loop Constrained Robust Optimal Control".
  "CL-CROC" as the acronym of "Closed-Loop Constrained Robust Optimal Control".
  "LQR" as the acronym of "Linear Quadratic Regulator".
  "CLQR" as the acronym of "Constrained Linear Quadratic Regulator".
  "MPC" as the acronym of "Model Predictive Control".
  "RHC" as the acronym of "Receding Horizon Control".
  "LP" as the acronym of "Linear Program".
  "QP" as the acronym of "Quadratic Program".

Definitions

Definition 0.1. *A polyhedron is a set that equals the intersection of a finite number of closed halfspaces.*

Definition 0.2. *A non-convex polyhedron is a non-convex set given by the union of a finite number of polyhedra.*

Definition 0.3. *Given a polyhedron $\mathcal{P} = \{x | Ax \leq b\}$, the faces of the polyhedron are the sets described by the same set of inequalities with some inequalities holding with equality, i.e., $\{x | Bx = c, Dx \leq d\}$ where $A = [B \;\, D]$ and*

$b = [c\ d]'$. The faces of zero dimension are the extreme points of $\mathcal{P}$. A facet is a face of maximal dimension not equal to the polyhedron.

Definition 0.4. *Two polyhedra $\mathcal{P}_i$, $\mathcal{P}_j$ of $\mathbb{R}^n$ are called neighboring polyhedra if their interiors are disjoint and $\mathcal{P}_i \cap \mathcal{P}_j$ is $(n-1)$-dimensional (i.e., is a common facet).*

Definition 0.5. *A function $h(\theta) : \Theta \to \mathbb{R}$, is quasiconvex if for each $\theta_1,\ \theta_2 \in \Theta$, $h(\lambda\theta_1 + (1-\lambda)\theta_2) \leq \max\{h(\theta_1), h(\theta_2)\}$ for each $\lambda \in (0,1)$.*

Definition 0.6. *A collection of sets $R_1, \ldots, R_N$ is a* partition *of a set Θ if* (i) $\bigcup_{i=1}^{N} R_i = \Theta$, (ii) $R_i \cap R_j = \emptyset$, $\forall i \neq j$. *Moreover $R_1, \ldots, R_N$ is a* polyhedral partition *of a polyhedral set Θ if $R_1, \ldots, R_N$ is a partition of Θ and the $\bar{R}_i$'s are polyhedral sets, where $\bar{R}_i$ denotes the closure of the set R_i.*

Definition 0.7. *A collection of sets $R_1, \ldots, R_N$ is a* partition in the broad sense *of a set Θ if* (i) $\bigcup_{i=1}^{N} R_i = \Theta$, (ii) $(R_i \backslash \partial R_i) \cap (R_j \backslash \partial R_j) = \emptyset$, $\forall i \neq j$, *where ∂ denotes the boundary. Moreover $R_1, \ldots, R_N$ is a* polyhedral partition in the broad sense *of a polyhedral set Θ if $R_1, \ldots, R_N$ is a partition in the broad sense of Θ and the R_i's are polyhedral sets.*

Definition 0.8. *A function $h(\theta) : \Theta \to \mathbb{R}^k$, where $\Theta \subseteq \mathbb{R}^s$, is* piecewise affine (PWA) *if there exists a partition $R_1,\ldots,R_N$ of Θ and $h(\theta) = H^i\theta + k^i$, $\forall \theta \in R_i$, $i = 1, \ldots, N$.*

Definition 0.9. *A function $h(\theta) : \Theta \to \mathbb{R}^k$, where $\Theta \subseteq \mathbb{R}^s$, is* piecewise affine on polyhedra *(PPWA) if there exists a polyhedral partition $R_1,\ldots,R_N$ of Θ and $h(\theta) = H^i\theta + k^i$, $\forall \theta \in R_i$, $i = 1, \ldots, N$.*

Definition 0.10. *A function $h(\theta) : \Theta \to \mathbb{R}^k$, where $\Theta \subseteq \mathbb{R}^s$, is* piecewise quadratic (PWQ) *if there exists a partition $R_1,\ldots,R_N$ of Θ and $h(\theta) = \theta' H^i\theta + k^i\theta + l^i$, $\forall \theta \in R_i$, $i = 1, \ldots, N$.*

Definition 0.11. *A function $h(\theta) : \Theta \to \mathbb{R}^k$, where $\Theta \subseteq \mathbb{R}^s$, is* piecewise quadratic on polyhedra *(PPWQ) if there exists a polyhedral partition $R_1,\ldots,R_N$ of Θ and $h(\theta) = \theta' H^i\theta + k^i\theta + l^i$, $\forall \theta \in R_i$, $i = 1, \ldots, N$.*

A piecewise affine function defined over a partition in the broad sense may be multi-valued if it is discontinuous (because of the double definition of the function along the boundary of two regions). However, continuous piecewise affine functions can be easily defined over a partition in the broad sense. In particular, it is simpler to define a continuous PPWA function on a polyhedral partition in the broad sense. This allows one to avoid keeping track for all the polyhedra of the partition which facets belong to the polyhedra and which not. The same holds for PWQ functions.

Part I

Multiparametric Programming

1

Multiparametric Programming: a Geometric Approach

In this chapter we introduce the concept of multiparametric programming and we recall the main results of nonlinear parametric programming. Then, we describe three algorithms for multiparametric linear programs (mp-LP), multiparametric quadratic programs (mp-QP) and multiparametric mixed-integer linear programs (mp-MILP).

1.1 Introduction

The operations research community has addressed parameter variations in mathematical programs at two levels: *sensitivity analysis*, which characterizes the change of the solution with respect to small perturbations of the parameters, and *parametric programming*, where the characterization of the solution for a full range of parameter values is sought. More precisely, programs which depend only on one scalar parameter are referred to as *parametric programs*, while problems depending on a vector of parameters are referred to as *multi-parametric programs*.

There are several reasons to look for efficient solvers of multiparametric programs. Typically, mathematical programs are affected by uncertainties due to factors which are either unknown or that will be decided later. Parametric programming systematically subdivides the space of parameters into characteristic regions, which depict the feasibility and corresponding performance as a function of the uncertain parameters, and hence provide the decision maker with a complete map of various outcomes.

Our interest in multiparametric programming arises from the field of system theory and optimal control. For discrete time dynamical systems finite time constrained optimal control problems can be formulated as mathematical programs where the cost function and the constraints are functions of the initial state of the dynamical system. In particular, Zadeh and Whalen [160] appear to have been the first ones to express the optimal control problem for constrained discrete time linear systems as a linear program. By using multiparametric programming we can characterize and compute the solution of the optimal control problem explicitly as a function of the initial state.

We are also motivated by the so-called *model predictive control* (MPC) technique. MPC is very popular in the process industry for the automatic regulation of process-units under operating constraints [117], and has attracted a considerable research effort in the last decade, as recently surveyed in [112]. MPC requires an optimization problem to be solved on-line in order to compute the next command action. Such an optimization problem depends on the current sensor measurements. The computation effort can be moved off-line by solving multiparametric programs, where the command inputs are the optimization variables and the measurements are the parameters [18].

The first method for solving *parametric linear programs* was proposed by Gass and Saaty [72], and since then extensive research has been devoted to sensitivity and (multi)-parametric analysis, as testified by the hundreds of references in [68] (see also [69] for recent advances in the field).

Multiparametric analysis makes use of the concept of a *critical region*. Given a parametric program, a critical region is a set of the parameters space were the local conditions for optimality remain unchanged.

The first method for solving *multiparametric linear programs* was formulated by Gal and Nedoma [70]. The method constructs the critical regions iteratively, by visiting the graph of bases associated with the LP tableau of the original problem. Subsequently only a few authors have dealt with multiparametric linear programming [68, 65, 132].

The first method for solving *multiparametric quadratic programs* was proposed by Bemporad and coauthors in [25]. The method constructs a critical region in a neighborhood of a given parameter, by using Karush-Kuhn-Tucker conditions for optimality, and then recursively explores the parameter space outside such a region. Other algorithms for solving mp-QPs appeared in [135, 147, 10] and will be reviewed briefly in Section 1.4.6.

In [1, 57] two approaches were proposed for solving mp-MILP problems. In both methods the authors use an mp-LP algorithm and a branch and bound strategy that avoids the complete enumeration of combinations of 0-1 integer variables by comparing the available bounds on the multiparametric solutions.

In this chapter we first recall the main results of nonlinear multiparametric programming [64], then we describe three algorithms for solving multiparametric linear programs (mp-LP), multiparametric quadratic programs (mp-QP) and multiparametric mixed-integer linear programs (mp-MILP).

The main idea of the three multiparametric algorithms presented in this chapter is to construct a critical region in a neighborhood of a given parameter, by using necessary and sufficient conditions for optimality, and then to recursively explore the parameter space outside such a region. For this reason the methods are classified as "geometric". All the algorithms are extremely simple to implement once standard solvers are available: linear programming, quadratic programming and mixed-integer linear programming for solving mp-LP, mp-QP and mp-MILP, respectively.

Note that though an LP can be viewed as a special case of a QP by setting the Hessian H to zero, the results of [25] on the mp-QP presented in Section 1.4 are restricted to the case $H > 0$. As a matter of fact, mp-LP deserves a special analysis, which provides insight into the properties of mp-LP and, leads to a different algorithm than mp-QP, which is described in detail in Section 1.3.

1.2 General Results for Multiparametric Nonlinear Programs

Consider the nonlinear mathematical program dependent on a parameter x appearing in the cost function and in the constraints

$$\begin{aligned} J^*(x) = \inf_z \ & f(z,x) \\ \text{subj. to } & g(z,x) \le 0 \end{aligned} \tag{1.1}$$

where $z \in M \subseteq \mathbb{R}^s$ is the optimization variable, $x \in X \subseteq \mathbb{R}^n$ is the parameter, $f : \mathbb{R}^s \times \mathbb{R}^n \to \mathbb{R}$ is the cost function and $g : \mathbb{R}^s \times \mathbb{R}^n \to \mathbb{R}^{n_g}$ are the constraints.

A small perturbation of the parameter x in the mathematical program (1.1) can cause a variety of results. Depending on the properties of the functions f and g the solution $z^*(x)$ may vary smoothly or change drastically as a function of x. We denote by $R(x)$ the point-to-set map that assigns to a parameter x the set of feasible z, i.e.

$$R(x) = \{z \in M | g(z,x) \le 0\} \tag{1.2}$$

by K^* the set of feasible parameters, i.e.,

$$K^* = \{x \in \mathbb{R}^n | R(x) \ne \varnothing\} \tag{1.3}$$

by $J^*(x)$ the real-valued function which expresses the dependence on x of the minimum value of the objective function over K^*, i.e.

$$J^*(x) = \inf_z \{f(z,x) | z \in R(x)\} \tag{1.4}$$

and by $Z^*(x)$ the point-to-set map which expresses the dependence on x of the set of optimizers, i.e.,

$$Z^*(x) = \{z \in R(x) | f(z,x) \le J^*(x)\} \tag{1.5}$$

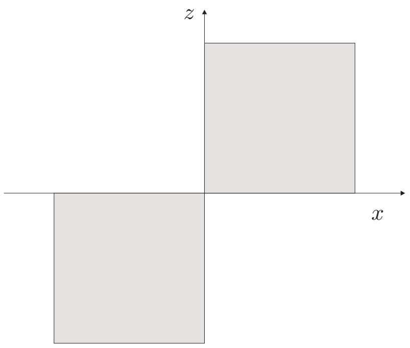
$J^*(x)$ will be referred to as optimal value function or simply value function, $Z^*(x)$ will be referred to as optimal set. If $Z^*(x)$ is a singleton for all x, then $z^*(x) \triangleq Z^*(x)$ will be called optimizer function. In this book we will assume that K^* is closed and $J^*(x)$ is finite for every x belonging to K^*. We denote by $g_i(z,x)$ the i-th component of the vector valued function $g(x,z)$.

In Chapter 2 of [64] Fiacco summarizes conditions under which the solution of nonlinear multiparametric programs (1.1) is locally well behaved and establish properties of the solution as a function of the parameters. The description of such conditions requires the definition of continuity of point-to-set maps. Before introducing this concept we will show thorough two simple examples that continuity of the constraints $g_i(z,x)$ with respect to x and z is not enough to imply any regularity of the value function and the optimizer function.

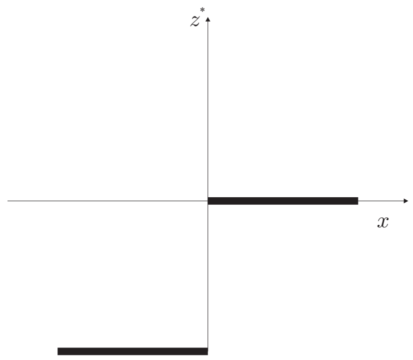
Example 1.1. Consider the following problem:

$$\begin{aligned} J^*(x) = \inf_z \ & z \\ \text{subj. to } & zx \ge 0 \\ & -10 \le z \le 10 \\ & -10 \le x \le 10 \end{aligned} \tag{1.6}$$

where $z \in \mathbb{R}$ and $x \in \mathbb{R}$. For each fixed x the set of feasible z is a segment. The point-to-map set $R(x)$ is plotted in Figure 1.1a. The function $g_1 : (z, x) \mapsto zx$ is continuous. Nevertheless, the value function $J^*(x) = z^*(x)$ has a discontinuity in the origin as can be seen in Figure 1.1b.



(a) Point to map set $R(x)$ corresponding to Example 1.1



(b) Value function z^* solution to Example 1.1

Fig. 1.1. Solution to Example 1.1

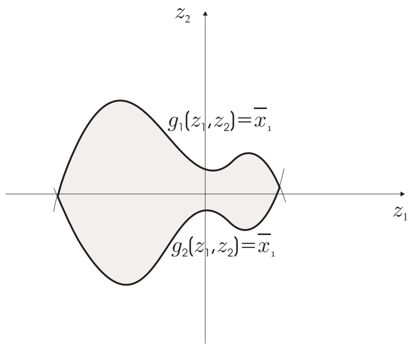
Example 1.2. Consider the following problem:

$$\begin{aligned} J^*(x) = \inf_{z_1, z_2} \ & -z_1 \\ \text{subj. to } & g_1(z_1, z_2) + x \leq 0 \\ & g_2(z_1, z_2) + x \leq 0 \end{aligned} \tag{1.7}$$

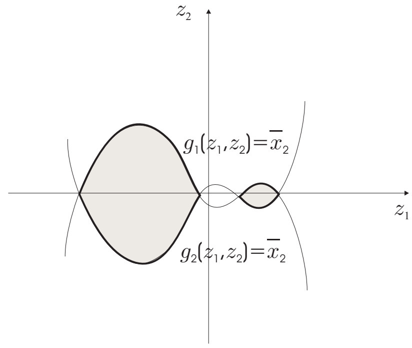
where examples of the functions $g_1(z_1, z_2)$ and $g_2(z_1, z_2)$ are plotted in Figures 1.2(a)–1.2(c). Figures 1.2(a)–1.2(c) also depicts the point to set map $R(x) = \{[z_1, z_2] \in \mathbb{R}^2 | g_1(z_1, z_2) + x \leq 0, \ g_2(z_1, z_2) + x \leq 0\}$ for three fixed x. Starting from $x = \bar{x}_1$ as x increases the domain of feasibility in the space z_1, z_2 shrinks; at the beginning it is connected (Figure 1.2(a)), then it becomes disconnected (Figure 1.2(b)) and eventually connected again (Figure 1.2(c)). No matter how smooth one chooses the functions g_1 and g_2, the value function $J^*(x) = z_1^*(x)$ will have a discontinuity at $x = \bar{x}_3$.

As mentioned before, the concept of continuity of point-to-set map is a critical requirement for the set $R(x)$ to lead to some regularity of the value and optimizer function. Consider a point-to-set map $R : x \in X \mapsto R(x) \subseteq M$. We give the following definitions:

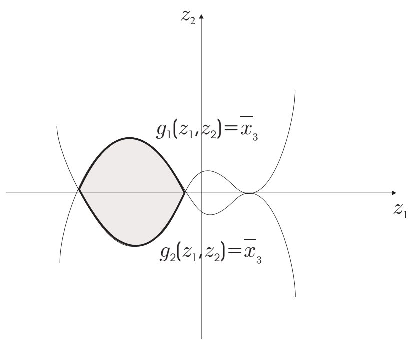
Definition 1.1. *$R(x)$ is open at a point $\bar{x} \in K^*$ if $\{x^k\} \subset K^*$, $x^k \to \bar{x}$ and $\bar{z} \in R(\bar{x})$ imply the existence of an integer m and a sequence $\{z^k\} \in M$ such that $z^k \in R(x^k)$ for $k \geq m$ and $z^k \to \bar{z}$*



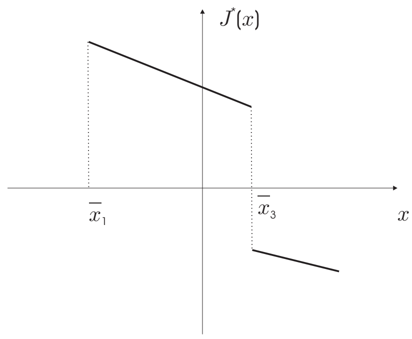
(a) Functions $g_1(z_1, z_2)$ and $g_2(z_1, z_2)$. In gray the set $R(\bar{x}_1)$ defined by problem (1.7)



(b) Set $R(\bar{x}_2)$ in Example 1.2



(c) Set $R(\bar{x}_3)$ in Example 1.2



(d) Value function

Fig. 1.2. Three projections of the point to map set defined by problem (1.7) in Example 1.2 and corresponding value function

Definition 1.2. *$R(x)$ is closed at a point $\bar{x} \in K^*$ if $\{x^k\} \subset K^*$, $x^k \to \bar{x}$, $z^k \in R(x^k)$, and $z^k \to \bar{z}$ imply $\bar{z} \in R(\bar{x})$.*

Definition 1.3. *$R(x)$ is continuous at a point $\bar{x}$ in K^* if it is both open and closed at $\bar{x}$. R is continuous in K^* if R is continuous for any x in K^*.*

The definitions above are illustrated through two examples.

Example 1.3. Consider

$$R(x) = \{z \in \mathbb{R} | z \in [0, 1] \text{ if } x < 1,\ z \in [0, 0.5] \text{ if } x \geq 1\}$$

The point-to-set map $R(x)$ is plotted in Figure 1.3. It easy to see that $R(x)$ is not closed but open. In fact, if one considers a sequence $\{x^k\}$ that converges to $\bar{x} = 1$ from the left and extracts the sequence $\{z^k\}$ plotted in Figure 1.3 converging to $\bar{z} = 0.75$, then $\bar{z} \notin R(\bar{x})$ since $R(1) = [0, 0.5]$.

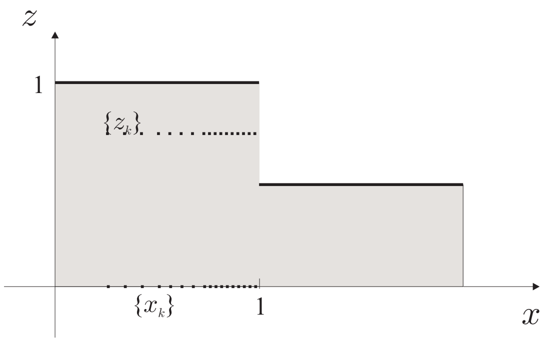


Fig. 1.3. Point to map set $R(x)$ corresponding to Example 1.3

Example 1.4. Consider

$$R(x) = \{z \in \mathbb{R} | z \in [0,1] \quad \text{if } x \leq 1, \; z \in [0, 0.5] \text{ if } x > 1\}$$

The point-to-set map $R(x)$ is plotted in Figure 1.4. It easy to verify that $R(x)$ is closed but not open. In fact, if one considers a sequence $\{x^k\}$ that converges to $\bar{x} = 1$ from the right and chooses $\bar{z} = 0.75 \in R(\bar{x})$, one is not able to construct a sequence $\{z^k\} \in M$ such that $z^k \in R(x^k)$ for and $z^k \to \bar{z}$

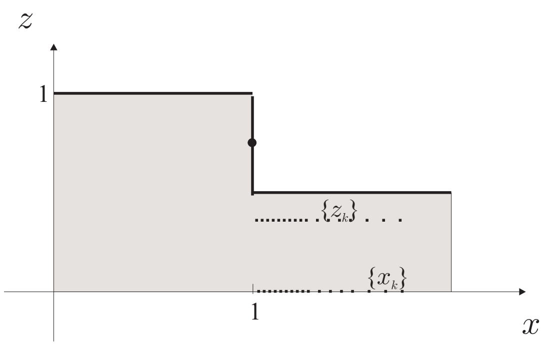


Fig. 1.4. Point to map set $R(x)$ corresponding to Example 1.4

The examples above are only illustrative. In general, it is difficult to test if a set is closed or open by applying the definitions. Several authors have proposed sufficient conditions on g_i which imply the continuity of $R(x)$. In the following we will summarize the main results of [131, 52, 88, 27].

Theorem 1.1. *If M is convex, if each component $g_i(z,x)$ of $g(z,x)$ is continuous on $M \times X$ and convex in z for each fixed $x \in X$ and if there exists a $\bar{z}$ such that $g(\bar{z},\bar{x}) < 0$, then $R(x)$ is a continuous map.*

The proof is given in [88]. Note that convexity in z for each x is not enough to imply the continuity of $R(x)$ everywhere in K^*. We remark that in Example 1.1 the origin does not satisfy the last hypothesis of Theorem (1.1).

Theorem 1.2. *If M is convex, if each component $g_i(z,x)$ of $g(x,z)$ is continuous on $M \times X$ and convex in z and x then $R(x)$ is a continuous map.*

The proof is simple and omitted here.

Now we are ready to give the two main theorems on the continuity of the value function and optimizer function.

Theorem 1.3. *Consider problem (1.1)–(1.2). If $R(x)$ is a continuous point-to-set map and $f(z,x)$ continuous, then $J^*(x)$ is continuous.*

Theorem 1.4. *Consider problem (1.1)–(1.2). If $R(x)$ is a continuous point-to-set map, $R(x)$ is convex for every $x \in K^*$, $f(z,x)$ continuous and strictly quasiconvex in z for each x, then $J^*(x)$ and $z^*(x)$ are continuous functions.*

Theorems 1.1 and 1.2 can be combined with Theorems 1.3 and 1.4 to get the following corollaries:

Corollary 1.1. *Consider the multiparametric nonlinear program (1.1). Assume that M is a compact convex set in $\mathbb{R}^s$, f and g are both continuous on $M \times \mathbb{R}^n$ and each component of g is convex on $M \times K^*$. Then $J^*(\cdot)$ is continuous for all $x \in K^*$*

Corollary 1.2. *Consider the multiparametric nonlinear program (1.1). Assume that M is a compact convex set in $\mathbb{R}^s$, f and g are both continuous on $M \times \mathbb{R}^n$ and each component of g is convex on M for each $x \in K^*$. Then $J^*(x)$ is continuous at x if it exists $\bar{z}$ such that $g(\bar{z},x) < 0$.*

1.3 Multiparametric Linear Programming

Consider the right-hand-side multiparametric linear program (mp-LP)

$$\begin{aligned} J^*(x) = \min_z \ & J(z,x) = c'z \\ \text{subj. to } & Gz \le W + Sx, \end{aligned} \tag{1.8}$$

where $z \in \mathbb{R}^s$ are the optimization variables, $x \in \mathbb{R}^n$ is the vector of parameters, $J(z,x) \in \mathbb{R}$ is the objective function and $G \in \mathbb{R}^{m\times s}$, $c \in \mathbb{R}^s$, $W \in \mathbb{R}^m$, and $S \in \mathbb{R}^{m\times n}$. Given a closed polyhedral set $K \subset \mathbb{R}^n$ of parameters,

$$K \triangleq \{x \in \mathbb{R}^n : \ Tx \le Z\}, \tag{1.9}$$

we denote by $K^* \subseteq K$ the region of parameters $x \in K$ such that the LP (1.8) is feasible. For any given $\bar{x} \in K^*$, $J^*(\bar{x})$ denotes the minimum value of the objective function in problem (1.8) for $x = \bar{x}$. The function $J^* : K^* \to \mathbb{R}$ will denote the function which expresses the dependence on x of the minimum value of the objective function over K^*, $J^*(\cdot)$ will be called value function. The set-valued function $Z^* : K^* \to 2^{\mathbb{R}^s}$, where $2^{\mathbb{R}^s}$ is the set of subsets of $\mathbb{R}^s$, will describe for any fixed $x \in K^*$ the set of optimizers $z^*(x)$ related to $J^*(\theta)$. We aim at determining the feasible region $K^* \subseteq K$ of parameters, the expression of the value function and the expression of one of the optimizer $z^*(x) \in Z^*(x)$.

We give the following definition of primal and dual degeneracy:

Definition 1.4. *For any given $x \in K^*$ the LP (1.8) is said to be primal degenerate if there exists a $z^*(x) \in Z^*(x)$ such that the number of active constraints at the optimizer is greater than the number of variables s.*

Definition 1.5. *For any given $x \in K^*$ the LP (1.8) is said to be dual degenerate if its dual problem is primal degenerate.*

The multiparametric analysis makes of the concept of *critical region* (CR). In [70] a critical region is defined as a subset of the parameter space on which a certain basis of the linear program is optimal. The algorithm proposed in [70] for solving multiparametric linear programs generates non-overlapping critical regions by generating and exploring the graph of bases. In the graph of bases the nodes represent optimal bases of the given multiparametric problem and two nodes are connected by an edge if it is possible to pass from one basis to another by one pivot step (in this case the bases are called neighbors). Our definition of critical regions is not associated with the bases but with the set of active constraints and is directly related to the definition given in [2, 113, 69, 65]. Below we give a definition of optimal partition directly related to that of Filippi [65], which is the extension of the idea in [2, 113] to the multiparametric case.

Let $J \triangleq \{1, \ldots, m\}$ be the set of constraint indices. For any $A \subseteq J$, let G_A and S_A be the submatrices of G and S, respectively, consisting of the rows indexed by A and denote with G_j, S_j and W_j the j-th row of G, S and W, respectively.

Definition 1.6. *The optimal partition of J associated with x is the partition $(A(x), NA(x))$ where*

$$\begin{array}{r}A(x) \triangleq \{j \in J : G_j z^*(x) - S_j x = W_j \text{ for all } z^*(x) \in Z^*(x)\} \\ NA(x) \triangleq \{j \in J : G_j z^*(x) - S_j x < W_j \text{ for some } z^*(x) \in Z^*(x)\}.\end{array}$$

It is clear that $(A(x), NA(x))$ are disjoint and their union is J. For a given $x^* \in K^*$ let $(A, NA) \triangleq (A(x^*), NA(x^*))$, and let

$$\begin{array}{r}CR_A \triangleq \{x \in K : A(x) = A\} \\ \overline{CR}_A \triangleq \{x \in K : A(x) \supseteq A\}.\end{array} \tag{1.10}$$

The set CR_A is the critical region related to the set of active constraints A, i.e., the set of all parameters x such that the constraints indexed by A are active at the optimum of problem (1.8). Clearly, $\overline{CR}_A \supseteq CR_A$.

The following result was proved by Filippi [65].

Theorem 1.5. *Let $(A, NA) \triangleq (A(x^*), NA(x^*))$ for some $x^* \in K$, and let d be the dimension of* range $G_A \bigcap$ range S_A*. If $d = 0$ then $CR_A = \{x^*\}$. If $d > 0$ then*

i) CR_A is an open[1] polyhedron of dimension d[2];
ii) $\overline{CR}_A$ is the closure of CR_A;
iii) every face of $\overline{CR}_A$ takes the form of $\overline{CR}_{A'}$ for some $A' \supseteq A$.

By Theorem 1.5 and the definition of critical regions in (1.10), it follows that the set K^* is always partitioned in a *unique* way. On the contrary, in the case of degeneracies, in the approach of [68] the partition is not uniquely defined, as it can be generated in a possibly exponentially number of ways, depending on the particular path followed by the algorithm to visit different bases [2, 28].

In this chapter, we aim at determining all the full-dimensional critical regions contained in K^* according to Definition 1.10. Rather than solving the problem by exploring the graph of bases of the associated LP tableau [70, 68], our approach is based on the direct exploration of the parameter space [25]. As will be detailed in the following sections, this has the following advantages: (*i*) a polynomial-time algorithm for LP can be used, (*ii*) degeneracy can be handled in a simpler way, (*iii*) the algorithm is easily implementable recursively, and (*iv*) the main ideas of the algorithm generalize to nonlinear multiparametric programming [25, 57]. In the absence of degeneracy the algorithm explores implicitly the graph of bases without resorting to pivoting. In case of degeneracy it avoids visiting the graph of degenerate bases. Therefore, the approach is different from other methods based on the simplex tableau [68].

---

[1] Given the polyhedron $B\xi \leq v$ we call open polyhedron the set $B\xi < v$.

[2] The dimension of a polyhedron $\mathcal{P}$ is defined here as the dimension of the smallest affine subspace containing $\mathcal{P}$.

Before going further, we recall some well known properties of the value function $J^*(x) : \mathbb{R}^n \to \mathbb{R}$ and of the set K^*.

Theorem 1.6. *(cf. [68, p. 178, Th. 1]). Assume that for a fixed $x^0 \in K$ there exists a finite optimal solution $z^*(x_0)$ of (1.8). Then, for all $x \in K$, (1.8) has either a finite optimum or no feasible solution.*

Theorem 1.7. *(cf. [68, p. 179, Th. 2]) Let $K^* \subseteq K$ be the set of all parameters x such that the LP (1.8) has a finite optimal solution $z^*(x)$. Then K^* is a closed polyhedral set in $\mathbb{R}^n$.*

The following Theorem 1.8 summarizes the properties enjoyed by the multiparametric solution, cf. [68, p. 180].

Theorem 1.8. *The function $J^*(\cdot)$ is convex and piecewise affine over K^* (and in particular affine in each critical region CR_{A_i}).*

If the optimizer $z^*(x)$ is unique for all $x \in K^*$, then the optimizer function $z^* : K^* \to \mathbb{R}^s$ is continuous and piecewise affine. Otherwise it is always possible to define a continuous and piecewise affine optimizer function $z^*(x) \in Z^*(x)$ for all $z \in K^*$.

In the next section we describe an algorithm to determine the set K^*, its partition into full-dimensional critical regions CR_{A_i}, the PWA value function $J^*(\cdot)$ and a PWA optimizer functions $z^*(\cdot)$.

1.3.1 Geometric Algorithm for mp-LP

The algorithm consists of two main steps, which can be summarized as follows:

1. Determine the dimension $n' \leq n$ of the smallest affine subspace $\mathcal{K}$ that contains K^*. If $n' < n$, find the equations in x which define $\mathcal{K}$.
2. Determine the partition of K^* into critical regions CR_{A_i}, and find the function $J^*(\cdot)$ and a PWA optimizer function $z^*(\cdot)$.

Below we give the details of the two steps. The first step is a preliminary one whose goal is to reduce the number of parameters in order to obtain a full-dimensional feasible region of parameters. This eases the second step, which computes the multiparametric solution and represents the core of the mp-LP algorithm.

1.3.2 Determining the Affine Subspace $\mathcal{K}$

In order to work with a minimal dimension of the parameter vector, the first step of the algorithm aims at finding the affine subspace $\mathcal{K} \subseteq \mathbb{R}^n$ containing the parameters x which render (1.8) feasible.

A first simple but important consideration concerns the column rank r_S of S. Clearly, if $r_S < n$, $n - r_S$ parameters can be eliminated by a simple coordinate transformation in $\mathbb{R}^n$. Therefore from now on, without loss of generality, we will assume that S has full column rank.

Besides this obvious preliminary reduction of the parameter space, there is another case where the number of parameters can be further reduced, as shown in the following example.

Example 1.5.

$$\begin{array}{ll} \min & 2z_1 + 3z_2 \\ \text{subj. to} & \begin{cases} z_1 + z_2 \leq 9 - x_1 - x_2 \\ z_1 - z_2 \leq 1 - x_1 - x_2 \\ z_1 + z_2 \leq 7 + x_1 + x_2 \\ z_1 - z_2 \leq -1 + x_1 + x_2 \\ \quad -z_1 \leq -4 \\ \quad -z_2 \leq -4 \\ \quad\ \ z_1 \leq 20 - x_2 \end{cases} \end{array} \tag{1.11}$$

where $K = \{x : \ -100 \leq x_1 \leq 100, \ -100 \leq x_2 \leq 100\}$. The reader can easily check that for any $x \in K^*$ the point $z = [4, 4]$ is the only feasible for problem (1.11), therefore the solution to (1.11) consists of one critical region:

$$z_1^* = 4 \ z_2^* = 4 \ \forall (x_1, x_2) \in CR_{\{1,2,3,4,5,6\}}$$

where $CR_{\{1,2,3,4,5,6\}}$ is

$$\begin{array}{c} x_1 + x_2 = 1 \\ -100 \leq x_1 \leq 100 \\ -100 \leq x_2 \leq 100. \end{array}$$

□

The example shows that, even if the matrix S has full column rank, the polyhedron K^* is contained in a sub-space of dimension $n' < n$, namely a line in $\mathbb{R}^2$.

Therefore, before solving the mp-LP problem, we need a test for checking the dimension n' of the smallest affine subspace $\mathcal{K}$ that contains K^*. Moreover, when $n' < n$, we need the equations describing $\mathcal{K}$ in $\mathbb{R}^n$. The equations are then used for a change of coordinates in order to reduce the number of parameters from n to n' and to get a polyhedron K^* that has full dimension in $\mathbb{R}^{n'}$.

Recall (1.8) and construct the LP problem in the space $\mathbb{R}^{s+n}$

$$\begin{array}{c} \min\limits_{z} J(z, x) = c'z \\ \text{subj. to } Gz - Sx \leq W. \end{array} \tag{1.12}$$

Clearly, the constraints in (1.12) define a polyhedron $\mathcal{P}$ in $\mathbb{R}^{s+n}$. The following lemma shows that the projection $\Pi_{\mathbb{R}^n}(\mathcal{P})$ of $\mathcal{P}$ on the parameter space $\mathbb{R}^n$ is K^*.

Lemma 1.1.

$$x^* \in K^* \iff \exists z : Gz - Sx^* \leq W \iff x^* \in \Pi_{\mathbb{R}^n}(\mathcal{P}).$$

As a direct consequence of Proposition 1.1, the dimension n' of the smallest affine subspace that contains K^* can be determined by computing the dimension of the projection $\Pi_{\mathbb{R}^n}(\mathcal{P})$.

Definition 1.7. *A* true inequality *of the polyhedron* $\mathcal{C} = \{\xi \in \mathbb{R}^s : \ B\xi \leq v\}$ *is an inequality* $B_i\xi \leq v_i$ *such that* $\exists \bar{\xi} \in \mathcal{C} : B_i\bar{\xi} < v_i$.

Given an H-representation of a polyhedron $\mathcal{C}$, i.e., a set of halfspaces defining $\mathcal{C}$, the following simple procedure determines the set I of all the not-true inequalities of $\mathcal{C}$.

Algorithm 1.3.1

Input: Polyhedron $\mathcal{C}$
Output: Set I of all the not-true inequalities of $\mathcal{C}$
let $I \leftarrow \emptyset$; $\mathcal{M} \leftarrow \{1, \ldots, m\}$;
while $\mathcal{M} \neq \emptyset$,
    let $j \leftarrow$ first element of $\mathcal{M}$;
    let $\mathcal{M} \leftarrow \mathcal{M} \setminus \{j\}$;
    solve the following LP problem:

$$\begin{aligned} \min \ & B_j\xi \\ s.t. \ & B_i\xi \leq v_i, \ \forall i \in \mathcal{M} \end{aligned} \tag{1.13}$$

    and **let** ξ^* be an optimizer;
    if $B_j\xi^* = v_j$ **then** $I \leftarrow I \cup \{j\}$;
    for $h \in \mathcal{M}$,
        if $B_h x^* < v_h$ **then** $\mathcal{M} \leftarrow \mathcal{M} \setminus \{h\}$;
    end ;
end .

Algorithm 1.3.2 describes a standard procedure to determine the dimension $n' \leq n$ of the smallest affine subspace $\mathcal{K}$ that contains K^*, and when $n' < n$ it finds the equations defining $\mathcal{K}$. In the following we will suppose without loss of generality that the set K is full-dimensional.

Algorithm 1.3.2

Input: Matrices G, S, W

Output: Dimension $n' \leq n$ of the smallest affine subspace $\mathcal{K}$ that contains K^* if $n' < n$, then $\mathcal{K} = \{x \in \mathbb{R}^n \; Tx = Z\}$

1 discard the true inequalities from the constraints in (1.12);

2 **if** no inequality is left **then** $\mathcal{K} \leftarrow \mathbb{R}^n$;

3 **else**

4 **let** $\mathcal{P}_a \triangleq \{(z,x) : \; G_a z - S_a x = W_a\}$ be the affine subspace obtained by collecting the remaining non-true inequalities;

5 **let** $\{u_1, \ldots, u_{k'}\}$ be a basis of the kernel of G'_a;

6 **if** $k' = 0$ **then** $\Pi_{\mathbb{R}^n}(\mathcal{P}_a)$ and (by Proposition (1.1)) K^* are full-dimensional, $\mathcal{K} \leftarrow \mathbb{R}^n$;

7 **else** $\mathcal{K} \leftarrow \{x | \; Tx = Z\}$, where

$$T = -\begin{bmatrix} u'_1 \\ \vdots \\ u'_{k'} \end{bmatrix} S_a, \; Z = \begin{bmatrix} u'_1 \\ \vdots \\ u'_{k'} \end{bmatrix} W_a; \tag{1.14}$$

8 **end** .

Step *5* can be computed by a standard singular value decomposition. When $k' > 0$, from the equations $Tx = Z$ a simple transformation (e.g. a Gauss reduction) leads to a new set of free n' parameters, where $n' = n - \operatorname{rank} T$. From now on, without loss of generality, we will assume that $\mathcal{K}$ is the whole space $\mathbb{R}^n$.

1.3.3 Determining the Critical Regions

In this section we detail the core of the mp-LP algorithm, namely the determination of the critical regions CR_{A_i} within the given polyhedral set K. We assume that there is neither primal nor dual degeneracy in the LP problems. These cases will be treated in Section 1.3.4. We denote by $z^* : K^* \rightarrow \mathbb{R}^s$ the real-valued optimizer function, where $Z^*(x) = \{z^*(x)\}$. The method uses primal feasibility to derive the H-polyhedral representation of the critical regions, the slackness conditions to compute the optimizer $z^*(x)$, and the dual problem to derive the optimal value function $J^*(x)$. The dual problem of (1.8) is defined as

$$\begin{aligned} \max_{y}\ & (W+Sx)'y \\ \text{subj. to } & G'y = c \\ & y \le 0. \end{aligned} \tag{1.15}$$

The primal feasibility, dual feasibility, and the slackness conditions for problems (1.8), (1.15) are

$$\text{PF:} \quad Gz \le W + Sx \tag{1.16a}$$
$$\text{DF:} \quad G'y = c, \quad y \le 0 \tag{1.16b}$$
$$\text{SC:} \quad (G_j z - W_j - S_j x)y_j = 0, \ \forall j \in J. \tag{1.16c}$$

Choose an arbitrary vector of parameters $x_0 \in \mathcal{K}$ and solve the primal and dual problems (1.8), (1.15) for $x = x_0$. Let z_0^* and y_0^* be the optimizers of the primal and the dual problem, respectively. The value of z_0^* defines the following optimal partition

$$\begin{aligned} \mathrm{A}(x_0) &\triangleq \{j \in J : G_j z_0^* - S_j x_0 - W_j = 0\} \\ \mathrm{NA}(x_0) &\triangleq \{j \in J : G_j z_0^* - S_j x_0 - W_j < 0\} \end{aligned} \tag{1.17}$$

and consequently the critical region $CR_{A(x_0)}$.

By hypothesis y_0^* is unique, and by definition of critical region y_0^* remains optimal for all $x \in CR_{A(x_0)}$. The value function in $CR_{A(x_0)}$ is

$$J^*(x) = (W + Sx)'y_0^* \tag{1.18}$$

which is an affine function of x on $CR_{A(x_0)}$, as was stated in Theorem 1.8. Moreover, for the optimal partition (1.17) the PF condition can be rewritten as

$$G_A z^*(x) = W_A + S_A x \tag{1.19a}$$
$$G_{NA} z^*(x) < W_{NA} + S_{NA} x. \tag{1.19b}$$

In the absence of dual degeneracy the primal optimizer is unique, and (1.19a) can be solved to get the solution $z^*(x)$. In fact, equations (1.19a) form a system of l equalities where, in the absence of primal degeneracy, $l = s$ is the number of active constraints. From (1.19a), it follows

$$z^*(x) = -G_A^{-1} S_A x + G_A^{-1} W_A = Ex + Q, \tag{1.20}$$

which implies the linearity of z^* with respect to x. From the primal feasibility condition (1.19b), we get immediately the representation of the critical region $CR_{A(x_0)}$

$$G_{NA}(Ex + Q) < W_{NA} + S_{NA} x. \tag{1.21}$$

The closure $\overline{CR}_{A(x_0)}$ of $CR_{A(x_0)}$ is obtained by replacing "$<$" by "$\le$" in (1.21).

Once the critical region $\overline{CR}_{A(x_0)}$ has been defined, the rest of the space $R^{\text{rest}} = K \backslash \overline{CR}_{A(x_0)}$ has to be explored and new critical regions generated. An effective approach for partitioning the rest of the space was proposed in [57] and formally proved in [25]. In the following we report the theorem that justifies such a procedure to characterize the rest of the region R^{rest}.

Theorem 1.9. *Let $Y \subseteq \mathbb{R}^n$ be a polyhedron, and $R_0 \triangleq \{x \in Y : Ax \leq b\}$ be a polyhedral subset of Y, where $b \in \mathbb{R}^{m\times 1}$, $R_0 \neq \emptyset$. Also let*

$$R_i = \left\{ x \in Y : \begin{array}{l} A^i x > b^i \\ A^j x \leq b^j, \forall j < i \end{array} \right\} i = 1, \ldots, m$$

where $b \in \mathbb{R}^{m\times 1}$ and let $R^{\text{rest}} \triangleq \cup_{i=1}^{m} R_i$. Then

(i) $R^{\text{rest}} \cup R_0 = Y$
(ii)$R_0 \cap R_i = \emptyset$, $R_i \cap R_j = \emptyset, \forall i \neq j$

i.e., $\{R_0, R_1, \ldots, R_m\}$ is a partition of Y.

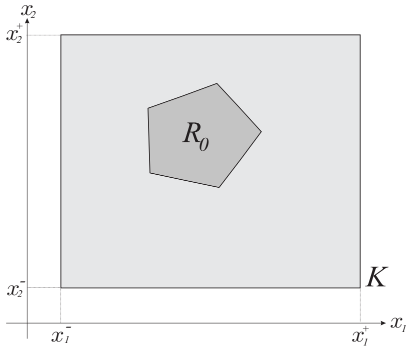
Proof: (i) We want to prove that given an $x \in Y$, then either x belongs to R_0 or to R_i for some i. If $x \in R_0$, we are done. Otherwise, there exists an index i such that $A^i x > b^i$. Let $i^* = \min_{i \leq m}\{i : A^i x > b^i\}$. Then $x \in R_{i^*}$, as $A^{i^*} x > b^{i^*}$ and $A^j x \leq b^j$, $\forall j < i^*$, by definition of i^*.

(ii) Let $x \in R_0$. Then there does not exist any i such that $A^i x > b^i$, which implies that $x \notin R_i$, $\forall i \leq m$. Let $x \in R_i$ and take $i > j$. Because $x \in R_i$, by definition of R_i $(i > j)$ $A^j x \leq b^j$, which implies that $x \notin R_j$. □

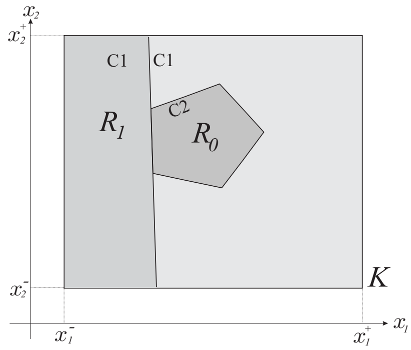
A Two Dimensional Example

In order to demonstrate the procedure proposed in Theorem 1.9 for partitioning the set of parameters K, consider the case when only two parameters x_1 and x_2 are present. As shown in Figure 1.5(a), K is defined by the inequalities $\{x_1^- \leq x_1 \leq x_1^+,\ x_2^- \leq x_2 \leq x_2^+\}$, and R_0 by the inequalities $\{C1 \leq 0,\ \ldots, C5 \leq 0\}$ where $C1$, …, $C5$ are linear in x. The procedure consists of considering one by one the inequalities which define R_0. Considering, for example, the inequality $C1 \leq 0$, the first set of the rest of the region $R^{\text{rest}} \triangleq K - R_0$ is given by $R_1 = \{C1 \geq 0,\ x_1 \geq x_1^-,\ x_2^- \leq x_2 \leq x_2^+\}$, which is obtained by reversing the sign of the inequality $C1 \leq 0$ and removing redundant constraints in K (see Figure 1.5(b)). Thus, by considering the rest of the inequalities, the complete rest of the region is $R^{\text{rest}} = \bigcup_{i=1}^{5} R_i$, where R_1, …, R_5 are given in Table 1.1 and are graphically reported in Figure 1.5(d). Note that for the case when K is unbounded, the inequalities involving K in Table 1.1 are simply suppressed.

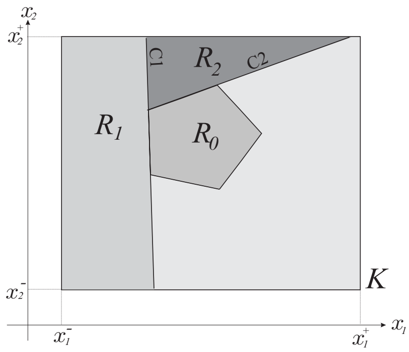
Remark 1.1. The procedure proposed in Theorem 1.9 for partitioning the set of parameters allows one to recursively explore the parameter space (see Remark 1.7 below). Such an iterative procedure terminates after a finite time,



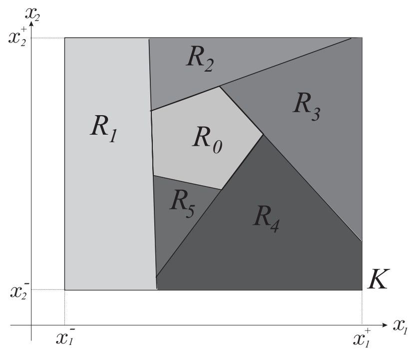
(a) Set of parameters K and initial region R_0



(b) Partition of R^{rest} - Step 1



(c) Partition of R^{rest} - Step 2



(d) Final partition of R^{rest}

Fig. 1.5. Two dimensional example: partition of the rest of the space $R^{\text{rest}} \triangleq K - R_0$

Region	Inequalities
R_1	$C1 \geq 0,\ x \geq x_1^-,\ x_2^- \leq x_2 \leq x_2^+$
R_2	$C2 \geq 0,\ C1 \leq 0,\ x_2 \leq x_2^+$
R_3	$C3 \geq 0,\ C2 \leq 0,\ x_2 \leq x_2^+,\ x_1 \leq x_1^+$
R_4	$C4 \geq 0,\ C1 \leq 0,\ C3 \leq 0,\ x_1 \leq x_1^+,\ x_2 \geq x_2^-$
R_5	$C5 \geq 0,\ C1 \leq 0,\ C4 \leq 0$

Table 1.1. Definition of the partition of $R^{\text{rest}} \triangleq K - R_0$

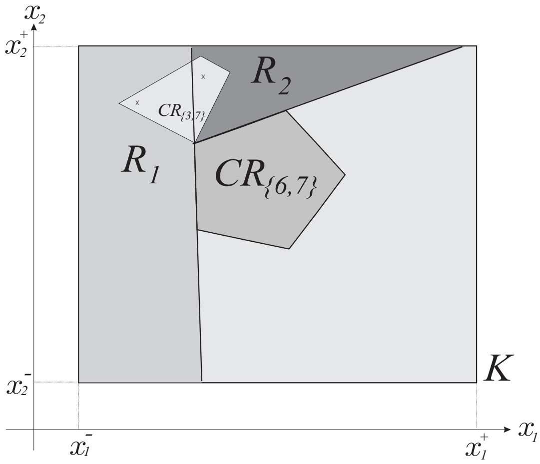


Fig. 1.6. Example: critical region explored twice

as the number of possible combinations of active constraints decreases with each iteration. However, this way of partitioning defines new polyhedral regions R_k to be explored that are not related to the critical regions which still need to be determined. This may split some of the critical regions, due to the artificial cuts induced by Theorem 1.9. In [25], post-processing is used to join cut critical regions. As an example, in Figure 1.6 the critical region $CR_{\{3,7\}}$ is discovered twice, one part during the exploration of R_1 and the second part during the exploration of R_2. Although algorithms exist for convexity recognition and computation of the union of polyhedra [21], the post-processing operation is computationally expensive. Therefore, in our algorithm the critical region obtained by (1.21) is not intersected with halfspaces generated by Theorem 1.9, which is only used to drive the exploration of the parameter space. As a result, no post processing is needed to join subpartitioned critical regions. On the other hand, some critical regions may appear more than once. Duplicates can be uniquely identified by the set of active constraints and can be easily eliminated. To this aim, in the implementation of the algorithm we keep a list of all the critical regions which have already been generated in order to avoid duplicates. In Figure 1.6 the critical region $CR_{\{3,7\}}$ is discovered twice but stored only once.

1.3.4 Degeneracy

Primal Degeneracy

By applying a Gauss reduction to (1.19a) we obtain

$$\begin{bmatrix} U & P \\ 0 & D \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} q \\ r \end{bmatrix}. \tag{1.22}$$

Assuming that U is nonsingular (the case $\det U = 0$, corresponding to dual degeneracy, will be addressed in Section (1.6), the optimizer is given by

$$z^*(x) = -U^{-1}Px + U^{-1}q = Ex + Q, \tag{1.23}$$

and in (1.18) one may choose any one of the dual optimizers y_0^* in order to characterize the value function. The H-polyhedral representation of the critical region $CR_{A(\theta_0)}$ is

$$G_{NA}(E\theta + Q) < W_{NA} + S_{NA}\theta \tag{1.24a}$$

$$D\theta = r. \tag{1.24b}$$

We distinguish between two cases:

Case 1. Matrix D is the null matrix and r is the null vector. Then we have a full-dimensional primal degenerate critical region $CR_{A(x_0)}$.

Case 2. The rank of D is $p > 0$. Then, $CR_{A(x_0)}$ has dimension $n' - p < n' = \dim(\mathcal{K})$. By Theorem 1.5 and the minimality of the dimension n' of $\mathcal{K}$ determined by Algorithm 1.3.1, we conclude that $CR_{A(x_0)}$ is an $(n' - p)$-dimensional face of another critical region $CR_{A'}$ for some combination $A' \supset A(x_0)$.

Remark 1.2. Note that case 2 occurs only if the chosen parameter vector x_0 lies on the face of two or more neighboring full-dimensional critical regions, while case 1 occurs when a full-dimensional set of parameters makes the LP problem (1.8) primal degenerate.

Remark 1.3. If case 2 occurs, to avoid further recursion of the algorithm not producing any full-dimensional critical region, and therefore lengthen the number of steps required to determine the solution to the mp-LP, we perturb the parameter x_0 by a random vector $\epsilon \in \mathbb{R}^n$, where

$$\|\epsilon\|_2 < \min_i\{\tfrac{|T_i x_0 - Z_i|}{\sqrt{T_i T_i'}}\}, \tag{1.25}$$

$\|\cdot\|_2$ denotes the standard Eucledian norm and $R_k = \{x : Tx \le Z\}$ is the polyhedral region where we are looking for a new critical region. Equation (1.25) ensures that the perturbed vector $x_0 = x_0 + \varepsilon$ is still contained in R_k.

Example 1.6. Consider the mp-LP problem

$$
\begin{array}{ll}
\min & z_1+z_2+z_3+z_4 \\
\text{subj. to} & \left\{
\begin{array}{r}
-z_1+z_5 \leq 0 \\
-z_1-z_5 \leq 0 \\
-z_2+z_6 \leq 0 \\
-z_2-z_6 \leq 0 \\
-z_3 \leq x_1+x_2 \\
-z_3-z_5 \leq x_2 \\
-z_3 \leq -x_1-x_2 \\
-z_3+z_5 \leq -x_2 \\
-z_4-z_5 \leq x_1+2x_2 \\
-z_4-z_5-z_6 \leq x_2 \\
-z_4+z_5 \leq -1x_1-2x_2 \\
-z_4+z_5+z_6 \leq -x_2 \\
z_5 \leq 1 \\
-z_5 \leq 1 \\
z_6 \leq 1 \\
-z_6 \leq 1
\end{array}
\right.
\end{array}
\tag{1.26}
$$

where K is given by

$$
\begin{array}{c}
-2.5 \leq x_1 \leq 2.5 \\
-2.5 \leq x_2 \leq 2.5.
\end{array}
\tag{1.27}
$$

A solution to the mp-LP problem (the mp-LP is also dual degenerate) is shown in Figure 1.7 and the constraints which are active in each associated critical region are reported in Table 1.2. Clearly, as $z \in \mathbb{R}^6$, $CR6$ and $CR11$ are primal degenerate full-dimensional critical regions.

In this examples dual degeneracy occurs as well. In particular, critical regions CR5-CR6-CR7, CR10-CR11-CR12, CR2-CR8 and CR1-CR13 form a group of 4 dual degenerate regions. Dual degeneracy will be discussed into details next.

Dual Degeneracy

If dual degeneracy occurs, the set $Z^*(x)$ may not be a singleton for some $x \in K^*$, and therefore the inequalities defining the critical region cannot be simply determined by substitution in (1.21). In order to get such inequalities, one possibility is to project the polyhedron defined by the equality and inequality (1.19) onto the parameter space (for efficient tools to compute the projection of a polyhedron see e.g. [67]), which however does not directly allow defining an optimizer $z^*(\cdot)$. In order to compute $z^*(x)$ in the dual degenerate region $CR_{\{A(x_0)\}}$ one can simply choose a particular optimizer on a vertex of the feasible set, determine set $\widehat{A}(x_0)$ of active constraints for which $G_{\widehat{A}(x_0)}$ is full rank, and compute a subset $\widehat{CR}_{\widehat{A}(x_0)}$ of the dual degenerate critical region (namely, the subset of parameters x such that only the constraints $\widehat{A}(x_0)$ are active at the optimizer, which is not a critical region in the sense of

Critical Region	Value function
CR1= $CR_{\{2,3,4,7,10,11\}}$	$2x_1+3x_2$
CR2= $CR_{\{1,3,4,5,9,13\}}$	$-2x_1-3x_2$
CR3= $CR_{\{1,3,4,6,9,13\}}$	$-x_1-3x_2-1$
CR4= $CR_{\{1,3,6,9,10,13\}}$	$-2x_2-1$
CR5= $CR_{\{1,2,3,7,10,11\}}$	x_1
CR6= $CR_{\{1,3,6,7,9,10,11,12\}}$	x_1
CR7= $CR_{\{1,3,7,10,11,15\}}$	x_1
CR8= $CR_{\{1,3,4,5,9,12\}}$	$-2x_1-3x_2$
CR9= $CR_{\{2,4,8,11,12,14\}}$	$2x_2-1$
CR10= $CR_{\{1,2,4,5,9,12\}}$	$-x_1$
CR11= $CR_{\{2,4,5,8,9,10,11,12\}}$	$-x_1$
CR12= $CR_{\{2,4,5,9,12,16\}}$	$-x_1$
CR13= $CR_{\{2,3,4,7,11,14\}}$	$2x_1+3x_2$
CR14= $CR_{\{2,3,4,8,11,14\}}$	x_1+3x_2-1

Table 1.2. Critical regions and corresponding value function for Example 1.6

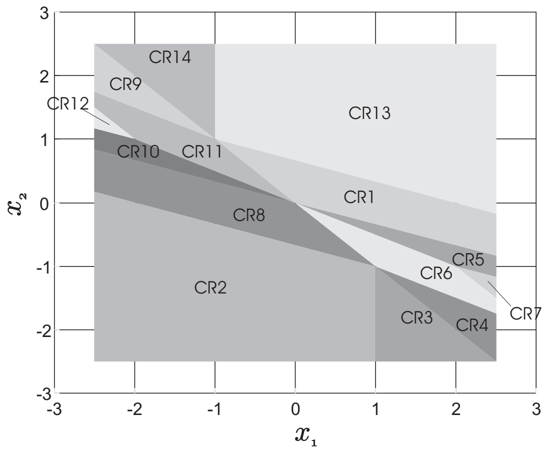


Fig. 1.7. Polyhedral partition of the parameter space corresponding to the solution of Example 1.6

Definition 1.6). The algorithm proceeds by exploring the space surrounding $\widehat{CR}_{\hat{A}(x_0)}$ as usual. The arbitrariness in choosing an optimizer leads to different ways of partitioning $CR_{\{A(x_0)\}}$, where the partitions can be simply calculated from (1.20) and (1.21) and, in general, may overlap. Nevertheless, in each region a unique optimizer is defined. The storing of overlapping regions can be avoided by intersecting each new region (inside the dual degenerate region) with the current partition computed so far. This procedure is illustrated in the following example.

Example 1.7. Consider the following mp-LP reported in [68, page 152]

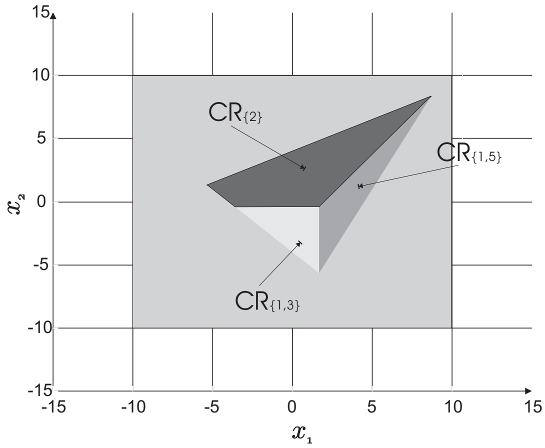


Fig. 1.8. Polyhedral partition of the parameter space corresponding to the solution of Example 1.7

$$
\begin{array}{ll}
\min & -2z_1 - z_2 \\
\text{subj. to} & \begin{array}{r} z_1 + 3z_2 \leq 9 - 2x_1 + 1x_2 \\ 2z_1 + z_2 \leq 8 + x_1 - 2x_2 \\ z_1 \leq 4 + x_1 + x_2 \\ -z_1 \leq 0 \\ -z_2 \leq 0 \end{array}
\end{array} \tag{1.28}
$$

where K is given by:

$$
\begin{array}{l}
-10 \leq x_1 \leq 10 \\
-10 \leq x_2 \leq 10.
\end{array} \tag{1.29}
$$

The solution is represented in Figure 1.8 and the critical regions are listed in Table 1.3.

Region	Optimum	Optimal value
$CR_{\{2\}}$	not single valued	$-x_1 + 2x_1 - 8$
$CR_{\{1,5\}}$	$z_1^* = -2x_1 + x_2 + 9,\ z_2^* = 0$	$4x_1 - 2x_2 - 18$
$CR_{\{1,3\}}$	$z_1^* = x_1 + x_2 + 4,\ z_2^* = -x_1 + 1.6667$	$-x_1 - 2x_2 - 9.6667$

Table 1.3. Critical regions and corresponding optimal value for Example 1.7

The critical region $CR_{\{2\}}$ is related to a dual degenerate solution with multiple optima. The analytical expression of $CR_{\{2\}}$ is obtained by projecting the H-polyhedron

$$\begin{array}{ll} z_1 + 3z_2 + 2x_1 - 1x_2 & < 9 \\ 2z_1 + z_2 - x_1 + 2x_2 & = 8 \\ z_1 - x_1 - x_2 & < 4 \\ -z_1 & < 0 \\ -z_2 & < 0 \end{array} \tag{1.30}$$

on the parameter-space to obtain:

$$\overline{CR}_{\{2\}} = \left\{ (x_1, x_2) : \begin{array}{ll} 2.5x_1 - 2x_2 & \leq 5 \\ -0.5x_1 + x_2 & \leq 4 \\ -12x_2 & \leq 5 \\ -x_1 - x_2 & \leq 4 \end{array} \right\} \tag{1.31}$$

For all $x \in CR_{\{2\}}$, only one constraint is active at the optimum, which makes the optimizer not unique.

Figures 1.9 and 1.10 show two possible ways of covering $CR_{\{2\}}$. The generation of overlapping regions is avoided by intersecting each new region with the current partition computed so far, as shown in Figure 1.11 where $\widetilde{CR}_{\{2,4\}}$, $\widetilde{CR}_{\{2,1\}}$ represents the intersected critical region.

Remark 1.4. In case of dual degeneracy, the method of [68] explores one of several possible paths on the graph of the bases. The degenerate region is split into non-overlapping sub-regions, where an optimizer is uniquely defined. As a result, the method of [68] provides a continuous mapping $z^*(x)$ inside the dual degenerate critical region.

Similarly as in [68], in our algorithm the particular choice of the optimal vertices determines the way of partitioning the degenerate critical region. However, this choice is arbitrary and therefore our method may lead to discontinuities over the artificial cuts inside the dual degenerate critical region in case overlapping regions are found.

In Figure 1.9, the regions are overlapping, and in Figure 1.11 artificial cuts are introduced, with consequent discontinuity of $z^*(x)$ at the boundaries inside the degenerate critical region $CR_{\{2\}}$. On the contrary, no artificial cuts are introduced in Figure 1.10. Therefore the mapping $z^*(x)$ is continuous over $CR_{\{2\}}$.

1.3.5 A Summary of the mp-LP Algorithm

Based on the above discussion, the mp-LP solver can be summarized in the following recursive Algorithm 1.3.3. Note that the algorithm generates a partition of the state space in the broad sense. The algorithm could be modified to store the critical regions as defined in (1.10) which are open sets, instead of storing its closure. In this case the algorithm has to explore and store all the critical regions that are not full-dimensional in order to generate a partition of the set of feasible parameters. From a practical point of view such procedure is not necessary since the value function and the optimizer are continuous functions of x.

Algorithm 1.3.3

Input: Matrices cc, G, W, S of the mp-LP (1.8) and set K in (1.9)
Output: Multiparametric solution to the mp-LP (1.8)

1 **let** $Y_k \leftarrow K$ be the current region to be explored.
2 **let** x_0 be in the interior of Y_k;
3 solve the LP (1.8), (1.15) for $x = x_0$;
4 **if** the optimizer is not unique, **then** let z_0^* be one of the optimal vertices of the LP (1.8) for $x = x_0$ **endif**
5 **let** $A(x_0)$ be the set of active constraints as in (1.17);
6 **if** there is primal degeneracy **then**
7   **let** U, P, D matrices as in (1.22) after a Gauss reduction to (1.19a);
8   determine $z^*(x)$ from (1.23) and $CR_{A(x_0)}$ from (1.24);
9   choose y_0^* among one of the possible optimizers;
10 **else**
11   determine $z^*(x)$ from (1.20) and $CR_{A(x_0)}$ from (1.21);
12 **endif** ;
13 **let** $J^*(x)$ as in (1.18) for $x = x_0$;
14 partition the rest of the region as in Theorem 1.9;
15 **for** each nonempty element CR_i of the partition **do**
16   $Y_k \leftarrow CR_i$ and **go to** Step 2;
17 **endfor** .

Remark 1.5. As remarked in Section 1.3.4, if rank(D) > 0 in step *7*, the region $CR_{A(x_0)}$ is not full-dimensional. To avoid further recursion in the algorithm which does not produce any full-dimensional critical region, after computing U, P, D if $D \neq 0$ one should compute a random vector $\epsilon \in \mathbb{R}^n$ satisfying (1.25) and such that the LP (1.8) is feasible for $x_0 + \epsilon$ and then repeat step *5* with $x_0 \leftarrow x_0 + \epsilon$.

Remark 1.6. Note that step *4* can be easily executed by using an active set method for solving the LP (1.8) for $x = x_0$. Note also that primal basic solutions are needed only to define optimizers in a dual degenerate critical region.

As remarked in the previous section, if one is only interested in characterizing the dual degenerate critical region, without characterizing one of the possible optimizer function $x^*(\cdot)$, step *4* can be avoided and instead of executing steps *6-12* one can compute the projection $CR_{A(\theta_0)}$ of the polyhedron (1.19) on K (note that $A(\theta_0)$ has to be the set of active constraints as defined in (1.6)).

Remark 1.7. The algorithm determines the partition of K recursively. After the first critical region is found, the rest of the region in K is partitioned into polyhedral sets $\{R_i\}$ as in Theorem 1.9. By using the same method, each set R_i is further partitioned, and so on. This can be represented as a search tree, with a maximum depth equal to the number of combinations of active constraints (see Section (1.3.6) below).

1.3.6 Complexity Analysis

Algorithm 1.3.3 solves an mp-LP by partitioning K^* into N_n convex polyhedral regions CR_{A_i}. During the execution, $N_r \geq N_n$ regions R_k are generated. N_r and N_n depend on the dimension s of the optimization vector, the number n of parameters, and the number m of constraints in the optimization problem (1.8). For the mono-parametric case, [119] proves that the complexity of the algorithm is exponential.

In an LP, in the absence of degeneracy, the optimizer is a vertex of the feasible set, and therefore at least s constraints must be active. The number of possible combinations of active constraints at the solution of an LP (i.e., the number of combinations of s constraints out of a set of m) is

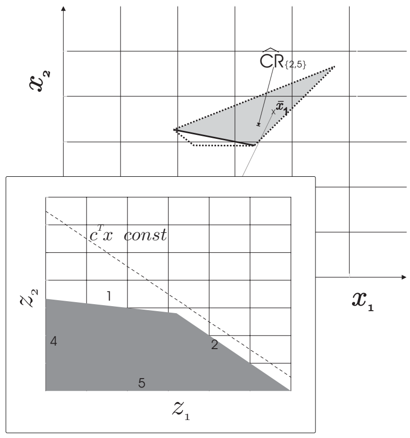
$$M = \binom{m}{s} = \frac{m!}{(m-s)!s!} \tag{1.32}$$

and represents an upper-bound on the number of different critical regions that are generated by the algorithm, i.e., $N_n \leq M$.

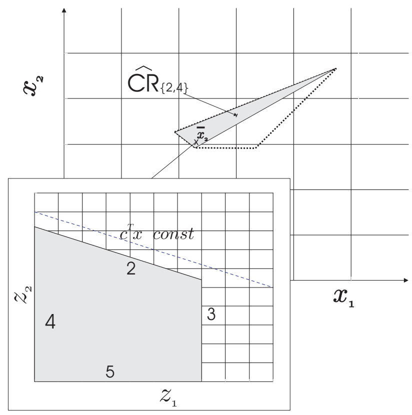
An upper-bound to N_n can be found by using the approach of [132], where the number N_n of critical regions is shown to be less than or equal to the number μ of extreme points of the feasible region of the dual problem (1.15), where μ does not depend on S and n. In the worst case, the feasible region is a polyhedron in $\mathbb{R}^m$ with $s+m$ facets. By recalling the result in [77] for computing an upper-bound to the number of extreme points of a polyhedron, we obtain

$$N_n \leq \mu \leq \binom{s+m-\lceil m/2 \rceil}{\lfloor m/2 \rfloor} + \binom{s+m-1-\lceil (m-1)/2 \rceil}{\lfloor (m-1)/2 \rfloor}. \tag{1.33}$$

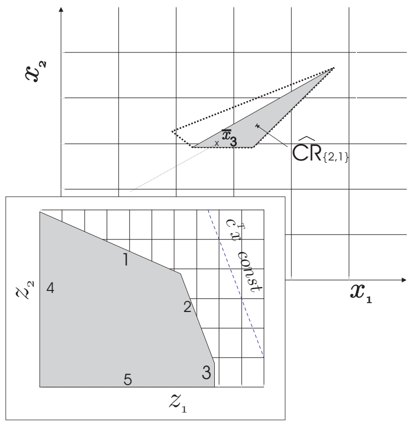
The number of regions N_r that can be generated by the algorithm is finite and therefore the algorithm will terminate in finite time. A worst-case estimate of N_r can be computed from the way Algorithm 1.3.3 generates regions R_i to explore the set of parameters K. The following analysis does not take into account that (*i*) redundant constraints are removed, and that (*ii*) possible empty sets are not further partitioned. The first region $R_0 = CR_{A(x_0)}$ is defined by the constraints $Gz(x) \leq W + Sx$ (m constraints). If there is no dual degeneracy and no primal degeneracy only s constraints can be active, and hence $CR_{A(x_0)}$ is defined by $q = m - s$ constraints. From Theorem 1.9, R^{rest} consists of q convex polyhedra $\{R_i\}$, each one defined by at most q inequalities. For each R_i, a new critical region CR is determined which consists of at most $2q$ inequalities (the additional q inequalities come from the condition $CR \subseteq R_i$), and therefore the corresponding R^{rest} partition includes at most $2q$ sets, each one defined by at most $2q$ inequalities. As mentioned above, this way of generating regions can be associated with a search tree. By induction, it is easy to prove that at the tree level $k+1$ there are $k!m^k$ regions, each one defined by at most $(k+1)q$ constraints. As observed earlier, each R is the largest set corresponding to a certain combination of active constraints. Therefore, the search tree has a maximum depth of μ, as at each level there is one admissible combination less. In conclusion, the number of regions is $N_r \leq \sum_{k=0}^{\mu-1} k!q^k$, each one defined by at most μq linear inequalities.



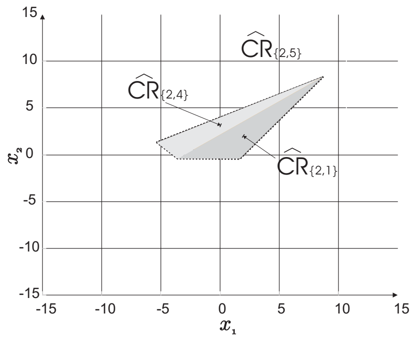
(a) First region $\widehat{CR}_{\{2,5\}} \subset CR_{\{2\}}$, and below the feasible set in the z-space corresponding to $\bar{x}_1 \in \widehat{CR}_{\{2,5\}}$



(b) Second region $\widehat{CR}_{\{2,4\}} \subset CR_{\{2\}}$, and below the feasible set in the z-space corresponding to $\bar{x}_2 \in \widehat{CR}_{\{2,4\}}$

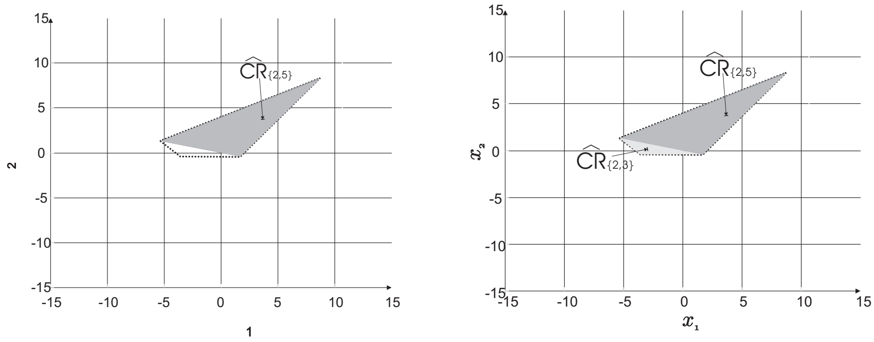


(c) Third region $\widehat{CR}_{\{2,1\}} \subset CR_{\{2\}}$, and below the feasible set in the z-space corresponding to $\bar{x}_3 \in \widehat{CR}_{\{2,1\}}$



(d) Final partition of $CR_{\{2\}}$. Note that the region $\widehat{CR}_{\{2,5\}}$ is hidden by region $\widehat{CR}_{\{2,4\}}$ and region $\widehat{CR}_{\{2,1\}}$

Fig. 1.9. A possible sub-partition of the degenerate region CR_2 in Example 1.7 where the regions $\widehat{CR}_{\{2,5\}}$ and $\widehat{CR}_{\{2,4\}}$ and $\widehat{CR}_{\{2,1\}}$ are overlapping. Note that below each picture the feasible set and the slope of the value function in the z-space is depicted when the parameter x is fixed at the point indicated by $\times$.



(a) First region $\widehat{CR}_{\{2,5\}} \subset CR_{\{2\}}$

(b) Second region $\widehat{CR}_{\{2,3\}} \subset CR_{\{2\}}$

Fig. 1.10. A possible solution to Example 1.7 where the regions $\widehat{CR}_{\{2,5\}}$ and $\widehat{CR}_{\{2,3\}}$ are non-overlapping

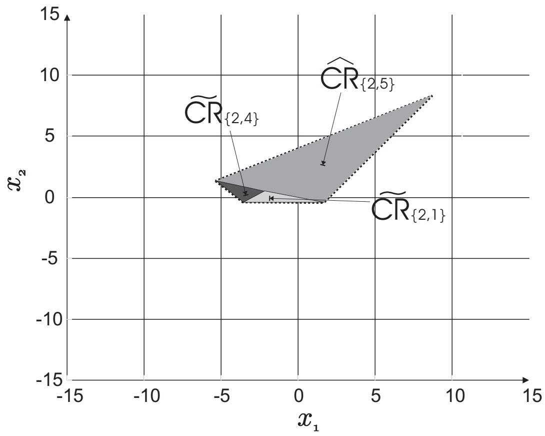


Fig. 1.11. A possible solution for Example 1.7 where $\widetilde{CR}_{\{2,4\}}$ is obtained by intersecting $\widehat{CR}_{\{2,4\}}$ with the complement of $\widehat{CR}_{\{2,5\}}$, and $\widetilde{CR}_{\{2,1\}}$ by intersecting $\widehat{CR}_{\{2,1\}}$ with the complement of $\widehat{CR}_{\{2,5\}}$ and $\widehat{CR}_{\{2,4\}}$

1.4 Multiparametric Quadratic Programming

This section is extracted from the paper [25] and modified to fit the structure of the book. It is included here because many of the subsequent results are based on it.

In this section we investigate multiparametric quadratic programs (mp-QP) of the form:

$$\begin{array}{ll} J^*(x) = \min_z & \left\{J(z,x) = \frac{1}{2}z'Hz\right\} \\ & \text{subj. to } Gz \leq W + Sx, \end{array} \tag{1.34}$$

where $z \in \mathbb{R}^s$ are the optimization variables, $x \in \mathbb{R}^n$ is the vector of parameters, $H \in \mathbb{R}^{s \times s}$, $H \succ 0$, $W \in \mathbb{R}^m$, and $S \in \mathbb{R}^{m \times n}$. Note that the general problem with $J(z,x) = z'Hz + x'Fz$ can always be transformed in the mp-QP (1.34) by using the variable substitution $\tilde{z} \triangleq z + H^{-1}F'x$.

Given a close polyhedral set $K \subset \mathbb{R}^n$ of parameters,

$$K \triangleq \{x \in \mathbb{R}^n : \ Tx \leq Z\}, \tag{1.35}$$

we denote by $K^* \subseteq K$ the region of parameters $x \in K$ such that the QP (1.34) is feasible and the optimum $J^*(x)$ is finite. For any given $\bar{x} \in K^*$, $J^*(\bar{x})$ denotes the minimum value of the objective function in problem (1.34) for $x = \bar{x}$. The function $J^* : K^* \to \mathbb{R}$ will denote the function which expresses the dependence on x of the minimum value of the objective function over K^*, $J^*(\cdot)$ will be called value function. The single-valued function $z^* : K^* \to \mathbb{R}^s$, will describe for any fixed $x \in K^*$ the optimizer $z^*(x)$ related to $J^*(x)$.

We aim at determining the feasible region $K^* \subseteq K$ of parameters x and at finding the expression of the value function $J^*(\cdot)$ and of the optimizer function $z^*(\cdot)$. In particular, we will prove that the optimizer function $z^*(x)$ is a continuous PPWA function of x.

Denote with G_j, S_j, W_j, T_j and Z_j the j-th row of G, S, W, T and Z, respectively. We give the following definition of active and weakly active constraints:

Definition 1.8. *The i-th constraint is active at x if $G_i z^*(x) - W_i - S_i x = 0$, it is inactive if $G_i z^*(x) - W_i - S_i x < 0$. We also define the i-th constraint as weakly active constraint if it is active and its corresponding Lagrange multiplier λ_i is zero.*

Let $J \triangleq \{1, \ldots, m\}$ be the set of constraint indices. For any $A \subseteq J$, let G_A and S_A be the submatrices of G and S, respectively, consisting of the rows indexed by A.

Definition 1.9. *The optimal partition of J associated with x is the partition $(A(x), NA(x))$*

$$\begin{array}{r} A(x) \triangleq \{j \in J : G_j z^*(x) - S_j x = W_j\} \\ NA(x) \triangleq \{j \in J : G_j z^*(x) - S_j x < W_j\} \end{array} \tag{1.36}$$

where $A(x)$ is the *optimal active set* and will be simply referred to as *the set of active constraints at* x.

As in the LP case, the multiparametric analysis uses the concept of *Critical Region* (CR). For a given $x^* \in K^*$ let $(A, NA) \triangleq (A(x^*), NA(x^*))$, and let

$$CR_A \triangleq \{x \in K : A(x) = A\} \tag{1.37}$$

The set CR_A is the critical region related to the set of active constraints A, i.e., the set of all parameters x such that the constraints indexed by A are active at the optimum of problem (1.34).

Definition 1.10. *For a given set of active constraints A we say that* Linear Independence Constraint Qualification*(LICQ) holds if the rows of G_A are linearly independent.*

In the following the set CR_i denotes the critical region related to the set of active constraints A_i.

1.4.1 Geometric Algorithm for mp-QP

As in the mp-LP, the algorithm consists of two main steps, which can be summarized as follows:

1. Determine the dimension $n' \le n$ of the smallest affine subspace $\mathcal{K}$ that contains K^*. If $n' < n$, find the equations in x which define $\mathcal{K}$.
2. Determine the partition of K^* into critical regions CR_i, and find the expression of the functions $J^*(\cdot)$ and $z^*(\cdot)$ for each critical region.

The first preliminary step is identical to the mp-LP case, in the following we detail the second step as appears in [25].

In order to start solving the mp-QP problem, we need an initial vector x_0 inside the polyhedral set K of parameters over which we want to solve the problem, such that the QP problem (1.34) is feasible for $x = x_0$.

A good choice for x_0 is the center of the largest ball contained in K for which a feasible z exists, determined by solving the LP

$$\begin{array}{ll} \max_{x,z,\epsilon} \epsilon & \\ \text{subj. to} & T_i x + \epsilon \|T_i\| \le Z_i, \quad i = 1, \dots, n_T \\ & Gz - Sx \le W \end{array} \tag{1.38}$$

where n_T is the number of rows T_i of the matrix T. In particular, x_0 will be Chebychev center of K when the QP problem (1.34) is feasible for such an x_0. If $\epsilon \le 0$, then the QP problem (1.34) is infeasible for all x in the interior of K. Otherwise, we fix $x = x_0$ and solve the QP problem (1.34), in order to obtain the corresponding optimal solution z_0. Such a solution is unique, because $H \succ 0$, and therefore uniquely determines a set of active constraints A_0 out of the constraints in (1.34). We can then prove the following result

Theorem 1.10. *Let $H \succ 0$. Consider a combination of active constraints A_0, and assume that LICQ holds. Then, the optimal z^* and the associated vector of Lagrange multipliers λ^* are uniquely defined affine functions of x over the critical region CR_0.*

Proof: The first-order Karush-Kuhn-Tucker (KKT) optimality conditions [12] for the mp-QP are given by

$$Hz^* + G'\lambda^* = 0,\ \lambda \in \mathbb{R}^m, \tag{1.39a}$$
$$\lambda_i(G_i z^* - W_i - S_i x) = 0,\ i = 1, \dots, m, \tag{1.39b}$$
$$\lambda^* \geq 0, \tag{1.39c}$$
$$Gz^* \leq W + Sx, \tag{1.39d}$$

We solve (1.39a) for z^*,

$$z^* = -H^{-1}G'\lambda^* \tag{1.40}$$

and substitute the result into (1.39b) to obtain the complementary slackness condition $\lambda^*(\ -G\ H^{-1}\ G'\ \lambda^*\ -\ W\ -\ Sx) = 0$. Let $\lambda^*_{NA_0}$ and $\lambda^*_{A_0}$ denote the Lagrange multipliers corresponding to inactive and active constraints, respectively. For inactive constraints, $\lambda^*_{NA_0} = 0$. For active constraints, $(-G_{A_0}H^{-1}G_{A_0}{}')\lambda^*_{A_0} - W_{A_0} - S_{A_0}x = 0$, and therefore

$$\lambda^*_{A_0} = -(G_{A_0}H^{-1}G_{A_0}{}')^{-1}(W_{A_0} + S_{A_0}x) \tag{1.41}$$

where $G_{A_0}, W_{A_0}, S_{A_0}$ correspond to the set of active constraints A_0, and $(G_{A_0}H^{-1}G_{A_0}{}')^{-1}$ exists because the rows of G_{A_0} are linearly independent. Thus λ^* is an affine function of x. We can substitute $\lambda^*_{A_0}$ from (1.41) into (1.40) to obtain

$$z^* = H^{-1}G_{A_0}{}'(G_{A_0}H^{-1}G_{A_0}{}')^{-1}(W_{A_0} + S_{A_0}x) \tag{1.42}$$

and note that z^* is also an affine function of x. □

Theorem 1.10 characterizes the solution only locally in the neighborhood of a specific x_0, but it does not provide the construction of the set CR_0 where this characterization remains valid. This region can be characterized immediately. The variable z^* from (1.40) must satisfy the constraints in (1.34)

$$GH^{-1}G_{A_0}{}'(G_{A_0}H^{-1}G_{A_0}{}')^{-1}(W_{A_0} + S_{A_0}x) \leq W + Sx \tag{1.43}$$

and by (1.39c) the Lagrange multipliers in (1.41) must remain nonnegative

$$-(G_{A_0}H^{-1}G_{A_0}{}')^{-1}(W_{A_0} + S_{A_0}x) \geq 0 \tag{1.44}$$

as we vary x. After removing the redundant inequalities from (1.43) and (1.44) we obtain a compact representation of CR_0. Obviously, CR_0 is a polyhedron in the x-space, and represents the largest set of $x \in K$ such that the combination of active constraints at the minimizer remains unchanged. Once the critical region CR_0 has been defined, the rest of the space $CR^{\text{rest}} = K - CR_0$ has to be explored and new critical regions generated. □

Remark 1.8. Note that a critical region CR_0 as defined in (1.37) is a set obtained by removing from a polyhedron some (possibly none or all) of it facets. Therefore CR_0 can be neither closed nor opened, while equations (1.43)–(1.44) describe its closure.

Theorem 1.9 in Section 1.3.1 provides a way of partitioning the non-convex set $K \setminus CR_0$ into polyhedral subsets R_i. For each R_i, a new vector x_i is determined by solving the LP (1.38), and, correspondingly, an optimum z_i^*, a set of active constraints A_i, and a critical region CR_i. Theorem 1.9 is then applied to partition $R_i \setminus CR_i$ into polyhedral subsets, and the algorithm proceeds iteratively. The complexity of the algorithm will be discussed in Section 1.4.2.

Note that Theorem 1.9 introduces cuts in the x-space which might split critical regions into subsets. Therefore, after the whole x-space has been covered, those polyhedral regions CR_i are determined where the function $z^*(x)$ is the same. If their union is a convex set, it is computed to permit a more compact description of the solution [21]. Alternatively, in the previous chapter it has been proposed not to intersect (1.43)–(1.44) with the partition generated by Theorem 1.9, and simply use Theorem 1.9 to guide the exploration. As a result, some critical regions may appear more than once. Duplicates can be easily eliminated by recognizing regions where the combination of active constraints is the same. In the sequel, we will denote by N_r the final number of polyhedral cells defining the mp-QP solution (i.e., after the union of neighboring cells or removal of duplicates, respectively).

Degeneracy

So far, we have assumed that the rows of G_{A_0} are linearly independent. It can happen, however, that by solving the QP (1.34) one determines a set of active constraints for which this assumption is violated. For instance, this happens when more than s constraints are active at the optimizer $z_0^* \in \mathbb{R}^s$, i.e., in a case of *primal degeneracy*. In this case the vector of Lagrange multipliers λ_0^* might not be uniquely defined, as the dual problem of (1.34) is not strictly convex. Note that *dual degeneracy* and nonuniqueness of z_0^* cannot occur, as $H \succ 0$. Let $G_{A_0} \in \mathbb{R}^{\ell \times s}$, and let $r = \operatorname{rank} G_{A_0}$, $r < \ell$. In order to characterize such a degenerate situation, consider the QR decomposition $G_{A_0} = \left[\begin{smallmatrix} R_1 \\ 0 \end{smallmatrix}\right] Q$ of G_{A_0}, and rewrite the active constraints in the form

$$R_1 z_0^* = W_1 + S_1 x \tag{1.45a}$$

$$0 = W_2 + S_2 x \tag{1.45b}$$

where $\left[\begin{smallmatrix} S_1 \\ S_2 \end{smallmatrix}\right] = Q^{-1} S_{A_0}$, $\left[\begin{smallmatrix} W_1 \\ W_2 \end{smallmatrix}\right] = Q^{-1} W_{A_0}$. If S_2 is nonzero, because of the equalities (1.45b) CR_0 is a lower dimensional region, which, in general, corresponds to a common boundary between two or more full-dimensional regions. Therefore it is not worth to explore this combination G_{A_0}, S_{A_0}, W_{A_0}. On the other hand, if both W_2 and S_2 are zero the KKT conditions do not lead directly to (1.43)–(1.44), but only to a polyhedron expressed in the (λ, x) space. In this case, a full-dimensional critical region can be obtained by a projection algorithm [67], which, however, is computationally expensive.

In this chapter we suggest a simpler way to handle such a degenerate situation, which consists of collecting r constraints arbitrarily chosen, and proceed with the new reduced set, therefore avoiding the computation of projections. Because of the recursive nature of the algorithm, the remaining other possible subsets of combinations of constraints leading to full-dimensional critical regions will automatically be explored later.

1.4.2 A Summary of the mp-QP Algorithm

Based on the above discussion and results, the main steps of the off-line mp-QP solver are outlined in the following algorithm.

Algorithm 1.4.1

Input: Matrices $H,\ G, W, S$ of Problem (1.34) and set K in (1.35)
Output: Multiparametric solution to Problem (1.34)

1 Let $K \subseteq \mathbb{R}^n$ be the set of parameters (states);
2 execute **partition**(K);
3 **end**.
procedure partition(Y)
4 **let** $x_0 \in Y$ and ϵ the solution to the LP (1.38);
5 **if** $\epsilon \leq 0$ **then exit**; (no full dimensional CR is in Y)
6 Solve the QP (1.34) for $x = x_0$ to obtain (z_0^*, λ_0^*);
7 Determine the set of active constraints A_0 when $z = z_0^*$, $x = x_0$, and build G_{A_0}, W_{A_0}, S_{A_0};
8 If $r = \operatorname{rank} G_{A_0}$ is less than the number ℓ of rows of G_{A_0}, take a subset of r linearly independent rows, and redefine G_{A_0}, W_{A_0}, S_{A_0} accordingly;
9 Determine $\lambda_{A_0}^*(x)$, $z^*(x)$ from (1.41) and (1.42);
10 Characterize the CR from (1.43) and (1.44);
11 Define and partition the rest of the region as in Theorem 1.9;
12 For each new sub-region R_i, **partition**(R_i); **end procedure**.

The algorithm explores the set K of parameters recursively: Partition the rest of the region as in Theorem 1.9 into polyhedral sets R_i, use the same method to partition each set R_i further, and so on. This can be represented as a search tree, with a maximum depth equal to the number of combinations of active constraints (see Sect. 1.4.5 below).

The algorithm solves the mp-QP problem by partitioning the given parameter set K into N_r convex polyhedral regions. Note that the algorithm generates a partition of the state space in the broad sense. The algorithm could be modified to store the critical regions as defined in (1.37) instead of storing its closure. This can be done by keeping track of which facet belongs to a certain critical region and which not. From a practical point of view, such procedure is not necessary since the value function and the optimizer are continuous functions of x.

1.4.3 Propagation of the Set of Active Constraints

We will summarize the main results published in [148] on the properties of the set of active constraints of two neighboring critical regions.

Definition 1.11. *Given a critical region CR_j we say that it is degenerate of the first type if there exists a point $x \in CR_j$ where the LICQ does not hold.*

Definition 1.12. *Given a critical region CR_j we say that it is degenerate of the second type if there exists a point $x \in CR_j$ for which some of the constraints in A_j are weakly active.*

We say that CR_j is a non-degenerate critical region if it is not degenerate of first and second type. We say that the mp-QP (1.34) is non-degenerate if its solutions consists of non-degenerate critical regions.

Theorem 1.11. *Consider two set of active constraints A_j and A_j and let CR_i and CR_j be the corresponding critical region, respectively. Assume that the mp-QP (1.34) is non-degenerate and that $\bar{C}R_i$ and $\bar{C}R_j$ are neighbors, where $\bar{C}R_i$ and $\bar{C}R_j$ denote the closure of the sets CR_i and CR_j, respectively. Then, $A_i \subset A_j$ and $\#A_i = \#A_j - 1$ or $A_j \subset A_i$ and $\#A_i = \#A_j + 1$.*

1.4.4 Continuity, Convexity and $C^{(1)}$ Properties

Convexity of the value function $J^*(x)$ and continuity of the solution $z^*(x)$ easily follow from the results on multiparametric programming summarized in Section 1.2. In the following we give a simple proof based on the linearity result of Theorem 1.10. This fact, together with the convexity of the set of feasible parameters $K^* \subseteq K$ and the piecewise linearity of the solution $z^*(x)$ is proved in the next Theorem.

Theorem 1.12. *Consider the multiparametric quadratic program (1.34) and let $H \succ 0$. Then, the set of feasible parameters $K^* \subseteq K$ is convex. The optimizer $z^*(x) : K^* \to \mathbb{R}^s$ is continuous and piecewise affine on polyhedra, in particular it is affine in each critical region, and the optimal solution $J^*(x) : K^* \to \mathbb{R}$ is continuous, convex and piecewise quadratic on polyhedra.*

Proof: We first prove convexity of K^* and $J^*(x)$. Take a generic x_1, $x_2 \in K^*$, and let $J^*(x_1)$, $J^*(x_2)$ and z_1, z_2 the corresponding optimal values and minimizers. Let $\alpha \in [0,1]$, and define $z_\alpha \triangleq \alpha z_1 + (1-\alpha)z_2$, $x_\alpha \triangleq \alpha x_1 + (1-\alpha)x_2$. By feasibility, z_1, z_2 satisfy the constraints $Gz_1 \leq W + Sx_1$, $Gz_2 \leq W + Sx_2$. These inequalities can be linearly combined to obtain $Gz_\alpha \leq W + Sx_\alpha$, and therefore z_α is feasible for the optimization problem (1.34) where $x(t) = x_\alpha$. This proves that $z(x_\alpha)$ exists, and therefore convexity of $K^* = \bigcup_i CR_i$. In particular, K^* is connected. Moreover, by optimality of $J^*(x_\alpha)$, $J^*(x_\alpha) \leq \frac{1}{2}z_\alpha' H z_\alpha$, and hence $J^*(x_\alpha) - \frac{1}{2}[\alpha z_1' H z_1 + (1-\alpha) z_2' H z_2] \leq \frac{1}{2} z_\alpha' H z_\alpha - \frac{1}{2}[\alpha z_1' H z_1 + (1-\alpha) z_2' H z_2] = \frac{1}{2}[\alpha^2 z_1' H z_1 + (1-\alpha)^2 z_2' H z_2 + 2\alpha(1-\alpha) z_2' H z_1 - \alpha z_1' H z_1 - (1-\alpha) z_2' H z_2] = -\frac{1}{2}\alpha(1-\alpha)(z_1 - z_2)' H (z_1 - z_2) \leq 0$, i.e. $J^*(\alpha x_1 + (1-\alpha)x_2) \leq \alpha J^*(x_1) + (1-\alpha) J^*(x_2)$, $\forall x_1, x_2 \in K$, $\forall \alpha \in [0,1]$, which proves the convexity of $J^*(x)$ on K^*. Within the closed polyhedral regions CR_i in K^* the solution $z^*(x)$ is affine (1.42). The boundary between two regions belongs to both closed regions. Since $H \succ 0$, the optimum is unique, and hence the solution must be continuous across the boundary. The fact that $J^*(x)$ is continuous and piecewise quadratic follows trivially. □

Let $J^*(x)$ be the convex and piecewise quadratic value function in (1.34):

$$J^*(x) = q_i(x) \triangleq x' Q_i x + T_i' x + V_i, \quad \text{if } x \in CR_i,\ i = 1, \ldots, N_r. \tag{1.46}$$

We will prove that if the mp-QP problem (1.34) is not degenerate then $J^*(x)$ is a $C^{(1)}$ function.

Theorem 1.13. *Assume that the mp-QP problem (1.34) is not degenerate. Consider the value function $J^*(x)$ in (1.46) and let CR_i, CR_j be the closure of two neighboring critical regions corresponding to the set of active constraints A_i and A_j, respectively, then*

$$Q_i - Q_j \preceq 0 \text{ or } Q_i - Q_j \succeq 0 \text{ and } Q_i \neq Q_j \tag{1.47}$$

and

$$Q_i - Q_j \preceq 0 \text{ iff } A_i \subset A_j, \tag{1.48}$$

Proof: Let CR_i and CR_j be the closure of two neighboring critical regions and A_i and A_j be the corresponding sets of active constraints at the optimum of QP (1.34). Let $A_i \subset A_j$. We want to prove that the difference between the quadratic terms of $q_i(x)$ and $q_j(x)$ is negative semidefinite, i.e., $Q_i - Q_j \preceq 0$ and that $Q_i \neq Q_j$.

Without loss of generality we can assume that $A_i = \varnothing$. If this is not the case a simple substitution of variables based on the set of active constraints $G_{A_i} z^* = W_{A_i} + S_{A_i} x$ transforms Problem (1.34) into a QP in a lower dimensional space.

For the unconstrained case we have $z^* = 0$ and $J_z^*(x) = 0$. Consequently

$$q_i(x) = 0. \tag{1.49}$$

For the constrained case, from equation (1.42) we obtain

$$q_j(x) = \frac{1}{2}x'(S'_{A_j}\Gamma^{-1}S_{A_j})x + W'_{A_j}\Gamma^{-1}S_{A_j}x + \frac{1}{2}W'_{A_j}\Gamma^{-1}W_{A_j}. \qquad (1.50)$$

where $\Gamma = G_{A_j}H^{-1}\tilde{G}'_{A_j}$, $\Gamma = \Gamma' \succ 0$. The difference of the quadratic terms of $q_i(x)$ and $q_j(x)$ gives

$$Q_i - Q_j = -\frac{1}{2}S'_{A_j}\Gamma^{-1}S_{A_j} \preceq 0. \qquad (1.51)$$

What is left to prove is $Q_i \neq Q_j$. We will prove this by showing that $Q_i = Q_j$ if and only if $CR_i = CR_j$. From (1.43)-(1.44) the polyhedron CR_j where the set of active constraints A_j is constant is defined as

$$CR_j = \{x \mid GH^{-1}G'_{A_j}\Gamma^{-1}(W_{A_j}+S_{A_j}x) \leq W+Sx,\ -\Gamma^{-1}(W_{A_j}+S_{A_j}x) \geq 0\}. \qquad (1.52)$$

From (1.51) we conclude that $Q_i = Q_j$ if and only if $S_{A_j} = 0$. The continuity of $J_z^*(x)$ implies that $q_i(x) - q_j(x) = 0$ on the common facet of CR_i and CR_j. Therefore, by comparing (1.49) and (1.50), we see that $S_{A_j} = 0$ implies $W_{A_j} = 0$. Finally, for $S_{A_j} = 0$ and $W_{A_j} = 0$, from (1.52) it follows that $CR_i = CR_j = \{x \mid 0 \leq W + Sx\}$. □

The following property of convex piecewise quadratic functions was proved in [132]:

Theorem 1.14. *Consider the value function $J^*(x)$ in (1.46) satisfying (1.47) and its quadratic expression $q_i(x)$ and $q_j(x)$ on two neighboring polyhedra CR_i, CR_j then*

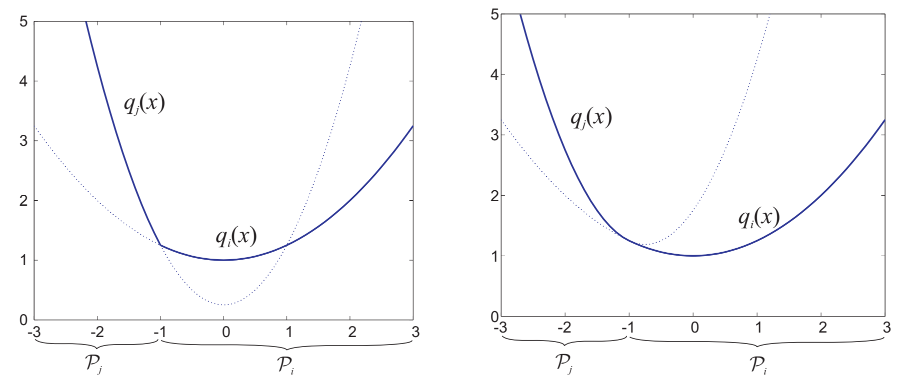
$$q_i(x) = q_j(x) + (a'x - b)(\gamma a'x - \bar{b}), \qquad (1.53)$$

where $\gamma \in \mathbb{R}/\{0\}$.

Equation (1.53) states that the functions $q_i(x)$ and $q_j(x)$ in two neighboring regions CR_i, CR_j of a convex piecewise quadratic function on polyhedra satisfying (1.47) either intersect on two parallel hyperplanes: $a'x - b$ and $\gamma a'x - \bar{b}$ if $\bar{b} \neq \gamma b$ (see Figure 1.12(a)) or are tangent in one hyperplane: $a'x - b$ if $\bar{b} = \gamma b$ (see Figure 1.12(b)). We will prove next that if the mp-QP problem (1.34) is not degenerate then $J^*(x)$ is a $C^{(1)}$ function by showing that the case depicted in Figure 1.12(a) is not consistent with Theorem 1.13. In fact, Figure 1.12(a) depicts the case $Q_i - Q_j \preceq 0$, that implies $A_i \subset A_j$ by Theorem 1.13. However $q_j(0) < q_i(0)$ and from the definitions of q_i and q_j this contradicts the fact that $A_i \subset A_j$.

Theorem 1.15. *Assume that the mp-QP problem (1.34) is not degenerate, then the value function $J^*(x)$ in (1.46) is $C^{(1)}$.*

Proof: We will prove by contradiction that $\bar{b} = \gamma b$. Suppose there exists two neighboring polyhedra CR_i and CR_j such that $\bar{b} \neq \gamma b$. Without loss of



(a) not differentiable one (b) differentiable one

Fig. 1.12. Two piecewise quadratic convex functions

generality assume that (i) $Q_i - Q_j \preceq 0$ and (ii) CR_i is in the halfspace $a'x \leq b$ defined by the common boundary. Let $\bar{\mathcal{F}}_{ij}$ be the common facet between CR_i and CR_j and $\mathcal{F}_{ij}$ its interior.

From (i) and from (1.53), either $\gamma < 0$ or $\gamma = 0$ if $Q_i - Q_j = 0$. Take $x_0 \in \mathcal{F}_{ij}$. For sufficiently small $\varepsilon \geq 0$, the point $x \triangleq x_0 - a\varepsilon$ belongs to CR_i.

Let $J^*(\varepsilon) \triangleq J^*(x_0 - a\varepsilon)$, $q_i(\varepsilon) \triangleq q_i(x_0 - a\varepsilon)$, and consider

$$q_i(\varepsilon) = q_j(\varepsilon) + (a'a\varepsilon)(\gamma a'a\varepsilon + (\bar{b} - \gamma b)). \tag{1.54}$$

From convexity of $J^*(\varepsilon)$, $J^{*-}(\varepsilon) \leq J^{*+}(\varepsilon)$ where $J^{*-}(\varepsilon)$ and $J^{*+}(\varepsilon)$ are the left and right derivatives of $J^*(\varepsilon)$ with respect to ε. This implies $q_j'(\varepsilon) \leq q_i'(\varepsilon)$ where $q_j'(\varepsilon)$ and $q_i'(\varepsilon)$ are the derivatives of $q_j(\varepsilon)$ and $q_i(\varepsilon)$, respectively. Condition $q_j'(\varepsilon) \leq q_i'(\varepsilon)$ is true if and only if $-(\bar{b} - \gamma b)) \leq 2\gamma(a'a)\varepsilon$, that implies $-(\bar{b} - \gamma b) < 0$ since $\gamma < 0$ and $\varepsilon > 0$.

From (1.54) $q_j(\varepsilon) < q_i(\varepsilon)$ for all $\varepsilon \in (0, \frac{-(\bar{b}-\gamma b)}{\gamma a'a})$.

Thus there exists $x \in CR_i$ with $q_j(x) < q_i(x)$. This is a contradiction since from Theorem 1.13, $A_i \subset A_j$. □

Note that in case of degeneracy the value function $J^*(x)$ in (1.46) may not be $C^{(1)}$. The following counterexample was given in [28].

Example 1.8. Consider the mp-QP (1.34) with

$$H = \begin{bmatrix} 3 & 3 & -1 \\ 3 & 11 & 23 \\ -1 & 23 & 75 \end{bmatrix}$$
$$G = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \\ -1 & -1 & -1 \\ -1 & -3 & -5 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \; W = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \; S = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.55}$$

and $K = \{x \in \mathbb{R} | 1 \leq x \leq 5\}$.

The problem was solved by using algorithm 1.4.1. The solution consists of 5 critical regions. The critical regions and the expression of the value function are reported in Table 1.4. The reader can verify that the value function is not continuously differentiable at $x = 3$. Indeed, at $x = 3$ the LICQ condition does not hold and therefore, the hypothesis of Theorem 1.15 is not fulfilled.

Region	Optimal value
$CR_{\{1,2,3,4,6\}} = \{x \| 1 \leq x < 1.5\}$	$2.5x^2 - 6x + 5$
$CR_{\{1,2,3,4\}} = \{x \| 1.5 \leq x \leq 2\}$	$0.5x^2 + 0.5$
$CR_{\{1,2,3,4,7\}} = \{x \| 2 < x < 3\}$	$x^2 - 2x + 2.5$
$CR_{\{1,2,3,4,5,7\}} = \{x \| x = 3\}$	$x^2 - 2x + 2.5$
$CR_{\{1,2,3,4,5\}} = \{x \| 3 < x \leq 5\}$	$5x^2 - 24x + 32.5$

Table 1.4. Critical regions and value function corresponding to the solution of Example 1.8

1.4.5 Complexity Analysis

The number N_r of regions in the mp-QP solution depends on the dimension n of the parameter, and on the number of degrees of freedom s and constraints m in the optimization problem (1.34). As the number of combinations of ℓ constraints out of a set of m is $\binom{m}{\ell} = \frac{m!}{(m-\ell)!\ell!}$, the number of possible combinations of active constraints at the solution of a QP is at most $\sum_{\ell=0}^{m} \binom{m}{\ell} = 2^m$.

Let N_n be the number of different regions that are generated by the mp-QP algorithm 1.4.1, $N_n \geq N_r$. A worst-case estimate of N_n can be computed from the way Algorithm 1.4.1 generates critical regions. The first critical region CR_0 is defined by the constraints $\lambda(x) \geq 0$ (m constraints) and $Gz(x) \leq W + Sx$ (m constraints). If the strict complementary slackness condition holds, only m constraints can be active, and hence every CR is defined by m constraints. From Theorem 1.9, CR^{rest} consists of m convex polyhedra R_i, defined by at most m inequalities. For each R_i, a new CR_i is determined which consists of $2m$ inequalities (the additional m inequalities come from the condition $CR_i \subseteq R_i$), and therefore the corresponding CR^{rest} partition includes $2m$ sets defined by $2m$ inequalities. As mentioned above, this way of generating regions can be associated with a search tree. By induction, it is easy to prove that at the tree level $k+1$ there are $k!m^k$ regions defined by $(k+1)m$ constraints. As observed earlier, each CR is the largest set corresponding to a certain combination of active constraints. Therefore, the search tree has a maximum depth of 2^m, as at each level there is one admissible combination less. In conclusion, the number of regions N_n explored in the solution to the mp-QP problem is $N_n \leq \sum_{k=0}^{2^m-1} k!m^k$, each one defined by at most $m2^m$ linear inequalities. Note that the above analysis is largely overestimating the complexity, as it does not take into account (*i*) the elimination of redundant constraints when a CR is generated, and (*ii*) that possible empty sets are not partitioned further.

1.4.6 Other Algorithms for solving mp-QP

The solution of multiparametric quadratic programs was also addressed by Seron, DeDoná and Goodwin in [135, 56] in parallel with the study of Bemporad and coauthors in [25]. The method in [135, 56] was proposed initially for the special class of mp-QPs deriving from optimal control of linear systems with input constraints only, but it can be easily extended to the more general mp-QP fromulation (1.34). More recently, new mp-QP solvers have been proposed by Tondel, Johansen and Bemporad in [147] and by Baotic in [10].

All these algorithms are based on an iterative procedure that builds up the parametric solution by generating new polyhedral regions of the parameter space at each step. The methods differ in the way they explore the parameter space, that is, the way they identify active constraints corresponding to the critical regions neighboring to a given critical region.

In [135, 56] the authors construct the unconstrained critical region and then generate neighboring critical regions by enumerating all possible combinations of active constraints.

In [147] the authors explore the parameter space outside a given region CR_i by examining its set of active constraints A_i. The critical regions neighboring to CR_i are constructed by elementary operations on the active constraints set A_i that can be seen as an equivalent "pivot" for the quadratic program. For this reason the method can be considered as an extension of the method of Gal [68] to multiparametric quadratic programming.

In [10] the author uses a direct exploration of the parameter space as in [25] but he avoids the partition of the state space described in Theorem 1.9. Given a polyhedral critical region CR_i, the procedure goes through all its facets and generates the center of each facet. For each facet $\mathcal{F}_i$ a new parameter x^i_ε is generated, by moving from the center of the facet in the direction of the normal to the facet by a small step. If such parameter x^i_ε is infeasible or is contained in a critical region already stored, then the exploration in the direction of $\mathcal{F}_i$

stops. Otherwise, the set of active constraints corresponding to the critical region sharing the facet $\mathcal{F}_i$ with the region CR_i is found by solving a QP for the new parameter x^i_ε.

In our experience the solvers in [10, 147] are the most efficient.

1.5 Multiparametric Mixed-Integer Linear Programming

Consider the mp-LP

$$\begin{aligned} J^*(x) = \min_z \ & \{J(z,x) = c'z\} \\ \text{subj. to } & Gz \leq W + Sx. \end{aligned} \tag{1.56}$$

where z is the optimization variable, $x \in \mathbb{R}^n$ is the vector of parameters, $G' = [G_1{}' \ldots G_m{}']$ and $G_j \in \mathbb{R}^n$ denotes the j-th row of G, $c \in \mathbb{R}^s$, $W \in \mathbb{R}^m$, and $S \in \mathbb{R}^{m \times n}$. When we restrict some of the optimization variables to be 0 or 1, $z \triangleq \{z_c, z_d\}$, $z_c \in \mathbb{R}^{s_c}$, $z_d \in \{0,1\}^{s_d}$ and $s \triangleq s_c + s_d$, we refer to (1.8) as a (right-hand-side) *multiparametric mixed-integer linear program* (mp-MILP).

1.5.1 Geometric Algorithm for mp-MILP

Consider the mp-MILP (1.56). Given a close polyhedral set $K \subset \mathbb{R}^n$ of parameters,

$$K \triangleq \{x \in \mathbb{R}^n : \ Tx \leq Z\} \tag{1.57}$$

we denote by $K^* \subseteq K$ the region of parameters $x \in K$ such that the MILP (1.56) is feasible and the optimum $J^*(x)$ is finite. For any given $\bar{x} \in K^*$, $J^*(\bar{x})$ denotes the minimum value of the objective function in problem (1.56) for $x = \bar{x}$. The function $J^* : K^* \to \mathbb{R}$ will denote the function which expresses the dependence on x of the minimum value of the objective function over K^*, $J^*(\cdot)$ will be called value function. The set-valued function $Z^* : K^* \to 2^{\mathbb{R}^{s_c}} \times 2^{\{0,1\}^{s_d}}$ will describe for any fixed $x \in K^*$ the set of optimizers $z^*(x)$ related to $J^*(x)$.

We aim at determining the region $K^* \subseteq K$ of feasible parameters x and at finding the expression of the value function $J^*(x)$ and the expression an optimizer function $z^*(x) \in Z^*(x)$.

Remark 1.9. If the parameters x are restricted to be only integer values, we can solve this problem by embedding it into problem (1.56) with x continuous.

Two main approaches have been proposed for solving mp-MILP problems. In [1], the authors develop an algorithm based on branch and bound (B&B) methods. At each node of the B&B tree an mp-LP is solved. The solution at the root node represents a valid lower bound, while the solution at a node where all the integer variables have been fixed represents a valid upper bound. As in standard B&B methods, the complete enumeration of combinations of 0-1 integer variables is avoided by comparing the multiparametric solutions, and by fathoming the nodes where there is no improvement of the value function.

In [57] an alternative algorithm was proposed, which will be detailed in this section. Problem (1.56) is alternatively decomposed into an mp-LP and an MILP subproblem. When the values of the binary variable are fixed, an mp-LP is solved, and its solution provides a parametric upper bound to the value function $J^*(x)$. When the parameters in x are treated as free variables, an

MILP is solved, that provides a new integer vector. The algorithm is composed of an *initialization step*, and a recursion between the solution of an *mp-LP subproblem* and an *MILP subproblem*.

Initialization

Solve the following MILP problem

$$\begin{array}{ll} \min\limits_{\{z,x\}} & c'z \\ \text{subj. to} & Gz - Sx \leq W \\ & x \in K \end{array} \tag{1.58}$$

where x is treated as an independent variable. If the MILP (1.58) is infeasible then the mp-MILP (1.56) admits no solution, i.e. $K^* = \emptyset$; otherwise its solution z^* provides a feasible integer variable $\bar{z}_d$.

mp-LP subproblem

At a generic step of the algorithm we have a polyhedral partition of the initial set of parameters K. For each polyhedron of the partition we know if

1. the MILP (1.58) is infeasible for all x belonging to the polyhedron.
2. the MILP (1.58) is feasible for all x belonging to the polyhedron and we have a current upper bound on the affine value function $J^*(x)$ (in the polyhedron) and an integer variable that improves the bound at least at a point x of the polyhedron,
3. the MILP (1.58) is feasible for all x belonging to the polyhedron and we know the optimal affine value function $J^*(x)$ inside the polyhedron.

Obviously the algorithm will continue to iterate only on the polyhedra corresponding to point 2. above (if there is no such a polyhedron then the algorithm ends) and in particular, at step j we assume to have stored

1. A list of N_j polyhedral regions CR_i and for each of them an associated parametric affine upper bound $\bar{J}_i(x)$ ($\bar{J}_i(x) = +\infty$ if no integer solution has been found yet in CR_i).
2. For each CR_i a set of integer variables $Z_i = \bar{z}_{d_i}^0, \ldots, \bar{z}_{d_i}^{Nb_i}$, that have already been explored in the region CR_i.
3. For each CR_i an integer feasible variable $\bar{z}_{d_i}^{Nb_i+1} \notin Z_i$ such that there exists z_c and $\hat{x} \in CR_i$ for which $Gz \leq W + S\hat{x}$ and $c'z < \bar{J}_i(\hat{x})$ where $z = \{z_c, \bar{z}_{d_i}^{Nb_i+1}\}$. That is, $\bar{z}_{d_i}^{Nb_i+1}$ is an integer variable that improves the current bound for at least one point of the current polyhedron.

At step $j = 0$, set: $N_0 = 1$, $CR_1 = K$, $Z_1 = \emptyset$, $\bar{J}_1 = +\infty$, $\bar{z}_{d_1}^1 = \bar{z}_d$.

For each CR_i we solve the following mp-LP problem

$$
\begin{aligned}
\tilde{J}_i(x) = \quad & \min_z c'z \\
& \text{subj. to } Gz \leq W + Sx \\
& \qquad z_d = \bar{z}_{d_i}^{Nb_i+1} \\
& \qquad x \in CR_i
\end{aligned}
\tag{1.59}
$$

By Theorem 1.8, the solution of mp-LP (1.59) provides a partition of CR_i into polyhedral regions $R_i^k,\ k = 1, \dots, N_{R_i}$ and and a PWA value function

$$
\tilde{J}_i(x) = (\tilde{JR}_i^k(x) \triangleq c_i^k x + p_i^k) \text{ if } x \in R_i^k,\ k = 1, \dots, N_{R_i} \tag{1.60}
$$

where $\tilde{JR}_i^j(x) = +\infty$ in R_i^j if the integer variable $\bar{z}_{d_i}$ is not feasible in R_i^j and a PWA continuous control law $z^*(x)$ ($z^*(x)$ is not defined in R_i^j if $\tilde{JR}_i^j(x) = +\infty$).

The function $\tilde{J}_i(x)$ will be an upperbound of $J^*(x)$ for all $x \in CR_i$. Such bound $\tilde{J}_i(x)$ on the value function have to be compared with the current bound $\bar{J}_i(x)$ in CR_i in order to obtain the lowest of the two parametric value functions and to update the bound.

While updating $\bar{J}_i(x)$ three cases are possible:

1. $\bar{J}_i(x) = \tilde{JR}_i^k(x)\ \forall x \in R_i^k$ if $(\tilde{JR}_i^k(x) \leq \bar{J}_i(x)\ \forall\ x \in R_i^k)$
2. $\bar{J}_i(x) = \bar{J}_i(x)\ \forall x \in R_i^k$ (if $\tilde{JR}_i^k(x) \geq \bar{J}_i(x)\ \forall\ x \in R_i^k$)
3. $\bar{J}_i(x) = \begin{cases} \bar{J}_i(x) & \forall x \in (R_i^k)_1 \triangleq \{x \in R_i^k | \tilde{JR}_i^k(x) \geq \bar{J}_i(x)\} \\ \tilde{JR}_i^k(x) & \forall x \in (R_i^k)_2 \triangleq \{x \in R_i^k | \tilde{JR}_i^k(x) \leq \bar{J}_i(x)\} \end{cases}$

The three cases above can be distinguished by using a simple linear program. In the third case, the region R_i^k is partitioned into two regions $(R_i^k)_1$ and $(R_i^k)_2$ that are convex polyhedra since $\tilde{JR}_i^k(x)$ and $\bar{J}_i(x)$ are affine functions of x.

After the mp-LP (1.59) has been solved for all $i = 1, \dots, N_j$ (the subindex j denotes the that we are at step j of the recursion) and the value function has been updated, each initial region CR_i has been subdivided into at most $2N_{R_i}$ polyhedral regions R_i^k and possibly $(R_i^k)_1$ and $(R_i^k)_2$ with a corresponding updated parametric bound on the value function $\bar{J}_i(x)$. For each R_i^k, $(R_i^k)_1$ and $(R_i^k)_2$ we define the set of integer variables already explored as $Z_i = Z_i \bigcup \bar{z}_{d_i}^{Nb_i+1}$, $Nb_i = Nb_i + 1$. In the sequel the polyhedra of the new partition will be referred to as CR_i.

MILP subproblem

At step j for each critical region CR_i (note that this CR_i are the output of the previous phase) we solve the following MILP problem.

$$\min_{\{z,x\}} \quad c'z \tag{1.61}$$
$$\text{subj. to} \quad Gz - Sx \leq W \tag{1.62}$$
$$c'z \leq \bar{J}_i(z) \tag{1.63}$$
$$z_d \neq \bar{z}^k_{d_i},\ k = 1, \ldots, Nb_i \tag{1.64}$$
$$x \in CR_i \tag{1.65}$$

where constraints (1.64) prohibits integer solutions that have been already analyzed in CR_i from appearing again and constraints (1.63) excludes integer solutions with higher values than the current upper bound. If problem (1.65) is infeasible then the region CR_i is excluded from further recursion and the current upper bound represents the final solution. If problem (1.65) is feasible, then the discrete optimal component $z^*_{d_i}$ is stored and represents a feasible integer variable that is optimal at least in one point of CR_i.

Recursion

For all the region CR_i not excluded from the MILP's subproblem (1.61)-(1.65) the algorithm continues to iterate between the mp-LP (1.59) with $\bar{z}^{Nb_i+1}_{d_i} = z^*_{d_i}$ and the MILP (1.61)-(1.65). The algorithm terminates when all the MILP's (1.61)-(1.65) are infeasible.

Note that the algorithm generates a partition of the state space in the broad sense. Some parameter x could belong to the boundary of several regions. Differently form the LP and QP case, the value function may be discontinuous and therefore such case has to be treated carefully. If a point x belong to different critical regions, the expressions of the value function associated to such the regions have to be compared in order to assign to x the right optimizer. Such procedure can be avoided by keeping track which facet belongs to a certain critical region and which not. Moreover, if the value function associated to the regions containing the same parameter x coincide this implies the presence of multiple optimizers.

1.5.2 Theoretical Results

The following properties of $J^*(x)$ and $Z^*(x)$ easily follow form the algorithm described above.

Theorem 1.16. *Consider the mp-MILP (1.56). Then, the set K^* is a (possibly non-convex) polyhedral set and the value function $J^*(\cdot)$ is piecewise affine on polyhedra. If the optimizer $z^*(x)$ is unique for all $x \in K^*$, then the optimizer function $z^*_c : K^* \to \mathbb{R}^{s_c}$, $z^*_d : K^* \to \{0,1\}^{s_d}$ is piecewise affine on polyhedra. Otherwise, it is always possible to define a piecewise affine optimizer function $z^*(x) \in Z^*(\theta)$ for all $x \in K^*$.*

Note that differently from the mp-LP, the set K^* can be disconnected and non-convex.

1.6 Multiparametric Mixed-Integer Quadratic Programming

Consider the mp-QP

$$
\begin{aligned}
J^*(x) = \min_{z} \quad & J(z,x) = z'H_1 z + z'H_2 x + xH_3 x + c_1 z + c_2 x \\
& \text{subj. to } Gz \leq W + Sx,
\end{aligned} \tag{1.66}
$$

When we restrict some of the optimization variables to be 0 or 1, $z \triangleq [z_c, z_d]$, where $z_c \in \mathbb{R}^{s_c}$, $z_d \in \{0,1\}^{s_d}$, we refer to (1.66) as a (right-hand-side) *multi-parametric mixed-integer quadratic program* (mp-MIQP). For a given polyhedral set $K \subseteq \mathbb{R}^n$ of parameters,

$$
K \triangleq \{x \in \mathbb{R}^n : \ Tx \leq Z\} \tag{1.67}
$$

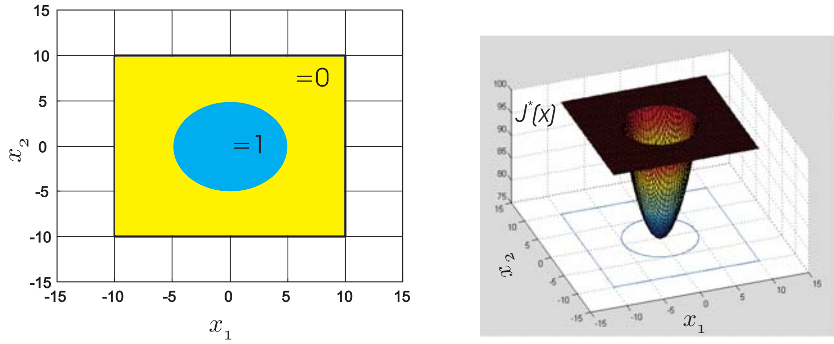
we denote by $K^* \subseteq K$ the region of parameters $x \in K$ such that the MIQP (1.66) is feasible and the optimum $J^*(x)$ is finite. For any given $\bar{x} \in K^*$, $J^*(\bar{x})$ denotes the minimum value of the objective function in problem (1.66) for $x = \bar{x}$. The function $J^* : K^* \to \mathbb{R}$ will denote the function which expresses the dependence on x of the minimum value of the objective function over K^*, $J^*(\cdot)$ will be called value function. The set-valued function $Z^* : K^* \to 2^{\mathbb{R}^{s_c}} \times 2^{\{0,1\}^{s_d}}$ will describe for any fixed $x \in K^*$ the set of optimizers $z^*(x)$ related to $J^*(x)$.

We aim at determining the region $K^* \subseteq K$ of feasible parameters x and at finding the expression of the value function $J^*(x)$ and the expression an optimizer function $z^*(x) \in Z^*(x)$.

We show with a simple example that the geometric approach of this chapter cannot be used for solving mp-MIQP's. Suppose u_1, u_2, x_1, $x_2 \in \mathbb{R}$ and $\delta \in \{0,1\}$, then the following mp-MIQP

$$
J^*(x_1,x_2) = \min_{u_1,u_2,\delta} \ u_1^2 + u_2^2 - 25\delta + 100
$$

$$
\text{subj. to} \quad \begin{bmatrix} 1 & 0 & 10 \\ -1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & -1 & 10 \\ 1 & 0 & -10 \\ -1 & 0 & -10 \\ 0 & 1 & -10 \\ 0 & -1 & -10 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \delta \end{bmatrix} \leq \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \tag{1.68}
$$



(a) Critical Regions (b) Value function

Fig. 1.13. Solution to the mp-MIQP (1.68)

can be simply solved by noting that for $\delta = 1$ $u_1 = x_1$ and $u_2 = x_2$ while for $\delta = 0$ $u_1 = u_2 = 0$. By comparing the value functions associated to $\delta = 0$ and $\delta = 1$ we obtain two critical regions

$$\begin{array}{l} CR_1 = \{x_1, x_2 \in \mathbb{R} |\ x_1^2 + x_2^2 \leq 75\} \\ CR_2 = \{x_1, x_2 \in \mathbb{R} |\ -10 \leq x_1 \leq 10,\ \ -10 \leq x_2 \leq 10,\ x_1^2 + x_2^2 > 75\} \end{array} \tag{1.69}$$

with the associated parametric optimizer

$$\begin{array}{l} u_1^*(x_1, x_2) = \begin{cases} x_1 & \text{if } [x_1, x_2] \in CR_1 \\ 0 & \text{if } [x_1, x_2] \in CR_2 \end{cases} \\ u_2^*(x_1, x_2) = \begin{cases} x_2 & \text{if } [x_1, x_2] \in CR_1 \\ 0 & \text{if } [x_1, x_2] \in CR_2 \end{cases} \end{array} \tag{1.70}$$

and parametric value function

$$J^*(x_1, x_2) = \begin{cases} x_1^2 + x_2^2 + 75 & \text{if } [x_1, x_2] \in CR_1 \\ 100 & \text{if } [x_1, x_2] \in CR_2 \end{cases} \tag{1.71}$$

The two critical regions and the value function are depicted in Figure 1.13.

This example demonstrate that, in general, the critical regions of an mp-MIQP cannot be decomposed into convex polyhedra. Therefore the way of partitioning the rest of the space presented in Theorem 1.9 cannot be applied here.

To the authors' knowledge, there does not exist an efficient method for solving general mp-MIQPs. In Chapter 8 we will present an algorithm that efficiently solves mp-MIQPs derived from optimal control problems for discrete time hybrid systems.

Part II

Optimal Control of Linear Systems

2

Constrained Finite Time Optimal Control

For discrete time linear systems we prove that the solution to constrained finite time optimal control problems is a time varying piecewise affine feedback control law. We give insight into the structure of the optimal control law and of the value function for optimal control problems with performance criteria based on quadratic and linear norms. We describe how the optimal control law can be efficiently computed by means of multiparametric linear and quadratic programming, for linear and quadratic performance criteria respectively.

2.1 Problem Formulation

Consider the linear time-invariant system

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ \quad y(t) = Cx(t) \end{cases} \tag{2.1}$$

subject to the constraints

$$Ex(t) + Lu(t) \leq M \tag{2.2}$$

at all time instants $t \geq 0$. [1]

In (2.1)–(2.2), $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, input, and output vector respectively. In (2.2), $Ex(t)+Lu(t) \leq M$ are the constraints on input and states.

Define the following cost function

$$J(U_N, x(0)) \triangleq \|Px_N\|_p + \sum_{k=0}^{N-1} \|Qx_k\|_p + \|Ru_k\|_p \tag{2.3}$$

where x_k denotes the state vector at time k obtained by starting from the state $x_0 = x(0)$ and applying to system (2.1) the input sequence $u_0, \ldots, u_{k-1}$. Consider the constrained finite time optimal control problem (CFTOC)

$$\begin{array}{rll} J^*(x(0)) = & \min_{U_N} & J(U_N, x(0)) \\ & \text{subj. to} & Ex_k + Lu_k \leq M,\ k = 0, \ldots, N-1 \\ & & x_N \in \mathcal{X}_f \\ & & x_{k+1} = Ax_k + Bu_k,\ k \geq 0 \\ & & x_0 = x(0) \end{array} \tag{2.4}$$

where N is the time horizon and $\mathcal{X}_f \subseteq \mathbb{R}^n$ is a terminal polyhedral region. In (2.3)–(2.4) we denote with $U_N \triangleq [u_0', \ldots, u_{N-1}']' \in \mathbb{R}^s$, $s \triangleq mN$ the optimization vector, with $\|Qx\|_p$ the p-norm of the vector x weighted with the

[1] The results of this chapter also hold for more general form of linear constraints $E_0x(0) + \cdots + E_Nx(N) + L_0u(0) + \cdots + L_Nu(N) \leq M$, arising, for example, from constraints on the input rate $\Delta u(t) \triangleq u(t) - u(t-1)$.

matrix Q, $p = 1, 2, \infty$. We denote with $\mathcal{X}_j$ the set of states x_j at time j for which (2.3)–(2.4) is feasible, i.e.,

$$\begin{aligned} \mathcal{X}_j &= \{x \in \mathbb{R}^n |\ \exists u\ \ (Ex + Lu \leq M,\ \text{and}\ Ax + Bu \in \mathcal{X}_{j+1}\ )\}, \\ &\quad j = 0, \ldots, N-1 && (2.5) \\ \mathcal{X}_N &= \mathcal{X}_f. && (2.6) \end{aligned}$$

In the following, we will assume that $Q = Q' \succeq 0$, $R = R' \succ 0$, $P \succeq 0$, for $p = 2$, and that Q, R, P are full column rank matrices for $p = 1, \infty$. We will also denote with $\mathcal{X}_0 \subseteq \mathbb{R}^n$ the set of initial states $x(0)$ for which the optimal control problem (2.4) is feasible.

Note that in the whole book we will distinguish between the *current* state $x(k)$ of system (2.1) at time k and the variable x_k in the optimization problem (2.4), that is the *predicted* state of system (2.1) at time k obtained by starting from the state $x_0 = x(0)$ and applying to system (2.1) the input sequence $u_0, \ldots, u_{k-1}$. Analogously, $u(k)$ is the input applied to system (2.1) at time k while u_k is the k-th optimization variable of the optimization problem (2.4).

If we set

$$p = 2,\ \{(x, u) \in \mathbb{R}^{n+m} | Ex + Lu \leq M\} = \mathbb{R}^{n+m},\ \mathcal{X}_f = \mathbb{R}^n, \qquad (2.7)$$

problem (2.4) becomes the standard unconstrained finite time optimal control problem whose solution can be expressed through the time varying state feedback control law

$$u^*(k) = K_k x(k) \quad k = 0, \ldots, N-1 \qquad (2.8)$$

where the gain matrices K_k are given by the equation

$$K_k = -(B' P_{k+1} B + R)^{-1} B' P_{k+1} A, \qquad (2.9)$$

and where the symmetric positive semidefinite matrices P_k are given recursively by the algorithm

$$\begin{aligned} P_N &= P, && (2.10) \\ P_k &= A'(P_{k+1} - P_{k+1} B (B' P_{k+1} B + R)^{-1} B P_{k+1}) A + Q. && (2.11) \end{aligned}$$

The optimal cost is given by

$$J^*(x(0)) = x(0)' P_0 x(0). \qquad (2.12)$$

If in addition to (2.7) we set $N = +\infty$, and assume that the pair (A, B) is stabilizable and the pair (D, A) is detectable (where $Q = D'D$), then problem (2.4)–(2.7) becomes the standard infinite time linear quadratic regulator (LQR) problem whose solution can be expressed as the state feedback control law

$$u^*(k) = Kx(k), \quad k = 0, \ldots, +\infty \tag{2.13}$$

where the gain matrix K is given by

$$K = -(B'PB + R)^{-1}B'P_\infty A \tag{2.14}$$

and where P_∞ is the unique solution of the algebraic matrix equation

$$P_\infty = A'(P_\infty - P_\infty B(B'P_\infty B + R)^{-1}BP_\infty)A + Q \tag{2.15}$$

within the class of positive semidefinite symmetric matrices.

In the following chapters we will show that the solution to problem (2.4) can again be expressed in feedback form where $u^*(k)$ is a continuous piecewise affine function on polyhedra of the state $x(k)$, i.e., $u^*(k) = f_k(x(k))$ where

$$f_k(x) = F_k^i x + g_k^i \quad \text{if} \quad H_k^i x \leq K_k^i, \; i = 1, \ldots, N_k^r. \tag{2.16}$$

H_k^i and K_k^i in equation (2.16) are the matrices describing the i-th polyhedron $CR_k^i = \{x \in \mathbb{R}^n | H_k^i x \leq K_k^i\}$ inside which the feedback optimal control law $u^*(k)$ at time k has the affine form $F_k^i x + g_k^i$. The polyhedra $CR_k^i, \; i = 1, \ldots, N_k^r$ are a partition in the broad sense sense of the set of feasible states of problem (2.4) at time k. Since the functions $f_k(x(k))$ are continuous the use of polyhedral partition in the broad sense will not cause any problem, on the contrary it will help keeping the exposition lighter and more clear.

We will give insight into the structure of the value function and describe how the optimal control law can be efficiently computed by means of multi-parametric linear and quadratic programming. We will distinguish the cases $p = 1, \infty$ and $p = 2$.

2.2 State Feedback Solution of CFTOC, 2-Norm Case

By substituting

$$x_k = A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \tag{2.17}$$

in (2.4), it can be rewritten in the form

$$\begin{aligned} J^*(x(0)) = \tfrac{1}{2}x'(0)Yx(0) + \min_{U_N} \quad & \tfrac{1}{2}U_N{}'HU_N + x'(0)FU_N \\ \text{subj. to } & GU_N \leq W + Ex(0) \end{aligned} \tag{2.18}$$

where $H = H' \succ 0$, H, F, Y, G, W, E are easily obtained from P, Q, R, (2.4) and (2.17), and it follows from the previous assumptions that $\left[\begin{smallmatrix} Y & F' \\ F & H \end{smallmatrix}\right] \succeq 0$. Note that the optimizer U_N is independent of the term involving Y in (2.18).

We view $x(0)$ as a vector of parameters and our goal is to solve (2.18) for all values of $x(0)$ of interest, and to make this dependence *explicit*. Note that the set $\mathcal{X}_0$ is a polyhedron and can be computed by projecting the poyhedron $\mathcal{P}_0 = \{(U_N, x(0)) \in \mathbb{R}^{s+n} |\ GU_N \leq W + Ex(0)\}$ on the $x(0)$ space.

Before proceeding further, it is convenient to define

$$z \triangleq U_N + H^{-1}F'x(0), \tag{2.19}$$

$z \in \mathbb{R}^s$, and to transform (2.18) by completing squares to obtain the equivalent problem

$$\begin{aligned} {J^*}_z(x(0)) = \min_z \quad & \tfrac{1}{2} z'Hz \\ \text{subj. to } & Gz \leq W + Sx(0), \end{aligned} \tag{2.20}$$

where $S \triangleq E + GH^{-1}F'$, and ${J^*}_z(x(0)) = J^*(x(0)) - \frac{1}{2}x(0)'(Y - FH^{-1}F')x(0)$. In the transformed problem the parameter vector $x(0)$ appears only on the rhs of the constraints.

Problem (2.20) is a multiparametric quadratic program that can be solved by using the algorithm described in Section 1.4. Once the multiparametric problem (2.20) has been solved for a polyhedral set $X \subset \mathbb{R}^n$, the solution ${U_N}^* = {U_N}^*(x(0))$ of CFTOC (2.4) and therefore $u^*(0) = u^*(x(0))$ is available explicitly as a function of the initial state $x(0)$.

Theorem 1.12 states that the solution $z^*(x(0))$ of the mp-QP problem (2.20) is a continuous and piecewise affine function on polyhedra of x. Clearly the same properties are inherited by the controller. The following Corollaries of Theorem 1.12 establish the analytical properties of the optimal control law and of the value function.

Corollary 2.1. *The control law $u^*(0) = f_0(x(0))$, $f_0 : \mathbb{R}^n \to \mathbb{R}^m$, obtained as a solution of the optimization problem (2.4) is continuous and piecewise affine on polyhedra*

$$f_0(x) = F_0^i x + g_0^i \quad \text{if} \quad x \in CR_0^i, \quad i = 1, \ldots, N_0^r \tag{2.21}$$

where the polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n | H_0^i x \leq K_0^i\}$, $i = 1, \ldots, N_0^r$ are a partition in the broad sense of the feasible polyhedron $\mathcal{X}_0$

Proof: By (2.19), ${U_N}^*(x) = z^*(x(0)) - H^{-1}F'x(0)$ is an affine function of x in each region $CR_0^i = \{x :\ H_0^i x \leq K_0^i,\ i = 1, \ldots, N_0^r\}$, and therefore its first component $u^*(0)$ is affine on CR_0^i. Also, u is a combination of continuous functions and is therefore continuous. □

Remark 2.1. Note that as discussed in Remark 1.8, the critical regions defined in (1.37) are in general sets that are neither can be neither closed nor opened. In Corollary 2.1 the polyhedron CR_0^i describe the closure of a critical region. As already mentioned in the introduction of this chapter, the function $f_0(x)$ is continuous and therefore the use of polyhedral partition in the broad sense will help keeping the exposition lighter and more clear.

Corollary 2.2. *The value function $J^*(x(0))$ obtained as solution of the optimization problem (2.4) is convex and piecewise quadratic on polyhedra. Moreover, if the mp-QP problem (2.20) is not degenerate, then the value function $J^*(x(0))$ is $C^{(1)}$.*

Proof: By Theorem 1.12 ${J^*}_z(x(0))$ is a a convex function of $x(0)$. As $\left[\begin{smallmatrix} Y & F' \\ F & H \end{smallmatrix}\right] \succeq 0$, its Schur complement $Y - FH^{-1}F' \succeq 0$, and therefore $J^*(x(0)) = {J^*}_z(x(0)) + \frac{1}{2}x(0)'(Y - FH^{-1}F')x(0)$ is a convex function, because it is the sum of convex functions. If the mp-QP problem (2.20) is not degenerate, then Theorem 1.15 implies that ${J^*}_z(x(0))$ is a $C^{(1)}$ function of $x(0)$ and therefore $J^*(x(0))$ is a $C^{(1)}$ function of $x(0)$. The results of Corollary 2.1 imply that $J^*(x(0))$ is piecewise quadratic. □

Remark 2.2. The relation between the design parameters of the optimal control problem (2.4) and the degeneracy of the mp-QP problem (2.20) are still under investigation. For example, it can be easily proven that the optimal control problem (2.4) with only input constraints leads to a non-degenerate mp-QP problem.

The solution of the multiparametric problem (2.20) provides the state feedback solution $u^*(k) = f_k(x(k))$ of CFTOC (2.4) for $k = 0$ and it also provides the open loop optimal control laws $u^*(k)$ as function of the initial state, i.e., $u^*(k) = u^*(k)(x(0))$. The state feedback PPWA optimal controllers $f_k : x(k) \mapsto u^*(k)$ for $k = 1, \ldots, N$ are computed in the following way. Consider the same CFTOC (2.4) over the shortened time horizon $[i, N]$

$$\begin{array}{ll} \min_{U_{N-i}} & \|Px_N\|_p + \sum_{k=i}^{N-1} \|Qx_k\|_p + \|Ru_k\|_p \\ \text{subj. to} & Ex_k + Lu_k \leq M,\ k = i, \ldots, N-1 \\ & x_{k+1} = Ax_k + Bu_k,\ k \geq 0 \\ & x_N \in \mathcal{X}_f \\ & x_i = x(i) \end{array} \tag{2.22}$$

where $U_{N-i} \triangleq [u'_i, \ldots, u'_{N-1}]$. We denote with $\mathcal{X}_i \subseteq \mathbb{R}^n$ the set of initial states $x(i)$ for which the optimal control problem (2.22) is feasible (as defined in (2.5)) and with U^*_{N-i} its optimizer. Problem (2.22) can be translated into the mp-QP

$$\begin{array}{ll} \min\limits_{U_{N-i}} & \frac{1}{2}{U_{N-i}}'HU_{N-i} + x'(i)FU_{N-i} \\ \text{subj. to } GU_{N-i} \leq W + Ex(i). & \end{array} \tag{2.23}$$

The first component of the multiparametric solution of (2.23) has the form

$$u^*(i) = f_i(x(i)),\ \forall x(i) \in \mathcal{X}_i, \tag{2.24}$$

where the control law $f_i : \mathbb{R}^n \to \mathbb{R}^m$, is continuous and PPWA

$$f_i(x) = F_i^j x + g_i^j \quad \text{if} \quad x \in CR_i^j, \; j = 1, \dots, N_i^r \tag{2.25}$$

and where the polyhedral sets $CR_i^j = \{x \in \mathbb{R}^n | H_i^j x \le K_i^j\}$, $j = 1, \dots, N_i^r$ are a partition in the broad sense of the feasible polyhedron $\mathcal{X}_i$. Therefore the feedback solution $u^*(k) = f_k(x(k)),\ k = 0, \dots, N-1$ of the CFTOC (2.4) is obtained by solving N mp-QP problems.

2.2.1 Complexity Analysis

A bound on the number N_i^r of polyhedral region of the PWA optimal control law f_i in (2.16) can be computed as explained in Section 1.4.5. The bound given in Section 1.4.5 represents an upper-bound on the number of different linear feedback gains which describe the controller. In practice, far fewer combinations are usually generated as x spans $\mathcal{X}_i$. Furthermore, the gains for the future input moves $u^*(i+1)(x(i)), \dots, u^*(N-1)(x(i))$ are not relevant for the control law. Thus, several different combinations of active constraints may lead to the same first m components $u^*(i)$ of the solution. On the other hand, the number N_i^r of regions of the piecewise affine solution is in general larger than the number of feedback gains, because non-convex critical regions may be split into several convex sets. For instance, the example reported in Fig. 2.3(a) involves 9 feedback gains, distributed over 23 regions of the state space. Therefore, for the characterization of the optimal controller, after step *2* of algorithm 1.4.1 the union of regions is computed where the first m components of the solution $U_{N-i}^*(x(i))$ are the same, by using the algorithm developed in [21]. This reduces the total number of regions in the partition for the optimal controller from N_i^r to ${N^{\text{oc}}}_i$.

Let $q_s \triangleq \operatorname{rank} S$, $q_s \le q$. For the dimension of the state $n > q_s$ the number of polyhedral regions N_i^r remains constant. To see this, consider the linear transformation $\bar{x} = Sx$, $\bar{x} \in \mathbb{R}^q$. Clearly $\bar{x}$ and x define the same set of active constraints, and therefore the number of partitions in the $\bar{x}$- and x-space are the same. Therefore, the number of partitions N_i^r of the x-space defining the optimal controller is insensitive to the dimension n of the state x for all $n \ge q_s$, i.e. to the number of parameters involved in the mp-QP. In particular, the additional parameters that we will introduce in Section 4.6 to extend optimal controller to reference tracking, disturbance rejection, soft constraints, variable constraints, and output feedback, do not affect significantly the number of polyhedral regions N_i^r (i.e., the complexity of the mp-QP) in the optimal controller (2.16)

The number q of constraints increases with N. For instance, $q = 2mN$ for control problems with input constraints only. From the analysis above, the larger N, the larger q, and therefore N_0^r. Note that many control problems involve input constraints only, and typically horizons $N = 2, 3$ or blocking of control moves are adopted, which reduces the number of constraints q.

2.2.2 Examples

Example 2.1. Consider the second order system

$$y(t) = \frac{2}{s^2 + 3s + 2} u(t),$$

sample the dynamics with $T = 0.1$ s, and obtain the state-space representation

$$\begin{cases} x(t+1) = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x(t) + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0 & 1.4142 \end{bmatrix} x(t) \end{cases} \tag{2.26}$$

We want to compute the state feedback optimal controller solution to problem (2.4) with $p = 2$, $N = 4$, $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.01$, P equal to the solution of the Lyapunov equation $P = A'PA + Q$, $\mathcal{X}_f = \mathbb{R}^2$ subject to the input constraints

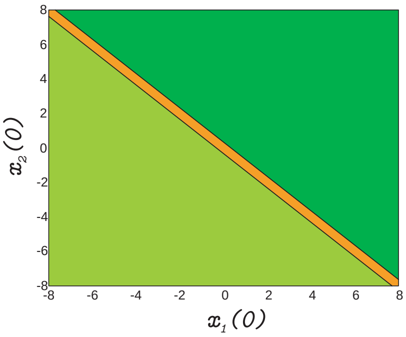
$$-2 \le u(k) \le 2, \; k = 0, \ldots, 3 \tag{2.27}$$

This task is addressed as shown in subsection (2.3). The feedback optimal solution $u^*(0), \ldots, u^*(3)$ was computed in less than 1 minute by solving 4 mp-QP problems in the region of the state space $\mathcal{X} = \{x \in \mathbb{R}^2 | \left[\begin{smallmatrix} -8 \\ -8 \end{smallmatrix}\right] \le x \le \left[\begin{smallmatrix} 8 \\ 8 \end{smallmatrix}\right]\}$. The corresponding polyhedral partitions of the state-space are depicted in Fig. 2.1. Only the last two optimal control moves are reported below:

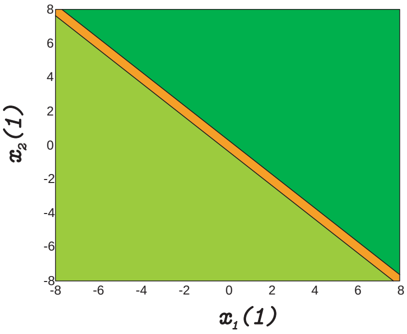
$$u^*(3) = \begin{cases} \begin{bmatrix} -9.597 & -9.627 \end{bmatrix} x(5) & \text{if } \begin{bmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \\ -1.000 & 0.000 \\ 0.000 & -1.000 \\ -0.706 & -0.708 \\ 0.706 & 0.708 \end{bmatrix} x(5) \le \begin{bmatrix} 8.000 \\ 8.000 \\ 8.000 \\ 8.000 \\ 0.147 \\ 0.147 \end{bmatrix} \quad \text{(Region \#1)} \\ 2.000 & \text{if } \begin{bmatrix} -1.000 & 0.000 \\ 0.000 & -1.000 \\ 0.706 & 0.708 \end{bmatrix} x(5) \le \begin{bmatrix} 8.000 \\ 8.000 \\ -0.147 \end{bmatrix} \quad \text{(Region \#2)} \\ -2.000 & \text{if } \begin{bmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \\ -0.706 & -0.708 \end{bmatrix} x(5) \le \begin{bmatrix} 8.000 \\ 8.000 \\ -0.147 \end{bmatrix} \quad \text{(Region \#3)} \end{cases}$$

$$u^*(2) = \begin{cases} \begin{bmatrix} -6.837 & -6.861 \end{bmatrix} x(4) & \text{if } \begin{bmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \\ -1.000 & 0.000 \\ 0.000 & -1.000 \\ -0.706 & -0.708 \\ 0.706 & 0.708 \end{bmatrix} x(4) \le \begin{bmatrix} 8.000 \\ 8.000 \\ 8.000 \\ 8.000 \\ 0.206 \\ 0.206 \end{bmatrix} \quad \text{(Region \#1)} \\ 2.000 & \text{if } \begin{bmatrix} 0.706 & 0.708 \\ -1.000 & 0.000 \\ 0.000 & -1.000 \end{bmatrix} x(4) \le \begin{bmatrix} -0.206 \\ 8.000 \\ 8.000 \end{bmatrix} \quad \text{(Region \#2)} \\ -2.000 & \text{if } \begin{bmatrix} -0.706 & -0.708 \\ 1.000 & 0.000 \\ 0.000 & 1.000 \end{bmatrix} x(4) \le \begin{bmatrix} -0.206 \\ 8.000 \\ 8.000 \end{bmatrix} \quad \text{(Region \#3)} \end{cases}$$

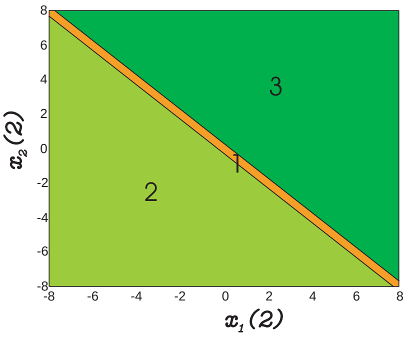
Figure 2.1 shows that that state feedback solution within the region of interests is defined over three regions at all time instants. The area of the



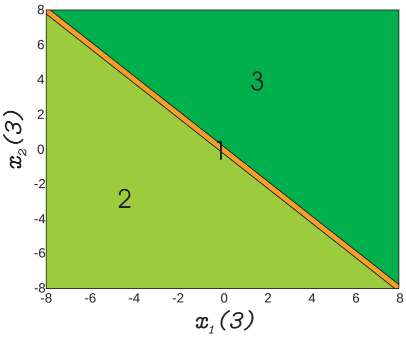
(a) Partition of the state space for the affine control law $u^*(0)$ ($N^{oc}{}_0 = 3$)



(b) Partition of the state space for the affine control law $u^*(1)$ ($N^{oc}{}_1 = 3$)



(c) Partition of the state space for the affine control law $u^*(2)$ ($N^{oc}{}_2 = 3$)



(d) Partition of the state space for the affine control law $u^*(3)$ ($N^{oc}{}_3 = 3$)

Fig. 2.1. Partition of the state space for optimal control law of Example 2.1

regions slightly changes from one step to another. Of course this is a special case, in general we expect the number of regions to increase with the horizon. Note that Figure 2.1 shows the resultant polyhedral partition once the regions with the same first component of the solution $u^*(i)$ have been joined as explained in Section 2.2.1. The mp-QP solver provides at each time step a number of critical region N_i^r that differs from the number $N^{oc}{}_i$ obtained after the union of the regions as explained in Section 2.2.1. For instance the polyhedral partition generated at time 0 consists of $N_0^r = 11$ regions and is depicted in Figure 2.2. In this example all the regions to the left and to the right of the unconstrained region have the first component of the optimal input saturated and can be joined in two big triangles.

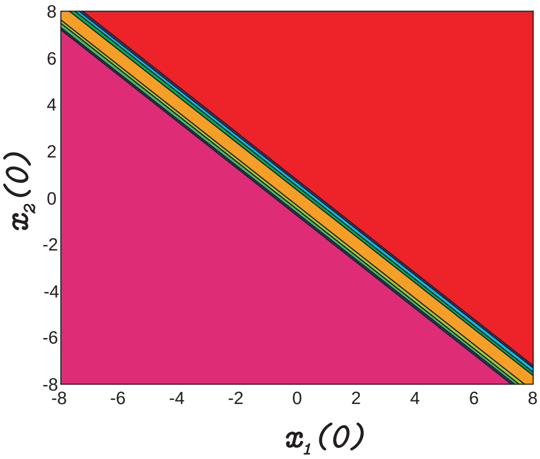


Fig. 2.2. Partition of the state space for the affine control law $u^*(0)$ of Example 2.1 before the union of the regions

Example 2.2. Consider the double integrator

$$y(t) = \frac{1}{s^2}u(t),$$

and its equivalent discrete-time state-space representation

$$\begin{cases} x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases} \tag{2.28}$$

obtained by setting $\ddot{y}(t) \approx \frac{\dot{y}(t+T)-\dot{y}(t)}{T}$, $\dot{y}(t) \approx \frac{y(t+T)-y(t)}{T}$, $T = 1$ s.

We want to compute the state feedback optimal controller that solves problem (2.4) with $p = 2$, $N = 5$, $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.1$, P equal to the solution of the Riccati equation (2.15), $\mathcal{X}_f = \mathbb{R}^2$ subject to the input constraints

$$-1 \le u(k) \le 1, \;\; k = 0, \ldots, 5 \tag{2.29}$$

and the state constraints

$$-15 \le x(k) \le 15, \;\; k = 0, \ldots, 5 \tag{2.30}$$

This task is addressed as shown in subsection (2.3). The feedback optimal solution $u^*(0), \ldots, u^*(5)$ was computed in less than 1 minute by solving 6 mp-QP problems and the corresponding polyhedral partitions of the state-space are depicted in Fig. 2.3. Only the last two optimal control moves are reported below:

$$u^*(5) = \begin{cases} \begin{bmatrix} -0.579 & -1.546 \end{bmatrix} x(5) & \text{if } \begin{bmatrix} 1.000 & 0.000 \\ -1.000 & 0.000 \\ -0.351 & -0.936 \\ 0.351 & 0.936 \end{bmatrix} x(5) \le \begin{bmatrix} 30.000 \\ 30.000 \\ 0.606 \\ 0.606 \end{bmatrix} \quad \text{(Region \#1)} \\ 1.000 & \text{if } \begin{bmatrix} 1.000 & 0.000 \\ -1.000 & 0.000 \\ 0.000 & -1.000 \\ 0.351 & 0.936 \end{bmatrix} x(5) \le \begin{bmatrix} 30.000 \\ 30.000 \\ 30.000 \\ -0.606 \end{bmatrix} \quad \text{(Region \#2)} \\ -1.000 & \text{if } \begin{bmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \\ -1.000 & 0.000 \\ -0.351 & -0.936 \end{bmatrix} x(5) \le \begin{bmatrix} 30.000 \\ 30.000 \\ 30.000 \\ -0.606 \end{bmatrix} \quad \text{(Region \#3)} \end{cases}$$

$$u^*(4) = \begin{cases} \begin{bmatrix} -0.579 & -1.546 \end{bmatrix} x(4) & \text{if } \begin{bmatrix} -0.351 & -0.936 \\ 0.351 & 0.936 \\ 0.767 & 0.641 \\ -0.767 & -0.641 \end{bmatrix} x(4) \le \begin{bmatrix} 0.606 \\ 0.606 \\ 2.428 \\ 2.428 \end{bmatrix} \\ & \text{(Region \#1)} \\ 1.000 & \text{if } \begin{bmatrix} -0.263 & -0.965 \\ 0.351 & 0.936 \\ 0.292 & 0.956 \\ -0.707 & -0.707 \end{bmatrix} x(4) \le \begin{bmatrix} 1.156 \\ -0.606 \\ -0.365 \\ 10.607 \end{bmatrix} \\ & \text{(Region \#2)} \\ -1.000 & \text{if } \begin{bmatrix} 0.263 & 0.965 \\ -0.351 & -0.936 \\ -0.292 & -0.956 \\ 0.707 & 0.707 \end{bmatrix} x(4) \le \begin{bmatrix} 1.156 \\ -0.606 \\ -0.365 \\ 10.607 \end{bmatrix} \\ & \text{(Region \#3)} \\ \begin{bmatrix} -0.435 & -1.425 \end{bmatrix} x(4) - 0.456 & \text{if } \begin{bmatrix} 0.707 & 0.707 \\ -0.292 & -0.956 \\ 0.292 & 0.956 \\ -0.767 & -0.641 \end{bmatrix} x(4) \le \begin{bmatrix} 10.607 \\ 0.977 \\ 0.365 \\ -2.428 \end{bmatrix} \\ & \text{(Region \#4)} \\ \begin{bmatrix} -0.435 & -1.425 \end{bmatrix} x(4) + 0.456 & \text{if } \begin{bmatrix} -0.707 & -0.707 \\ -0.292 & -0.956 \\ 0.292 & 0.956 \\ 0.767 & 0.641 \end{bmatrix} x(4) \le \begin{bmatrix} 10.607 \\ 0.365 \\ 0.977 \\ -2.428 \end{bmatrix} \\ & \text{(Region \#5)} \\ 1.000 & \text{if } \begin{bmatrix} 1.000 & 0.000 \\ 0.707 & 0.707 \\ -0.707 & -0.707 \\ 0.000 & -1.000 \\ 0.292 & 0.956 \\ 0.263 & 0.965 \end{bmatrix} x(4) \le \begin{bmatrix} 30.000 \\ 10.607 \\ 10.607 \\ 16.000 \\ -0.977 \\ -1.156 \end{bmatrix} \\ & \text{(Region \#6)} \\ -1.000 & \text{if } \begin{bmatrix} -1.000 & 0.000 \\ 0.707 & 0.707 \\ -0.000 & 1.000 \\ -0.707 & -0.707 \\ -0.292 & -0.956 \\ -0.263 & -0.965 \end{bmatrix} x(4) \le \begin{bmatrix} 30.000 \\ 10.607 \\ 16.000 \\ 10.607 \\ -0.977 \\ -1.156 \end{bmatrix} \\ & \text{(Region \#7)} \end{cases}$$

Note that by increasing the horizon N, the control law changes only far away from the origin, the more in the periphery the larger N. This must be expected from the results of Section 3.1, as the set where the CFTOC law approximates the constrained infinite-horizon linear quadratic regulator (CLQR) problem gets larger when N increases [50, 134].

Preliminary ideas about the constrained linear quadratic regulator for the double integrator were presented in [145], and are in full agreement with our results.

2.3 State Feedback Solution of CFTOC, $1, \infty$-Norm Case

In the following section we will concentrate on the use of ∞-norm, the results can be extended easily to cost functions with 1-norm or mixed $1/\infty$ norms.

The optimal control formulation (2.4) can be rewritten as a linear program by using the following standard approach (see e.g. [46]). The sum of components of any vector $\{\varepsilon_1^x, \dots, \varepsilon_N^x,\ \varepsilon_1^u, \dots, \varepsilon_N^u\}$ that satisfies

$$\begin{array}{l} -\mathbf{1}_n \varepsilon_k^x \leq Q x_k,\ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_n \varepsilon_k^x \leq -Q x_k,\ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_r \varepsilon_N^x \leq P x_N, \\ -\mathbf{1}_r \varepsilon_N^x \leq -P x_N, \\ -\mathbf{1}_m \varepsilon_k^u \leq R u_k,\ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_m \varepsilon_k^u \leq -R u_k,\ k = 0, 1, \dots, N-1 \end{array} \tag{2.31}$$

represents an upper bound on $J(U_N, x(0))$, where $\mathbf{1}_k \triangleq [\underbrace{1\ \dots\ 1}_{k}]'$,

$$x_k = A^k x_0 + \textstyle\sum_{j=0}^{k-1} A^j B u_{k-1-j}, \tag{2.32}$$

and the inequalities (2.31) hold componentwise. It is easy to prove that the vector $z \triangleq \{\varepsilon_0^x, \dots, \varepsilon_N^x, \varepsilon_0^u, \dots, \varepsilon_{N-1}^u, u_0, \dots, u_{N-1}\} \in \mathbb{R}^s$, $s \triangleq (m+1)N + N + 1$, that satisfies equations (2.31) and simultaneously minimizes $J(z) = \varepsilon_0^x + \dots + \varepsilon_N^x + \varepsilon_0^u + \dots + \varepsilon_{N-1}^u$ also solves the original problem (2.4), i.e., the same optimum $J^*(x(0))$ is achieved [160, 46]. Therefore, problem (2.4) can be reformulated as the following LP problem

$$\min_{z} \quad \left\{J(z) = \varepsilon_0^x + \ldots + \varepsilon_N^x + \varepsilon_0^u + \ldots + \varepsilon_{N-1}^u\right\} \tag{2.33a}$$

$$\text{subj. to} \quad -\mathbf{1}_n \varepsilon_k^x \leq \pm Q \left[A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \right], \tag{2.33b}$$

$$-\mathbf{1}_r \varepsilon_N^x \leq \pm P \left[A^N x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j} \right], \tag{2.33c}$$

$$-\mathbf{1}_m \varepsilon_k^u \leq \pm R u_k, \tag{2.33d}$$

$$E A^k x_0 + \sum_{j=0}^{k-1} E A^j B u_{k-1-j} + L u_k \leq M, \tag{2.33e}$$

$$A^k x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j} \in \mathcal{X}_f, \tag{2.33f}$$

$$k = 0, \ldots, N-1$$

$$x_0 = x(0) \tag{2.33g}$$

where constraints (2.33b)–(2.33f) are componentwise, and $\pm$ means that the constraint is duplicated for each sign, as in (2.31). Note that the $1-$norm in space requires the introduction of nN slack variables for the terms $\|Qx_k\|_1$, $\varepsilon_{k,i} \geq \pm Q^i x_k \ k = 0, 2, \ldots, N-1, \ i = 1, 2, \ldots, n$, plus r slack variables for the terminal penalty $\|Px_N\|_1$, $\varepsilon_{N,i} \geq \pm P^i x_N \ i = 1, 2, \ldots, r$, plus mN slack variables for the input terms $\|Ru_k\|_1$, $\varepsilon_{k,i}^u \geq \pm R^i u_k \ k = 0, 1, \ldots, N-1, \ i = 1, 2, \ldots, m$.

Note also that although any combination of 1- and ∞-norms leads to a linear program, our choice is motivated by the fact that ∞-norm over time could result in a poor closed-loop performance (only the largest state deviation and the largest input would be penalized over the prediction horizon), while 1-norm over space leads to an LP with a larger number of variables.

Denote with r and p the number of rows of P and E, respectively. Problem (2.33) can be rewritten in the more compact form

$$\begin{aligned} \min_{z} \quad & c'z \\ \text{subj. to } & Gz \leq W + Sx(0) \end{aligned} \tag{2.34}$$

where $c \in \mathbb{R}^s$, $G \in \mathbb{R}^{q \times s}$, $S \in \mathbb{R}^{q \times n}$, $W \in \mathbb{R}^q$, $q \triangleq 2(nN + 2mN + r) + pN$, and c and the submatrices G_ϵ, W_ϵ, S_ϵ associated with the constraints (2.33b)-(2.33d) are

$$
\begin{aligned}
c &= [\overbrace{1\ \dots\ 1}^{N+1}\ \overbrace{1\ \dots\ 1}^{N}\ \overbrace{0\ \dots\ 0}^{mN}]' \\
G_\epsilon &= \left[\begin{array}{cccccccccc}
\multicolumn{4}{c}{\overbrace{\hphantom{-\mathbf{1}_n\ 0\ \dots\ 0}}^{N+1}} & \multicolumn{3}{c}{\overbrace{\hphantom{0\ \dots\ 0}}^{N}} & \multicolumn{3}{c}{\overbrace{\hphantom{0\ 0\ \dots\ 0}}^{mN}} \\
-\mathbf{1}_n & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots\ 0 \\
-\mathbf{1}_n & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots\ 0 \\
0 & -\mathbf{1}_n & \dots & 0 & 0 & \dots & 0 & QB & 0 & \dots\ 0 \\
0 & -\mathbf{1}_n & \dots & 0 & 0 & \dots & 0 & -QB & 0 & \dots\ 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots\ \dots \\
0 & \dots & -\mathbf{1}_n & 0 & 0 & \dots & 0 & QA^{N-2}B & QA^{N-3}B & \dots\ 0 \\
0 & \dots & -\mathbf{1}_n & 0 & 0 & \dots & 0 & -QA^{N-2}B & -QA^{N-3}B & \dots\ 0 \\
0 & \dots & 0 & -\mathbf{1}_r & 0 & \dots & 0 & PA^{N-1}B & PA^{N-2}B & \dots\ PB \\
0 & \dots & 0 & -\mathbf{1}_r & 0 & \dots & 0 & -PA^{N-1}B & -PA^{N-2}B & \dots\ -PB \\
0 & 0 & \dots & 0 & -\mathbf{1}_m & \dots & 0 & R & 0 & \dots\ 0 \\
0 & 0 & \dots & 0 & -\mathbf{1}_m & \dots & 0 & -R & 0 & \dots\ 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots\ \dots \\
0 & 0 & \dots & 0 & 0 & \dots & -\mathbf{1}_m & 0 & 0 & \dots\ R \\
0 & 0 & \dots & 0 & 0 & \dots & -\mathbf{1}_m & 0 & 0 & \dots\ -R
\end{array}\right] \\
W_\epsilon &= [\overbrace{0\ \dots\ 0}^{2nN+2r}\ \overbrace{0\ \dots\ 0}^{2mN}]' \\
S_\epsilon &= [\overbrace{-I\ I\ (-QA)'\ (QA)'\ (-QA^2)'\ \dots\ (QA^{N-1})'}^{2n(N+1)}\ \overbrace{(-PA^N)'\ (PA^N)'}^{2r}\ \overbrace{0_n\ \dots\ 0_n}^{2mN}]'.
\end{aligned}
$$

As in the 2-norm case, by treating $x(0)$ as a vector of parameters, the problem (2.34) becomes a *multiparametric linear program* (mp-LP) that can be solved as described in Section 1.3. Once the multiparametric problem (2.33) has been solved for a polyhedral set $X \subset \mathbb{R}^n$ of states, the explicit solution $z^*(x(0))$ of (2.34) is available as a piecewise affine function of $x(0)$, and the optimal control law $u^*(0)$ is also available explicitly, as the optimal input $u^*(0)$ consists simply of m components of $z^*(x(0))$

$$u^*(0) = [0\ \dots 0\ I_m\ 0\ \dots\ 0] z^*(x(0)). \tag{2.35}$$

Theorem 1.8 states that the solution $z^*(x)$ of the mp-LP problem (2.34) is a continuous and piecewise affine function on polyhedra of x. Clearly the same properties are inherited by the controller. The following Corollaries of Theorem 1.8 summarize the analytical properties of the optimal control law and of the value function.

Corollary 2.3. *The control law $u^*(0) = f_0(x(0))$, $f_0 : \mathbb{R}^n \to \mathbb{R}^m$, obtained as a solution of the optimization problem (2.4) is continuous and PPWA*

$$f_0(x) = F_0^i x + g_0^i \quad \textit{if} \quad x \in CR_0^i,\ i = 1, \dots, N_0^r \tag{2.36}$$

where the polyhedral sets $CR_0^i \triangleq \{H_0^i x \le k_0^i\}$, $i = 1, \dots, N_0^r$, are a partition in the broad sense of the feasible set $\mathcal{X}_0$

Corollary 2.4. *The value function $J^*(x)$ obtained as a solution of the optimization problem (2.4) is convex and piecewise linear on polyhedra.*

Remark 2.3. Note from the results of Section 1.3, that if the optimizer of problem (2.4) is not unique for some $x(0)$ then a continuous optimizer function of the form (2.36) can always be chosen.

The multiparametric solution of (2.34) provides the open-loop optimal profile $u^*(0), \ldots, u^*(N-1)$ as an affine function of the initial state $x(0)$. The functions $f_k : x(k) \mapsto u^*(k)$ for $k = 1, \ldots, N$ can be computed as explained in Section 2.2 by solving N mp-LP problems.

2.3.1 Complexity Analysis

A bound on the number N_i^r of polyhedral region of the PWA optimal control law f_i in (2.16) can be computed as explained in Section 1.3.6. The same discussion of Section 2.2.1 applies in this context. In particular, the bound given in Section 1.3.6 represents an upper-bound on the number of different linear feedback gains which describe the controller. In practice, far fewer combinations are usually generated as x spans $\mathcal{X}_i$. Furthermore, the gains for the future input moves $u^*(i+1)(x(i)), \ldots, u^*(N-1)(x(i))$ are not relevant for the control law. Thus, several different combinations of active constraints may lead to the same first m components $u^*(i)$ of the solution. On the other hand, the number N_i^r of regions of the piecewise affine solution is in general larger than the number of feedback gains, because possible non-convex critical regions may be split into several convex sets. Therefore, for the characterization of the optimal controller, the union of regions is computed where the first m components of the solution $U_{N-i}^*(x(i))$ are the same. This reduces the total number of regions in the partition for the optimal controller from N_i^r to ${N^{\text{oc}}}_i$.

2.3.2 Example

Example 2.3. Consider the double integrator system (2.28). We want to compute the state feedback optimal controller that minimizes the performance measure

$$\sum_{k=0}^{3} \left\| \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{k+1} \right\|_\infty + |0.8 u_k| \tag{2.37}$$

subject to the input constraints

$$-1 \leq u(k) \leq 1, \;\; k = 0, \ldots, 3 \tag{2.38}$$

and the state constraints

$$-10 \leq x(k) \leq 10, \;\; k = 1, \ldots, 4 \tag{2.39}$$

This task is addressed as shown in subsection (2.3). The optimal feedback solution $u^*(0), \ldots, u^*(3)$ was computed in less than 1 minute by solving 4 mp-LP problems and the corresponding polyhedral partitions of the state-space are depicted in Fig. 2.4. Only the last two optimal control moves are reported below:

$$
u^*(2) = \begin{cases}
-1.00 & \text{if } \begin{bmatrix} 0.00 & 0.09 \\ 0.09 & 0.18 \\ -0.10 & -0.10 \\ -0.25 & -1.00 \\ 0.10 & 0.10 \\ -0.33 & -1.00 \end{bmatrix} x(2) \leq \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ -0.75 \\ 1.00 \\ -0.67 \end{bmatrix} \quad \text{(Region \#1)} \\
1.00 & \text{if } \begin{bmatrix} 0.00 & -0.09 \\ 0.10 & 0.10 \\ -0.09 & -0.18 \\ 0.25 & 1.00 \\ 0.33 & 1.00 \\ -0.10 & -0.10 \end{bmatrix} x(2) \leq \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ -0.75 \\ -0.67 \\ 1.00 \end{bmatrix} \quad \text{(Region \#2)} \\
0 & \text{if } \begin{bmatrix} 0.25 & 1.00 \\ -0.33 & -1.00 \\ 0.10 & 0.10 \end{bmatrix} x(2) \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 1.00 \end{bmatrix} \quad \text{(Region \#3)} \\
\begin{bmatrix} -0.50 & -1.50 \end{bmatrix} x(2) & \text{if } \begin{bmatrix} 0.10 & 0.10 \\ -1.00 & -1.00 \\ 0.33 & 1.00 \\ -0.33 & -1.00 \end{bmatrix} x(2) \leq \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \\ 0.67 \end{bmatrix} \quad \text{(Region \#4)} \\
\begin{bmatrix} -0.33 & -1.33 \end{bmatrix} x(2) & \text{if } \begin{bmatrix} -0.25 & -1.00 \\ -1.00 & -1.00 \\ 0.10 & 0.10 \\ 0.25 & 1.00 \end{bmatrix} x(2) \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 1.00 \\ 0.75 \end{bmatrix} \quad \text{(Region \#5)} \\
0 & \text{if } \begin{bmatrix} -0.25 & -1.00 \\ 0.33 & 1.00 \\ -0.10 & -0.10 \end{bmatrix} x(2) \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 1.00 \end{bmatrix} \quad \text{(Region \#6)} \\
\begin{bmatrix} -0.33 & -1.33 \end{bmatrix} x(2) & \text{if } \begin{bmatrix} 0.25 & 1.00 \\ 1.00 & 1.00 \\ -0.10 & -0.10 \\ -0.25 & -1.00 \end{bmatrix} x(2) \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 1.00 \\ 0.75 \end{bmatrix} \quad \text{(Region \#7)} \\
\begin{bmatrix} -0.50 & -1.50 \end{bmatrix} x(2) & \text{if } \begin{bmatrix} -0.10 & -0.10 \\ 1.00 & 1.00 \\ -0.33 & -1.00 \\ 0.33 & 1.00 \end{bmatrix} x(2) \leq \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \\ 0.67 \end{bmatrix} \quad \text{(Region \#8)}
\end{cases}
$$

$$
u^*(3) = \begin{cases}
1.00 & \text{if } \begin{bmatrix} -0.10 & -0.10 \\ 0.10 & 0.10 \\ 0.33 & 1.00 \\ 0.00 & -0.09 \end{bmatrix} x(3) \leq \begin{bmatrix} 1.00 \\ 1.00 \\ -0.67 \\ 1.00 \end{bmatrix} \quad \text{(Region \#1)} \\
-1.00 & \text{if } \begin{bmatrix} 0.10 & 0.10 \\ -0.10 & -0.10 \\ -0.33 & -1.00 \\ 0.00 & 0.09 \end{bmatrix} x(3) \leq \begin{bmatrix} 1.00 \\ 1.00 \\ -0.67 \\ 1.00 \end{bmatrix} \quad \text{(Region \#2)} \\
\begin{bmatrix} -0.50 & -1.50 \end{bmatrix} x(3) & \text{if } \begin{bmatrix} 0.10 & 0.10 \\ -0.10 & -0.10 \\ 0.33 & 1.00 \\ -0.33 & -1.00 \end{bmatrix} x(3) \leq \begin{bmatrix} 1.00 \\ 1.00 \\ 0.67 \\ 0.67 \end{bmatrix} \quad \text{(Region \#3)}
\end{cases}
$$

2.4 Time Varying Systems

The results of the previous sections can be applied to time-variant systems

$$\begin{cases} x(t+1) = A(t)x(t) + B(t)u(t) \\ y(t) = C(t)x(t), \end{cases} \tag{2.40}$$

subject to time-variant constraints

$$E(t)x(t) + L(t)u(t) \leq M(t). \tag{2.41}$$

The optimal control problem

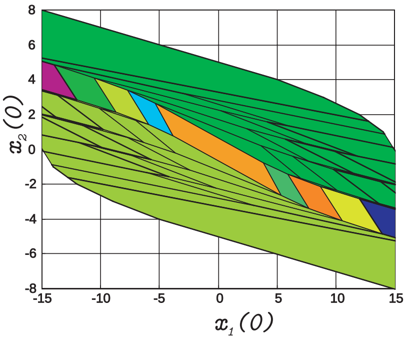
$$\begin{array}{rll} J^*(x(0)) = \min_{U_N} & J(U_N, x(0)) & \\ \text{subj. to} & E(k)x_k + L(k)u_k \leq M(k),\ k = 0, \ldots, N-1 & \\ & x_{k+1} = A(k)x_k + B(k)u_k,\ k \geq 0 & \\ & x_N \in \mathcal{X}_f & \\ & x_0 = x(0) & \end{array} \tag{2.42}$$

can be formulated and solved as a mp-QP or mp-LP depending on the norm p chosen. The following result is a simple corollary of the properties of the mp-QP and mp-LP

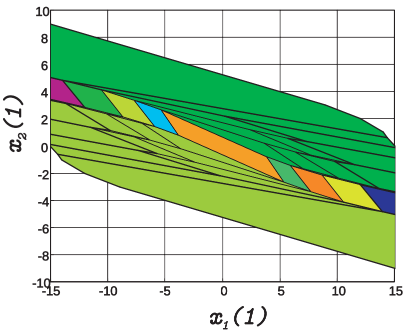
Corollary 2.5. *The control law $u^*(k) = f_k(x(k)),\ f_k : \mathbb{R}^n \rightarrow \mathbb{R}^m$, defined by the optimization problem (2.42) is continuous and piecewise affine*

$$f_k(x) = F_k^j x + g_k^j \quad \textit{if} \quad x \in CR_k^j,\ j = 1, \ldots, N_k^r \tag{2.43}$$

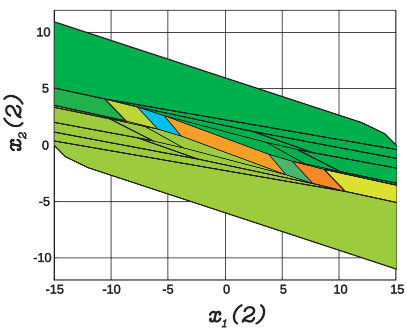
where the polyhedral sets $CR_k^j = \{x \in \mathbb{R}^n | H_k^j x \leq K_k^j\},\ j = 1, \ldots, N_k^r$ are a partition in the broad sense of the feasible polyhedron $\mathcal{X}_k$.



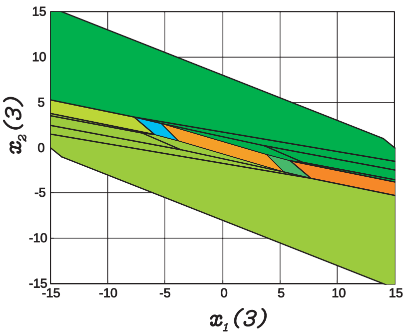
(a) Partition of the state space for the affine control law $u^*(0)$ ($N^{oc}{}_0 = 53$)



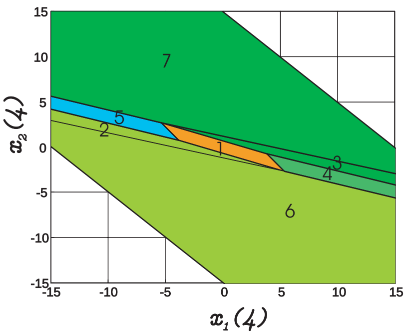
(b) Partition of the state space for the affine control law $u^*(1)$ ($N^{oc}{}_1 = 39$)



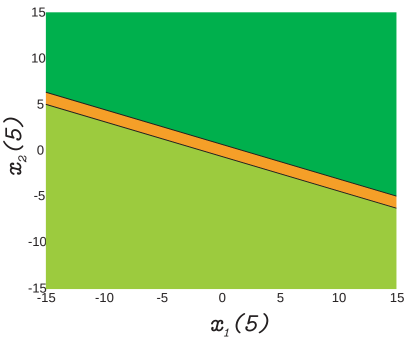
(c) Partition of the state space for the affine control law $u^*(2)$ ($N^{oc}{}_2 = 25$)



(d) Partition of the state space for the affine control law $u^*(3)$ ($N^{oc}{}_3 = 15$)

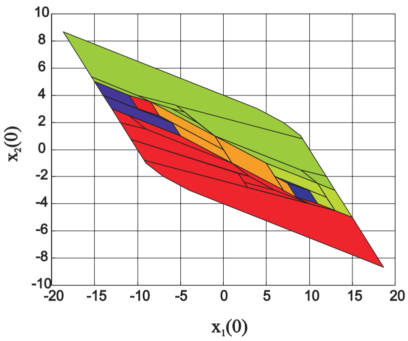


(e) Partition of the state space for the affine control law $u^*(4)$ ($N^{oc}{}_4 = 7$)

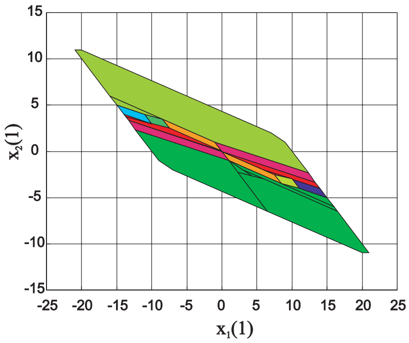


(f) Partition of the state space for the affine control law $u^*(5)$ ($N^{oc}{}_5 = 3$)

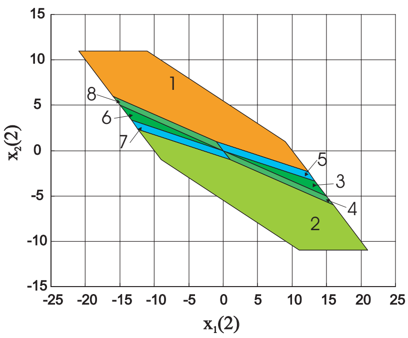
Fig. 2.3. Partition of the state space for optimal control law of Example 2.2



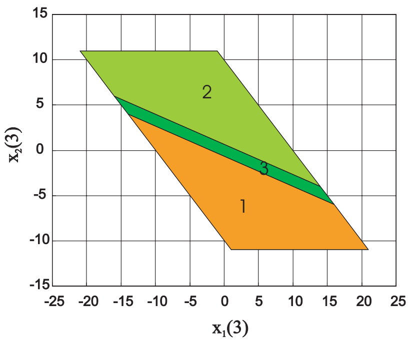
(a) Partition of the state space for the affine control law $u^*(0)$ ($N^{oc}{}_0 = 30$)



(b) Partition of the state space for the affine control law $u^*(1)$ ($N^{oc}{}_1 = 20$)



(c) Partition of the state space for the affine control law $u^*(2)$ ($N^{oc}{}_2 = 8$)



(d) Partition of the state space for the affine control law $u^*(3)$ ($N^{oc}{}_3 = 3$)

Fig. 2.4. Partition of the state space for optimal control law of Example 2.3

3

Constrained Infinite Time Optimal Control

3.1 Solution to the Infinite Time Constrained LQR Problem

Consider the infinite-horizon linear quadratic regulation problem with constraints (CLQR)

$$\begin{array}{rl} J_\infty^*(x(0)) = \min_{u_0,u_1,\ldots} & \sum_{k=0}^{\infty} \|Qx_k\|_2 + \|Ru_k\|_2 \\ \text{subj. to} & Ex_k + Lu_k \leq M,\ k = 0,\ldots,\infty \\ & x_{k+1} = Ax_k + Bu_k,\ k \geq 0 \\ & x_0 = x(0) \end{array} \tag{3.1}$$

and the set

$$\mathcal{X}_0{}^\infty = \{x(0) \in \mathbb{R}^n |\ \text{Problem (3.1) is feasible and } J_\infty^*(x(0)) < +\infty\}. \tag{3.2}$$

In their pioneering work [141], Sznaier and Damborg showed that the CFTOC problem (2.4), with $P = P_\infty$, P_∞ satisfying (2.15), sometimes also provides the solution to the infinite-horizon linear quadratic regulation problem with constraints (3.1). This holds for a certain set of initial conditions $x(0)$, which depends on the length of the finite horizon. This idea was reconsidered later by Chmielewski and Manousiouthakis [50] and independently by Scokaert and Rawlings [134], and recently also by Chisci and Zappa [49]. In [134], the authors extend the idea of [141] by showing that the CLQR is stabilizing for all $x(0) \in \mathcal{X}_0{}^\infty$, and that the CLQR problem can be solved by CFTOC (2.4) for a finite N. The horizon N depends on the initial state $x(0)$, and they provide an algorithm that computes an upper-bound on N for a given $x(0)$. On the other hand, Chmielewski and Manousiouthakis [50] describe an algorithm which provides a *semi-global* upper-bound for the horizon N. Namely, for any given compact set $\mathcal{X} \subseteq \mathcal{X}_0{}^\infty$ of initial conditions, the algorithm in [50] provides a sufficiently long horizon N such that the CFTOC (2.4) solves the infinite horizon problem (3.1). The algorithm has been revised recently in [25].

In Corollary 2.1 we have proven that the solution to (2.4) is continuous and piecewise affine on polyhedra. Therefore, for a compact polyhedral set of initial conditions $\mathcal{X} \subseteq \mathcal{X}_0{}^\infty$, the solution to the CLQR problem (3.1) is also continuous and piecewise affine on polyhedra.

Corollary 3.1. *The control law $u^*(k) = f(x(k))$, $f : \mathbb{R}^n \to \mathbb{R}^m$, obtained as a solution of the optimization problem (3.1) in a compact polyhedral set of the initial conditions $\mathcal{X} \subseteq \mathcal{X}_0{}^\infty$ is continuous, time invariant and piecewise affine on polyhedra*

$$f(x) = F^i x + g^i \quad \text{if} \quad x \in CR^i, \quad i = 1,\ldots,N^r \tag{3.3}$$

where the polyhedral sets $CR^i = \{x \in \mathbb{R}^n | H^i x \leq K^i\}$, $i = 1,\ldots,N^r$ are a partition in the broad sense of the polyhedron $\mathcal{X} \cap \mathcal{X}_0{}^\infty$

Note that if the constraint set $\{(x,u) \in \mathbb{R}^{n+m} | Ex + Lu \leq M\}$ is bounded then the set $\mathcal{X}_0{}^{\infty}$ is a compact set, and therefore the solution of the CFTOC (2.4) for an appropriately chosen N coincides with the global solution of CLQR (3.1).

One of the major drawback of the algorithm presented by Chmielewski and Manousiouthakis in [50] is that the *semi-global* upper-bound resulting from their algorithm is generally very conservative. The approach of Scokaert and Rawlings in [134] is less conservative but provides an horizon that is dependent on the initial state $x(0)$ and therefore cannot be used to compute the explicit solution (3.3) to the infinite time CLQR. In [79] Grieder et al. provide an algorithm to compute the solution (3.3) to the infinite time constrained linear quadratic regulator (3.1) in an efficient way. The algorithm makes use of a multiparametric quadratic program solver and of reachability analysis. To the author's knowledge there is no other algorithm in the literature to compute the solution (3.3) to the infinite time CLQR.

3.2 Examples

Example 3.1. The infinite time constrained LQR (3.1) was solved for Example 2.1 in less than 15 seconds in the region $\mathcal{X} = \{x \in \mathbb{R}^2 | \left[\begin{smallmatrix}-100\\-100\end{smallmatrix}\right] \leq x \leq \left[\begin{smallmatrix}100\\100\end{smallmatrix}\right]\}$ by using the approach presented in [79]. The state-space is divided into 3 polyhedral regions and is depicted in Fig. 3.1.

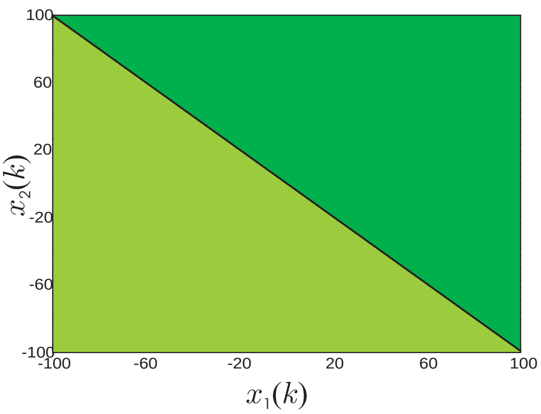


Fig. 3.1. Partition of the state space for infinite time optimal control law of Example 3.1 ($N^{\mathrm{oc}} = 3$)

Example 3.2. The infinite time constrained LQR (3.1) was solved for Example 2.2 by using the approach presented in [79] in less the 15 seconds. The state-space is divided into 1033 polyhedral regions and is depicted in Fig. 3.2.

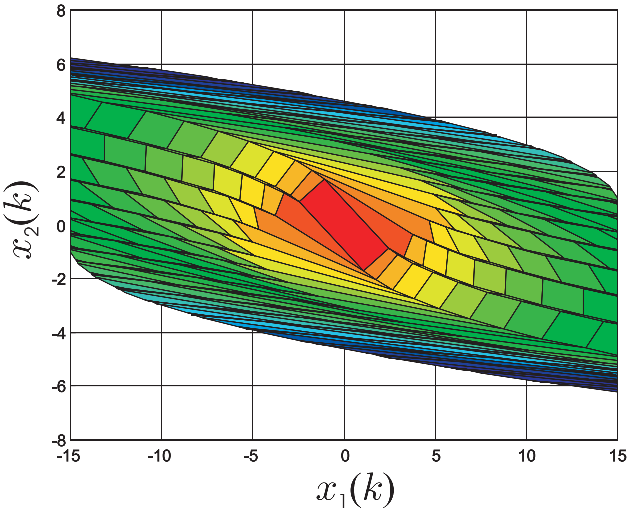


Fig. 3.2. Partition of the state space for infinite time optimal control law of Example 3.2 ($N^{oc} = 1033$)

4

Receding Horizon Control

Model predictive control is the only advanced control technology that has made a substantial impact on industrial control problems: its success is largely due to its almost unique ability to handle, simply and effectively, hard constraints on control and states.

(D.Mayne, 2001 [110])

4.1 Introduction

In the previous chapter we have discussed the solution of constrained finite time optimal control problems for linear systems. An infinite horizon controller can be designed by repeatedly solving finite time optimal control problems in a receding horizon fashion as described next. At each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon. The optimal command signal is applied to the process only during the following sampling interval. At the next time step a new optimal control problem based on new measurements of the state is solved over a shifted horizon. The resultant controller will be referred to as Receding Horizon Controller (RHC).

A receding horizon controller where the finite time optimal control law is computed by solving an on-line optimization problem is usually referred to as *Model Predictive Control* (MPC). For complex constrained multivariable control problems, model predictive control has become the accepted standard in the process industries [124]. The performance measure of the finite time optimal control problem solved by MPC at each time step is usually expressed as a *quadratic* or a *linear* criterion, so that the resulting optimization problem can be cast as a quadratic program (QP) or linear program (LP), respectively, for which a rich variety of efficient active-set and interior-point solvers are available.

Some of the first industrial MPC algorithms like IDCOM [128] and DMC [51] were developed for constrained MPC with quadratic performance indices. However, in those algorithms input and output constraints were treated in an indirect ad-hoc fashion. Only later, algorithms like QDMC [71] overcame this limitation by employing quadratic programming to solve constrained MPC problems with quadratic performance indices. Later an extensive theoretical effort was devoted to analyze such schemes, provide conditions for guaranteeing feasibility and closed-loop stability, and highlight the relations between MPC and linear quadratic regulation (see the recent survey [112]).

The use of linear programming was proposed in the early sixties by Zadeh and Whalen for solving optimal control problems [160], and by Propoi [123], who perhaps first conceived the idea of MPC. Later, only a few other authors have investigated MPC based on linear programming [48, 46, 73, 126], where the performance index is expressed as the sum of the ∞-norm or 1-norm of the input command and of the deviation of the state from the desired value.

Since RHC requires to solve at each sampling time an open-loop constrained finite time optimal control problem as a function of the current state, the results of the previous chapters lead to a different approach for RHC implementation. Precomputing off-line the explicit piecewise affine feedback policy (that provides the optimal control for all states) reduces the on-line computation of the RHC control law to a function evaluation, therefore avoiding the on-line solution of a quadratic or linear program.

This technique is attractive for a wide range of practical problems where the computational complexity of on-line optimization is prohibitive. It also provides insight into the structure underlying optimization-based controllers, describing the behaviour of the RHC controller in different regions of the state space. In particular, for RHC based on LP it highlights regions where idle control occurs and regions where the LP has multiple optima. Moreover, for applications where safety is crucial, a piecewise-affine control law is easier to verify than a mathematical program solver.

The new technique is not intended to replace MPC, especially not in some of the larger applications (systems with more than 50 inputs and 150 outputs have been reported from industry). It is expected to enlarge its scope of applicability to situations which cannot be covered satisfactorily with anti-windup schemes or where the on-line computations required for MPC are prohibitive for technical or cost reasons, such as those arising in the automotive and aerospace industries.

In this chapter we first review the basics of RHC. We discuss the stability of RHC and for RHC based on $1/\infty$-norm we provide guidelines for choosing the terminal weight so that closed-loop stability is achieved. Then, in Sections 4.3 and 4.4 the piecewise affine feedback control structure of QP-based and LP-based RHC is obtained as a simple corollary of the results of the previous chapters.

4.2 Problem Formulation

Consider the problem of regulating to the origin the discrete-time linear time-invariant system (2.1) while fulfilling the constraints (2.2).

Receding Horizon Control (RHC) solves such a constrained regulation problem in the following way. Assume that a full measurement of the state $x(t)$ is available at the current time t. Then, the finite time optimal control problem (2.4), (restated below)

$$\begin{array}{rl} J^*(x(t)) = \min_{U_t} & J(U_t, x(t)) \triangleq \|Px_N\|_p + \sum_{k=0}^{N-1} \|Qx_k\|_p + \|Ru_{k,t}\|_p \\ \text{subj. to} & Ex_k + Lu_{k,t} \leq M,\ k = 0, \dots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_{k+1} = Ax_k + Bu_{k,t},\ k \geq 0 \\ & x_0 = x(t) \end{array} \tag{4.1}$$

is solved at each time t, where $U_t = \{u_{0,t}, \ldots, u_{N-1,t}\}$ and where the time invariance of the plants permits the use of initial time 0 rather than t in the finite time optimal control problem. The sub-index t simply denotes the fact that the finite time optimal control problem (2.4) is solved for $x_0 = x(t)$.

Let $U_t^* = \{u_{0,t}^*, \ldots, u_{N-1,t}^*\}$ be the optimal solution of (4.1) at time t and $J^*(x(t))$ the corresponding value function. Then, the first sample of U_t^* is applied to system (2.1)

$$u(t) = u_{0,t}^*. \tag{4.2}$$

The optimization (4.1) is repeated at time $t+1$, based on the new state $x_0 = x(t+1)$, yielding a *moving* or *receding horizon* control strategy.

Denote by $f_0(x(k)) = u_{0,k}^*(x(k))$ the receding horizon control law when the current state is $x(k)$, the closed loop system obtained by controlling (2.1) with the RHC (4.1)-(4.2) is

$$x(k+1) = Ax(k) + Bf_0(x(k)) \triangleq s(x(k)), \; k \geq 0 \tag{4.3}$$

Note that the notation used in this chapter is slightly different from the previous chapter. Because of the receding horizon strategy, there is the need to distinguish between the input $u^*(k+l)$ applied to the plant at time $k+l$, and optimizer $u_{k,l}^*$ of the problem (4.1) at time $k+l$ obtained by solving (4.1) for $t = l$ and $x_0 = x(l)$.

To keep the exposition simpler we will assume that the constraints $Ex(k)+Lu(k) \leq M$ are decoupled in $x(k)$ and $u(k)$ for $k = 0, \ldots, N-1$ and we will denote by $\mathbb{X}$ and $\mathbb{U}$ the polyhedra of admissible states and inputs, respectively.

As in the previous chapter, $\mathcal{X}_i$ denotes the set of feasible states for problem (4.1) at time i

$$\mathcal{X}_i = \{x | \exists u \in \mathbb{U} \text{ such that } Ax + Bu \in \mathcal{X}_{i+1}\} \cap \mathbb{X}, \text{ with } \mathcal{X}_N = \mathcal{X}_f. \tag{4.4}$$

The main issue regarding the receding horizon control policy is the stability of the resulting closed-loop system (4.3). *In general, stability is not ensured by the RHC law (4.1)-(4.2).* Usually the terminal weight P and the terminal constraint set $\mathcal{X}_f$ are chosen to ensure closed-loop stability. In the next section we will briefly review sufficient conditions for stability, a more extensive treatment can be found in the survey [112, 111]. Before going further we need to recall the following definition

Definition 4.1. *A set $X \subseteq \mathbb{X}$ is said to be controlled invariant for system (2.1) if for every $x \in X$ there exists a $u \in \mathbb{U}$ such that $Ax + Bu \in X$*

Definition 4.2. *A set $X \subseteq \mathbb{X}$ is said to be positively invariant for system (4.3) if for every $x(0) \in X$ the system evolution satisfies $x(k) \in X$, $\forall k \in \mathbb{N}^+$.*

Definition 4.3. *A function $\phi : \mathbb{R}^+ \to \mathbb{R}^+$ is of* **class K** *if it is continuous, strictly increasing and $\phi(0) = 0$.*

The following theorem provides sufficient conditions for the stability of the equilibrium point of a discrete time system (see [154] p.267)

Theorem 4.1. *The equilibrium* $\mathbf{0}$ *of system (4.3) is uniformly asymptotically stable if there exist a function* $V : \mathbb{Z}^+ \times \mathbb{R}^n \to \mathbb{R}$, *a constant* r *and functions* α, β, γ *of class* K, *such that*

$$\begin{aligned} &V(k,0) = 0 \ \forall k \geq 0 \\ &\alpha(\|x\|) \leq V(k,x) \leq \beta(\|x\|) \ \forall k \geq 0 \ \forall x \text{ such that } \|x\| \leq r \\ &V(k+1, s(x)) - V(k,x) \leq -\gamma(\|x\|) \ \forall k \geq 0 \ \forall x \text{ such that } \|x\| \leq r \end{aligned}$$

Theorem 4.2. *The equilibrium* $\mathbf{0}$ *of system (4.3) is exponentially stable if there exist a function* $V : \mathbb{Z}^+ \times \mathbb{R}^n \to \mathbb{R}$, *constants* $a, b, c, r > 0$ *and* $d > 1$ *such that*

$$\begin{aligned} &V(k,0) = 0 \ \forall k \geq 0 \\ &a\|x\|^d \leq V(k,x) \leq b\|x\|^d \ \forall k \geq 0 \ \forall x \text{ such that } \|x\| \leq r \\ &V(k+1, s(x)) - V(k,x) \leq -c\|x\|^d \ \forall k \geq 0 \ \forall x \text{ such that } \|x\| \leq r \end{aligned}$$

4.2.1 Stability of RHC

In the survey papers [112, 111] the authors try to distill from the extensive literature on RHC and present the essential principles that ensure closed loop stability of RHC. In this section we will recall one of the main stability result presented in [112, 111]. The stability analysis is slightly modified to fit the structure of the book.

Theorem 4.3. *Assume that*

(A0) $Q = Q' \succeq 0$, $R = R' \succ 0$, $P \succeq 0$, *if* $p = 2$ *and* Q, R, P *are full column rank matrices if* $p = 1, \infty$.
(A1) *The sets* $\mathbb{X}$ *and* $\mathbb{U}$ *contain the origin and* $\mathcal{X}_f$ *is closed.*
(A2) $\mathcal{X}_f$ *is control invariant,* $\mathcal{X}_f \subseteq X$.
(A3) $\min_{u \in \mathbb{U},\ Ax+Bu \in \mathcal{X}_f} (\|P(Ax+Bu)\|_p - \|Px\|_p + \|Qx\|_p + \|Ru\|_p) \leq 0$, $\forall x \in \mathcal{X}_f$.
(A4) $\|Px\|_p \leq \beta(|x|)$, *where* $\beta(\cdot)$ *is a function of class* K.

Then,
(i) the state of the closed-loop system (4.3) converges to the origin, i.e., $\lim_{k\to\infty} x(k) = 0$,
(ii) the origin of the closed loop system (4.3) is asymptotically stable with domain of attraction $\mathcal{X}_0$.

Proof: The proof follows from Lyapunov arguments, close in spirit to the arguments of [96, 19] where it is established that the value function $J^*(\cdot)$ of (4.1) is a Lyapunov function for the closed-loop system. Let $x(0) \in \mathcal{X}_0$

and let $U_0^* = \{u_{0,0}, \ldots, u_{N-1,0}\}$ be the optimizer of problem (4.1) for $t = 0$ and $\mathbf{x}_0 = \{x_{0,0}, \ldots, x_{N,0}\}$ be the corresponding optimal state trajectory. Let $x(1) = x_{1,0} = Ax(0) + Bu_{0,0}$ and consider problem (4.1) for $t = 1$. We will construct an upper bound to $J^*(x(1))$. Consider the sequence $\tilde{U}_1 = \{u_{1,0}, \ldots, u_{N-1,0}, v\}$ and the corresponding state trajectory resulting from the initial state $x(1)$, $\tilde{\mathbf{x}}_1 = \{x_{1,0}, \ldots, x_{N,0}, Ax_{N,0} + Bv\}$. The input $\tilde{U}_1$ will be feasible for the problem at $t = 1$ iff $v \in \mathbb{U}$ keeps $x_{N,0}$ in $\mathcal{X}_f$ at step N of the prediction, i.e., $Ax_{N,0}+Bv \in \mathcal{X}_f$. Such v exists by hypothesis (A2). $J(x(1), \tilde{U}_1)$ will be an upperbound to $J^*(x(1))$. Since the trajectories generated by U_0^* and $\tilde{U}_1$ overlap, except for the first and last sampling intervals, it is immediate to show that

$$J^*(x(1)) \leq J(x(1), \tilde{U}_1) = J(x(0)) - \|Qx_{0,0}\|_p - \|Ru_{0,0}\|_p - \|Px_{N,0}\|_p + \\ +(\|Qx_{N,0}\|_p + \|Rv\|_p + \|P(Ax_{N,0} + Bv)\|_p) \tag{4.5}$$

Let $x = x_{0,0} = x(0)$ and $u = u_{0,0}$. Under assumption (A3) equation (4.5) becomes

$$J^*(Ax + Bu) - J^*(x) \leq -\|Qx\|_p - \|Ru\|_p, \ \forall x \in \mathcal{X}_0. \tag{4.6}$$

Equation (4.6) and the hypothesis (A0) on the matrices P and Q ensure that $J^*(x)$ decreases along the state trajectories of the closed loop system (4.3) for any $x \in \mathcal{X}_0$. Since $J^*(x)$ is lower-bounded by zero and since the state trajectories generated by the closed loop system (4.3) starting from any $x(0) \in \mathcal{X}_0$ lie in $\mathcal{X}_0$ for all $k \geq 0$, equation (4.6) is sufficient to ensure that the state of the closed loop system converges to zero as $k \to 0$ if the initial state lies in $\mathcal{X}_0$. We have proven (i).

Before proving (ii) we will first prove that $J^*(x) \leq \|Px\|_p, \ \forall x \in \mathcal{X}_f$ [91]. Note that for any $x \in \mathcal{X}_0$ and for any feasible input sequence $\{u_{0,0}, \ldots, u_{N-1,0}\}$ for problem (4.1) starting from the initial state $x = x_{0,0}$ whose corresponding state trajectory is $\{x_{0,0}, x_{1,0}, \ldots, x_{N,0}\}$ we have

$$J^*(x) \leq \sum_{i=0}^{N-1} (\|Qx_{i,0}\|_p + \|Ru_{i,0}\|_p) + \|Px_{N,0}\|_p \leq \|Px\|_p \tag{4.7}$$

where the last inequality follows from assumption (A3). Then from assumption (A4) we can state

$$J^*(x) \leq \beta(\|x\|), \ \forall x \in \mathcal{X}_f. \tag{4.8}$$

The stability of the origin follows from the positive invariance of $\mathcal{X}_0$, and from the fact that $J^*(x) \geq \alpha(\|x\|)$ for some function $\alpha(\cdot)$ of class K. To conclude, we have proven that $J^*(x)$ satisfies

$$\alpha(\|x\|) \leq J^*(x) \leq \beta(\|x\|) \ \ \forall x \in \mathcal{X}_f \tag{4.9}$$

$$J^*(s(x)) - J^*(x) \leq \gamma(\|x\|), \ \forall x \in \mathcal{X}_0 \tag{4.10}$$

Therefore, J^* satisfies the hypothesis of Theorem (4.1) and (ii) is proved.

□

Remark 4.1. Note that if the norm $p = 2$ is used in the optimal control problem (4.1), then from the proof of Theorem 4.3 and Theorem 4.1 we can ensure the *exponential stability* of the origin of the closed loop system (4.3).

Remark 4.2. Note that if $\mathcal{X}_f$ is a controlled invariant, then $\mathcal{X}_{i+1} \subseteq \mathcal{X}_i,\ i = 0, \dots, N-1$ and $\mathcal{X}_i$ are controlled invariant. Moreover $\mathcal{X}_0$ is positively invariant for the closed loop system (4.3) under the assumptions $(A1) - (A4)$ [111, 97].

Next we will discuss in more detail the hypothesis (A2) and (A3) by briefly reviewing the choices of P and $\mathcal{X}_f$ that appeared in the literature [111]. In part of the literature the constraint $\mathcal{X}_f$ is not used. However, in this literature the terminal region constraint $\mathcal{X}_f$ is implicit. In fact, if the constraint on the terminal state x_N is missing, then it is required that the horizon N is sufficiently large to ensure feasibility of the RHC (4.1)–(4.2) at all time instants t. This will automatically satisfy the constraint $x_N \in \mathcal{X}_f$ in (4.1)–(4.2) for all $t \geq 0$.

Keerthi and Gilbert [96] were first to propose specific choices of P and $\mathcal{X}_f$, namely $\mathcal{X}_f = 0$ and $P = 0$. Under this assumption (and a few other mild ones) Keerthi and Gilbert prove the stability for general nonlinear performance functions and nonlinear models. The proof follows the lines of the proof of Theorem 4.3.

The next two results were developed for RHC formulations based on the squared Euclidean norm ($p = 2$). A simple choice $\mathcal{X}_f$ is obtained by choosing $\mathcal{X}_f$ as the maximal output admissible set [75] for the closed-loop system $x(k+1) = (A + BK)x(k)$ where K is the associated unconstrained infinite-time optimal controller (2.14). With this choice the assumption (A3) in Theorem 4.3 becomes

$$x'(A'(P - PB(B'PB + R)^{-1}BP)A + Q - P)x \leq 0,\ \forall x \in \mathcal{X}_f \tag{4.11}$$

which is satisfied if P is chosen as the solution P_∞ of the the standard algebraic Riccati equation (2.15) for system (2.1).

If system (2.1) is asymptotically stable, then $\mathcal{X}_f$ can be chosen as the positively invariant set of the autonomous system $x(k+1) = Ax(k)$ subject to the state constraints $x \in \mathbb{X}$. Therefore in $\mathcal{X}_f$ the input $\mathbf{0}$ is feasible and the assumption (A3) in Theorem 4.3 becomes

$$-\|Px\|_2 + \|PAx\|_2 + \|Qx\|_2 \leq 0, \forall x \in \mathcal{X}_f \tag{4.12}$$

which is satisfied if P solves the standard Lyapunov equation.

In the following we extend the previous result to RHC based on the infinity norm, the extension to the one norm is an easy exercise. In particular we will show how to solve (4.12) when $p = \infty$.

4.2.2 Stability, ∞-Norm case

Consider the inequality (4.12) when $p = \infty$:

$$-\|Px\|_\infty + \|PAx\|_\infty + \|Qx\|_\infty \le 0,\ \forall x \in \mathbb{R}^n. \tag{4.13}$$

We assume that the matrix A is stable and we suggest a procedure for constructing a matrix P satisfying (4.13). Unlike the 2-norm case, the condition that matrix A has all the eigenvalues in the open disk $\|\lambda_i(A)\| < 1$ is not sufficient for the existence of a squared matrix P satisfying (4.13) [30].

First, we will focus on a simpler problem by removing the factor $\|Qx\|_\infty$ from condition (4.13)

$$-\|\tilde{P}x\|_\infty + \|\tilde{P}Ax\|_\infty \le 0,\ \forall x \in \mathbb{R}^n. \tag{4.14}$$

The existence and the construction of a matrix $\tilde{P}$ that satisfies condition (4.14), has been addressed in different forms by several authors [30, 122, 100, 80, 31, 32]. There are two equivalent ways of tackling this problem: find a Lyapunov function for the autonomous part of system (2.1) of the form

$$\Psi(x) = \|\tilde{P}x\|_\infty \tag{4.15}$$

with $\tilde{P} \in \mathbb{R}^{r\times n}$ full column rank, $r \ge n$, or equivalently compute a symmetrical positively invariant polyhedral set [32] for the autonomous part of system (2.1).

The following theorem, proved in [100, 122], states necessary and sufficient conditions for the existence of the Lyapunov function (4.15) satisfying condition (4.14).

Theorem 4.4. *The function $\Psi(x) = \|\tilde{P}x\|_\infty$ is a Lyapunov function for the autonomous part of system (2.1) if and only if there exists a matrix $H \in \mathbf{C}^{r\times r}$ such that*

$$\tilde{P}A - H\tilde{P} = 0 \tag{4.16a}$$

$$\|H\|_\infty < 1 \tag{4.16b}$$

□

In [100, 122] the authors proposed an efficient way to compute $\Psi(x)$ by constructing matrices $\tilde{P}$ and H satisfying (4.16).

In [122] the author shows how to construct matrices $\tilde{P}$ and H in (4.16) with the only assumption that A is stable. The number r of rows of $\tilde{P}$ is always finite for $|\lambda_i| < 1$, where $|\lambda_i|$ are the moduli of the eigenvalues of A. However this approach has the drawback that the number of rows r may go to infinity when some $|\lambda_i|$ approaches 1.

In [100] the authors construct a square matrix $\tilde{P} \in \mathbb{R}^{n\times n}$ under the assumption that the matrix A in (2.1) has distinct eigenvalues $\lambda_i = \mu_i + j\sigma_i$ located in the open square $|\mu_i| + |\sigma_i| < 1$.

What described so far can be used to solve the original problem (4.13). By using the results of [100, 122], the construction of a matrix P satisfying condition (4.13) can be performed by exploiting the following result:

Lemma 4.1. *Let $\tilde{P}$ and H be matrices satisfying conditions (4.16), with $\tilde{P}$ full rank. Let $\sigma \triangleq 1 - \|H\|_\infty$, $\rho \triangleq \|Q\tilde{P}^{\#}\|_\infty$, where $\tilde{P}^{\#} \triangleq (\tilde{P}'\tilde{P})^{-1}\tilde{P}'$ is the left pseudoinverse of $\tilde{P}$. The square matrix*

$$P = \frac{\rho}{\sigma}\tilde{P} \tag{4.17}$$

satisfies condition (4.13).

Proof: Since P satisfies $PA = HP$, we obtain $-\|Px\|_\infty + \|PAx\|_\infty + \|Qx\|_\infty = -\|Px\|_\infty + \|HPx\|_\infty + \|Qx\|_\infty \leq (\|H\|_\infty - 1)\|Px\|_\infty + \|Qx\|_\infty \leq (\|H\|_\infty - 1)\|Px\|_\infty + \|Q\tilde{P}^{\#}\|_\infty\|\tilde{P}x\|_\infty = 0$. Therefore, (4.13) is satisfied. □

To conclude, we point out that if in problem (4.1) we choose P to be the solution of (4.13) and $\mathcal{X}_f$ to be the positively invariant set of the autonomous system $x(k+1) = Ax(k)$ subject to the state constraints $x \in \mathbb{X}$, then by Theorem 4.3 the RHC law stabilize the system (2.1) in $\mathcal{X}_0$.

Remark 4.3. If $P \in \mathbb{R}^{n\times n}$ is given in advance rather than computed as in Proposition 4.1, condition (4.13) can be tested numerically, either by enumeration (3^{2n} LPs) or, more conveniently, through a mixed-integer linear program with $(5n+1)$ continuous variables and $4n$ integer variables.

Remark 4.4. If the matrix A is unstable, the procedure for constructing P can be still applied by pre-stabilizing system (2.1) via a linear controller without taking care of the constraints. Then, the output vector can be augmented by including the original (now state-dependent) inputs, and saturation constraints can be mapped into additional output constraints in (4.1).

Remark 4.5. In [126] the authors use a different approach based on Jordan decomposition to construct a stabilizing terminal weighting function for the RHC law (4.1)-(4.2). The resulting function leads to a matrix P with $r = 2^{n-n_0-1} + n2^{n_0-1}$ rows where n_0 is the algebraic multiplicity of the zero eigenvalues of matrix A. Apparently, the result holds only for matrices A with stable and real eigenvalues, and therefore, in general, the approach presented in this section is preferable.

4.3 State Feedback Solution of RHC, 2-Norm Case

The state feedback receding horizon controller (4.1)-(4.2) with $p = 2$ for system (2.1) can be obtained explicitly by setting

$$u(t) = f_0^*(x(t)), \tag{4.18}$$

where $f_0^*(x_0) : \mathbb{R}^n \to \mathbb{R}^m$ is the piecewise affine solution to the CFTOC (4.1) computed as explained in Section 2.21.

We remark that the implicit form (4.1)-(4.2) and the explicit form (4.18) are *equal*, and therefore the stability, feasibility, and performance properties mentioned in the previous sections are automatically inherited by the piecewise affine control law (4.18). Clearly, the explicit form (4.18) has the advantage of being easier to implement, and provides insight into the type of action of the controller in different regions CR_i of the state space.

4.3.1 Examples

Example 4.1. Consider the constrained double integrator system (2.28)–(2.30). We want to regulate the system to the origin by using the RHC problem (4.1)–(4.2) where $p = 2$, $N = 2$, $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.01$, and $P = P_\infty$ where P_∞ solves (2.15). We consider three cases:

1. no terminal state constraints: $\mathcal{X}_f = \mathbb{R}^n$,
2. $\mathcal{X}_f = 0$,
3. $\mathcal{X}_f$ is the positively invariant set of the closed-loop system $x(k+1) = (A+BL)$ where L is the infinite-time unconstrained optimal controller (2.14).

Case 1: $\mathcal{X}_f = \mathbb{R}^n$. The mp-QP problem associated with the RHC has the form (2.18) with

$$H = \begin{bmatrix} 0.8170 & 0.3726 \\ 0.3726 & 0.2411 \end{bmatrix}, \; F = \begin{bmatrix} 0.3554 & 0.1396 \\ 1.1644 & 0.5123 \end{bmatrix},$$

$$G = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \; W = \begin{bmatrix} 15 \\ 15 \\ 15 \\ 15 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \; E = \begin{bmatrix} -1 & -1 \\ 0 & -1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

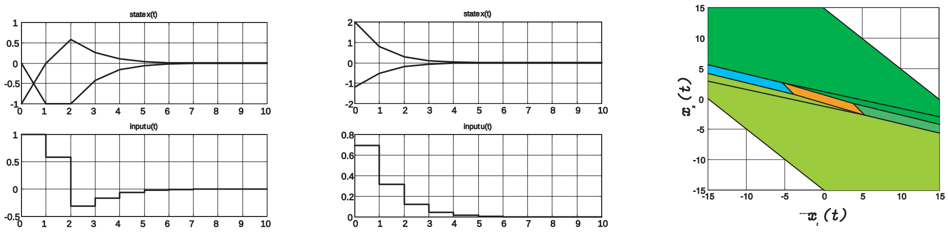
The solution of the mp-QP was computed in less than one second. The corresponding polyhedral partition of the state-space is depicted in Fig. 4.1. The RHC law is:

$$
u(t) = \begin{cases}
\begin{bmatrix} -0.579 & -1.546 \end{bmatrix} x(t) & \text{if } \begin{bmatrix} -0.351 & -0.936 \\ 0.351 & 0.936 \\ 0.767 & 0.641 \\ -0.767 & -0.641 \end{bmatrix} x(t) \le \begin{bmatrix} 0.606 \\ 0.606 \\ 2.428 \\ 2.428 \end{bmatrix} \\
& \text{(Region \#1)} \\
1.000 & \text{if } \begin{bmatrix} -0.263 & -0.965 \\ 0.351 & 0.936 \\ 0.292 & 0.956 \\ -0.707 & -0.707 \end{bmatrix} x(t) \le \begin{bmatrix} 1.156 \\ -0.606 \\ -0.365 \\ 10.607 \end{bmatrix} \\
& \text{(Region \#2)} \\
-1.000 & \text{if } \begin{bmatrix} 0.263 & 0.965 \\ -0.351 & -0.936 \\ -0.292 & -0.956 \\ 0.707 & 0.707 \end{bmatrix} x(t) \le \begin{bmatrix} 1.156 \\ -0.606 \\ -0.365 \\ 10.607 \end{bmatrix} \\
& \text{(Region \#3)} \\
\begin{bmatrix} -0.435 & -1.425 \end{bmatrix} x(t) - 0.456 & \text{if } \begin{bmatrix} 0.707 & 0.707 \\ -0.292 & -0.956 \\ 0.292 & 0.956 \\ -0.767 & -0.641 \end{bmatrix} x(t) \le \begin{bmatrix} 10.607 \\ 0.977 \\ 0.365 \\ -2.428 \end{bmatrix} \\
& \text{(Region \#4)} \\
\begin{bmatrix} -0.435 & -1.425 \end{bmatrix} x(t) + 0.456 & \text{if } \begin{bmatrix} -0.707 & -0.707 \\ -0.292 & -0.956 \\ 0.292 & 0.956 \\ 0.767 & 0.641 \end{bmatrix} x(t) \le \begin{bmatrix} 10.607 \\ 0.365 \\ 0.977 \\ -2.428 \end{bmatrix} \\
& \text{(Region \#5)} \\
1.000 & \text{if } \begin{bmatrix} 0.707 & 0.707 \\ -0.707 & -0.707 \\ 0.000 & -1.000 \\ 0.292 & 0.956 \\ 0.263 & 0.965 \end{bmatrix} x(t) \le \begin{bmatrix} 10.607 \\ 10.607 \\ 16.000 \\ -0.977 \\ -1.156 \end{bmatrix} \\
& \text{(Region \#6)} \\
-1.000 & \text{if } \begin{bmatrix} 0.707 & 0.707 \\ -0.000 & 1.000 \\ -0.707 & -0.707 \\ -0.292 & -0.956 \\ -0.263 & -0.965 \end{bmatrix} x(t) \le \begin{bmatrix} 10.607 \\ 16.000 \\ 10.607 \\ -0.977 \\ -1.156 \end{bmatrix} \\
& \text{(Region \#7)}
\end{cases}
$$

It can be noticed that the RHC law is simply the solution $u^*(2)$ of Example 2.2. We test the closed-loop behaviour from the initial condition $x(0) = [0, \ -1]$, which is depicted in Fig. 4.1(a) and from the initial condition $x(0) = [-1.2, \ 2]$, which is depicted in Fig. 4.1(b).

The union of the regions depicted in Fig. 4.1(c) is $\mathcal{X}_0$ and should not be confused with the region of attraction of the RHC closed-loop. We did not use any terminal constraints and therefore we cannot conclude that $\mathcal{X}_0$ is the region of attraction of the RHC. For instance, by starting at $x(0) = [-14, 14]$ (for which a feasible solution exists), the RHC controller runs into infeasibility at step $t = 2$.

Case 2: $\mathcal{X}_f = 0$. The mp-QP problem associated with the RHC has the form (2.18) with



(a) Closed-loop RHC trajectories from x(0)=(0,-1)

(b) Closed-loop RHC trajectories from x(0)=(-1.2,2)

(c) Polyhedral partition of the state-space, $N^{oc}{}_0 = 7$

Fig. 4.1. Double integrator Example (4.1), Case 1

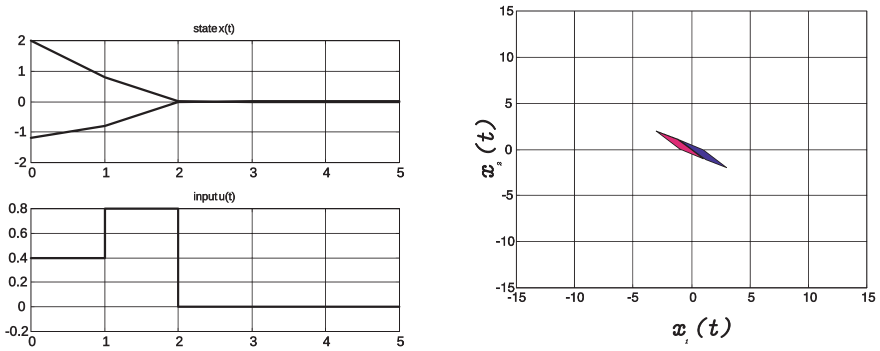
$$H = \begin{bmatrix} 0.8170 & 0.3726 \\ 0.3726 & 0.2411 \end{bmatrix}, \; F = \begin{bmatrix} 0.3554 & 0.1396 \\ 1.1644 & 0.5123 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \\ 1 & 1 \\ -1 & -1 \end{bmatrix}, \; W = \begin{bmatrix} 15.0000 \\ 15.0000 \\ 15.0000 \\ 15.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}, \; E = \begin{bmatrix} -1 & -1 \\ 0 & -1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -2 \\ 1 & 2 \\ 0 & -1 \\ 0 & 1 \end{bmatrix}$$

The solution of the mp-QP was computed in less than one second. The corresponding polyhedral partition of the state-space is depicted in Fig. 4.2. The RHC law is:

$$u(t) = \begin{cases} \begin{bmatrix} -0.579 & -1.546 \end{bmatrix} x(t) \text{ if } \begin{bmatrix} 0.680 & 0.734 \\ -0.680 & -0.734 \\ -0.683 & -0.730 \\ 0.683 & 0.730 \end{bmatrix} x(t) \le \begin{bmatrix} 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \end{bmatrix} & \text{(Region \#1)} \\[2ex] \begin{bmatrix} -1.000 & -2.000 \end{bmatrix} x(t) \text{ if } \begin{bmatrix} -0.447 & -0.894 \\ 0.707 & 0.707 \\ 0.447 & 0.894 \\ -0.678 & -0.735 \end{bmatrix} x(t) \le \begin{bmatrix} 0.447 \\ 0.707 \\ 0.447 \\ -0.000 \end{bmatrix} & \text{(Region \#2)} \\[2ex] \begin{bmatrix} -1.000 & -2.000 \end{bmatrix} x(t) \text{ if } \begin{bmatrix} 0.447 & 0.894 \\ -0.707 & -0.707 \\ -0.447 & -0.894 \\ 0.678 & 0.735 \end{bmatrix} x(t) \le \begin{bmatrix} 0.447 \\ 0.707 \\ 0.447 \\ -0.000 \end{bmatrix} & \text{(Region \#3)} \end{cases}$$

We test the closed-loop behaviour from the initial condition $x(0) = [-1.2,\ 2]$, which is depicted in Fig. 4.2(a) (the initial condition $x(0) = [0,\ -1]$



(a) Closed-loop RHC trajectories from $x(0) = [1, 1]$

(b) Polyhedral partition of the state-space, $N^{oc}{}_0 = 7$

Fig. 4.2. Double integrator Example (4.1), Case 2

is infeasible). The union of the regions depicted in Fig. 4.2(b) is $\mathcal{X}_0$. From Theorem 4.3, $\mathcal{X}_0$ is also the domain of attraction of the RHC law. However, by comparing Fig. 4.2 with Fig. 4.1, it can be noticed that the end-point constraint has two negative effects. First, the performance is worsened. Second, the feasibility region is extremely small.

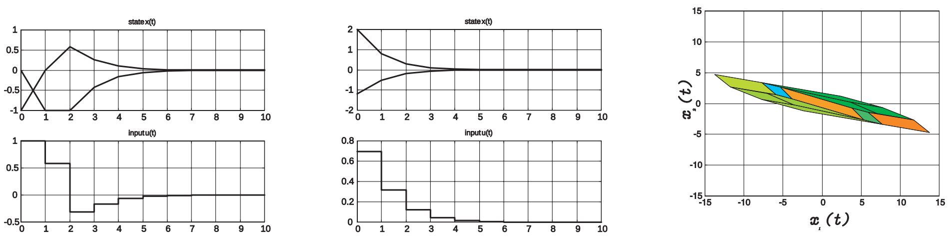
Case 3: $\mathcal{X}_f$ positively invariant set. The set $\mathcal{X}_f$ is

$$\mathcal{X}_f = \{x \in \mathbb{R}^2 | \begin{bmatrix} -0.5792 & -1.5456 \\ 0.5792 & 1.5456 \\ 0.3160 & 0.2642 \\ -0.3160 & -0.2642 \end{bmatrix} x \le \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix} \} \tag{4.19}$$

The mp-QP problem associated with the RHC has the form (2.18) with

$$H = \begin{bmatrix} 0.8170 & 0.3726 \\ 0.3726 & 0.2411 \end{bmatrix}, \quad F = \begin{bmatrix} 0.3554 & 0.1396 \\ 1.1644 & 0.5123 \end{bmatrix},$$

$$G = \begin{bmatrix} 0 & 0 \\ 1.0000 & 0 \\ 0 & 0 \\ -1.0000 & 0 \\ 1.0000 & 0 \\ -1.0000 & 0 \\ 0 & 1.0000 \\ 0 & -1.0000 \\ -2.1248 & -1.5456 \\ 2.1248 & 1.5456 \\ 0.5802 & 0.2642 \\ -0.5802 & -0.2642 \end{bmatrix}, \quad W = \begin{bmatrix} 15 \\ 15 \\ 15 \\ 15 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad E = \begin{bmatrix} -1.0000 & -1.0000 \\ 0 & -1.0000 \\ 1.0000 & 1.0000 \\ 0 & 1.0000 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.5792 & 2.7040 \\ -0.5792 & -2.7040 \\ -0.3160 & -0.8962 \\ 0.3160 & 0.8962 \end{bmatrix}$$



(a) Closed-loop RHC trajectories from x(0)=(1,1)

(b) Closed-loop RHC trajectories from x(0)=(-1.2,2)

(c) Polyhedral partition of the state-space, ${N^{oc}}_0 = 7$

Fig. 4.3. Double integrator Example (4.1) Case 3

The solution of the mp-QP was computed in less than one second. The corresponding polyhedral partition of the state-space is depicted in Fig. 4.3. The RHC law is:

$$u(t) = \begin{cases}
\begin{bmatrix} -0.579 & -1.546 \end{bmatrix} x(t) & \text{if } \begin{bmatrix} -0.351 & -0.936 \\ 0.351 & 0.936 \\ 0.767 & 0.641 \\ -0.767 & -0.641 \end{bmatrix} x(t) \leq \begin{bmatrix} 0.606 \\ 0.606 \\ 2.428 \\ 2.428 \end{bmatrix} \\ & \text{(Region \#1)} \\
1.000 & \text{if } \begin{bmatrix} -0.263 & -0.965 \\ -0.478 & -0.878 \\ 0.351 & 0.936 \\ 0.209 & 0.978 \\ 0.292 & 0.956 \end{bmatrix} x(t) \leq \begin{bmatrix} 1.156 \\ 1.913 \\ -0.606 \\ 0.152 \\ -0.365 \end{bmatrix} \\ & \text{(Region \#2)} \\
-1.000 & \text{if } \begin{bmatrix} 0.263 & 0.965 \\ 0.478 & 0.878 \\ -0.351 & -0.936 \\ -0.209 & -0.978 \\ -0.292 & -0.956 \end{bmatrix} x(t) \leq \begin{bmatrix} 1.156 \\ 1.913 \\ -0.606 \\ 0.152 \\ -0.365 \end{bmatrix} \\ & \text{(Region \#3)} \\
\begin{bmatrix} -0.435 & -1.425 \end{bmatrix} x(t) - 0.456 & \text{if } \begin{bmatrix} -0.292 & -0.956 \\ 0.292 & 0.956 \\ 0.729 & 0.685 \\ -0.767 & -0.641 \end{bmatrix} x(t) \leq \begin{bmatrix} 0.977 \\ 0.365 \\ 3.329 \\ -2.428 \end{bmatrix} \\ & \text{(Region \#4)} \\
\begin{bmatrix} -0.435 & -1.425 \end{bmatrix} x(t) + 0.456 & \text{if } \begin{bmatrix} -0.292 & -0.956 \\ 0.292 & 0.956 \\ -0.729 & -0.685 \\ 0.767 & 0.641 \end{bmatrix} x(t) \leq \begin{bmatrix} 0.365 \\ 0.977 \\ 3.329 \\ -2.428 \end{bmatrix} \\ & \text{(Region \#5)} \\
1.000 & \text{if } \begin{bmatrix} -0.209 & -0.978 \\ 0.209 & 0.978 \\ -0.333 & -0.943 \\ 0.292 & 0.956 \\ 0.263 & 0.965 \end{bmatrix} x(t) \leq \begin{bmatrix} 1.689 \\ -0.966 \\ 1.941 \\ -0.977 \\ -1.156 \end{bmatrix} \\ & \text{(Region \#6)} \\
1.000 & \text{if } \begin{bmatrix} -0.209 & -0.978 \\ 0.209 & 0.978 \\ -0.447 & -0.894 \\ 0.478 & 0.878 \end{bmatrix} x(t) \leq \begin{bmatrix} 0.966 \\ 0.152 \\ 2.860 \\ -1.913 \end{bmatrix} \\ & \text{(Region \#7)} \\
-1.000 & \text{if } \begin{bmatrix} -0.209 & -0.978 \\ 0.209 & 0.978 \\ 0.333 & 0.943 \\ -0.292 & -0.956 \\ -0.263 & -0.965 \end{bmatrix} x(t) \leq \begin{bmatrix} -0.966 \\ 1.689 \\ 1.941 \\ -0.977 \\ -1.156 \end{bmatrix} \\ & \text{(Region \#8)} \\
-1.000 & \text{if } \begin{bmatrix} -0.209 & -0.978 \\ 0.209 & 0.978 \\ 0.447 & 0.894 \\ -0.478 & -0.878 \end{bmatrix} x(t) \leq \begin{bmatrix} 0.152 \\ 0.966 \\ 2.860 \\ -1.913 \end{bmatrix} \\ & \text{(Region \#9)} \\
\begin{bmatrix} -0.273 & -1.273 \end{bmatrix} x(t) - 1.198 & \text{if } \begin{bmatrix} -0.209 & -0.978 \\ 0.209 & 0.978 \\ 0.707 & 0.707 \\ -0.729 & -0.685 \end{bmatrix} x(t) \leq \begin{bmatrix} 1.689 \\ -0.152 \\ 6.409 \\ -3.329 \end{bmatrix} \\ & \text{(Region \#10)} \\
\begin{bmatrix} -0.273 & -1.273 \end{bmatrix} x(t) + 1.198 & \text{if } \begin{bmatrix} -0.209 & -0.978 \\ 0.209 & 0.978 \\ -0.707 & -0.707 \\ 0.729 & 0.685 \end{bmatrix} x(t) \leq \begin{bmatrix} -0.152 \\ 1.689 \\ 6.409 \\ -3.329 \end{bmatrix} \\ & \text{(Region \#11)}
\end{cases}$$

We test the closed-loop behaviour from the initial condition $x(0) = [0,\ -1]$, which is depicted in Fig. 4.3(a). and from the initial condition $x(0) = [-1.2,\ 2]'$, which is depicted in Fig. 4.3(b). The union of the regions depicted in Fig. 4.3(c) is $\mathcal{X}_0$. Note that from Theorem 4.3 the set $\mathcal{X}_0$ is also the domain of attraction of the RHC law. The feasibility region is larger than in the case 2 and the performance is also improved with respect to case 2. However, since the prediction horizon is small, the feasibility region is smaller than the domain $\mathcal{X}_0^\infty$ of the infinite time CLQR in Figure 2.2.

4.4 State Feedback Solution of RHC, $1,\infty$-Norm Case

The state feedback receding horizon controller (4.1)–(4.2) with $p = 1, \infty$ for system (2.1) can be obtained explicitly by setting

$$u(t) = f_0^*(x(t)), \tag{4.20}$$

where $f_0^*(x_0) : \mathbb{R}^n \to \mathbb{R}^m$ is the piecewise affine solution to the CFTOC (4.1) computed as explained in Section 2.36. As in the 2 norm case the explicit form (4.20) has the advantage of being easier to implement, and provides insight into the type of action of the controller in different regions CR_i of the state space. Such insight will be discussed in detail in the following section.

4.4.1 Idle Control and Multiple Optima

There are two main issues regarding the implementation of an RHC control law based on linear programming: idle control and multiple solutions. The first corresponds to an optimal move $u(t)$ which is persistently zero, the second to the degeneracy of the LP problem. The explicit mp-LP approach of this book allows us to easily recognize both situations.

By analyzing the explicit solution of the RHC law, one can locate immediately the critical regions where the matrices F_0^i, g_0^i in (4.20) are zero, i.e., where the controller provides idle control. A different tuning of the controller is required if such polyhedral regions appear and the overall performance is not satisfactory.

The second issue is the presence of multiple solutions, that might arise from the degeneracy of the dual problem (1.15). Multiple optima are undesirable, as they might lead to a fast switching between the different optimal control moves when the optimization program (2.34) is solved on-line, unless interior-point methods are used. The mp-LP solvers [68, 37] can detect critical regions of degeneracy and partition them into sub-regions where a unique optimum is defined. In this case, the algorithm presented in [68] guarantees the continuity of the resulting optimal control law $u(t) = f_0^*(x(t))$, while the algorithm proposed in [37] could lead to a non-continuous piecewise affine control law $u(t) = f_0^*(x(t))$ inside regions of dual degeneracies. Example 4.4 will illustrate an RHC law where multiple optima and idle control occur.

4.4.2 Examples

Example 4.2. We provide here the explicit solution to the unconstrained RHC regulation example proposed in [126]. The non-minimum phase system

$$y(t) = \frac{s-1}{3s^2+4s+2}u(t)$$

is sampled at a frequency of 10 Hz, to obtain the discrete-time state-space model

$$\begin{cases} x(t+1) = \begin{bmatrix} 0.87212 & -0.062344 \\ 0.093516 & 0.99681 \end{bmatrix} x(t) + \begin{bmatrix} 0.093516 \\ 0.0047824 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0.33333 & -1 \end{bmatrix}. \end{cases}$$

In [126], the authors minimize $\sum_{k=0}^{N} 5|y_k| + |u_{k,t}|$, with the horizon length $N = 30$. Such an RHC problem can be rewritten in the form (4.1), by defining $Q = \left[\begin{smallmatrix} 1.6667 & -5 \\ 0 & 0 \end{smallmatrix}\right]$, $R = 1$ and $P = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$. Note that since Q, P are singular matrices, the sufficient condition for convergence of the closed loop system of Theorem 4.3 does not hold. The solution of the mp-LP problem was computed in 20 s by running the mp-LP solver [37] in Matlab on a Pentium III-450 MHz and the corresponding polyhedral partition of the state-space is depicted in Fig. 4.4(a). The RHC law is

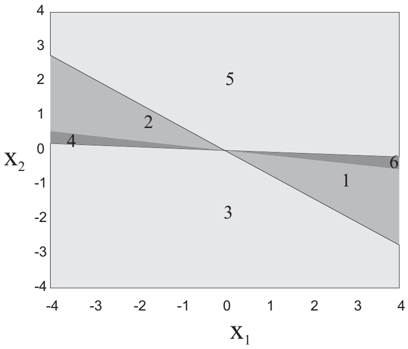
$$u = \begin{cases} 0 & \text{if } \left[\begin{smallmatrix} -108.78 & -157.61 \\ 5.20 & 37.97 \end{smallmatrix}\right] x \leq \left[\begin{smallmatrix} 0.00 \\ 0.00 \end{smallmatrix}\right] & \text{(Region \#1)} \\ 0 & \text{if } \left[\begin{smallmatrix} -5.20 & -37.97 \\ 108.78 & 157.61 \end{smallmatrix}\right] x \leq \left[\begin{smallmatrix} 0.00 \\ 0.00 \end{smallmatrix}\right] & \text{(Region \#2)} \\ \left[\begin{smallmatrix} -10.07 & -14.59 \end{smallmatrix}\right] x & \text{if } \left[\begin{smallmatrix} 20.14 & 29.19 \\ 0.09 & 1.86 \end{smallmatrix}\right] x \leq \left[\begin{smallmatrix} 0.00 \\ 0.00 \end{smallmatrix}\right] & \text{(Region \#3)} \\ \left[\begin{smallmatrix} -14.55 & -106.21 \end{smallmatrix}\right] x & \text{if } \left[\begin{smallmatrix} 29.11 & 212.41 \\ -1.87 & -38.20 \end{smallmatrix}\right] x \leq \left[\begin{smallmatrix} 0.00 \\ 0.00 \end{smallmatrix}\right] & \text{(Region \#4)} \\ \left[\begin{smallmatrix} -10.07 & -14.59 \end{smallmatrix}\right] x & \text{if } \left[\begin{smallmatrix} -20.14 & -29.19 \\ -0.09 & -1.86 \end{smallmatrix}\right] x \leq \left[\begin{smallmatrix} 0.00 \\ 0.00 \end{smallmatrix}\right] & \text{(Region \#5)} \\ \left[\begin{smallmatrix} -14.55 & -106.21 \end{smallmatrix}\right] x & \text{if } \left[\begin{smallmatrix} -29.11 & -212.41 \\ 1.87 & 38.20 \end{smallmatrix}\right] x \leq \left[\begin{smallmatrix} 0.00 \\ 0.00 \end{smallmatrix}\right] & \text{(Region \#6)} \end{cases} \tag{4.21}$$

In Figure 4.4(b) the closed-loop system is simulated from the initial state $x_0 = \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$. Note the idle control behaviour during the transient.

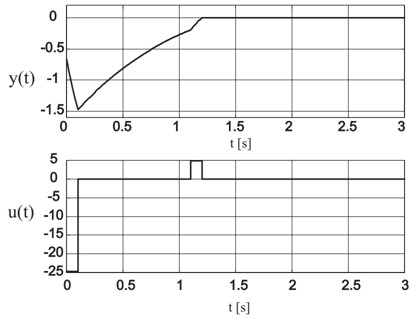
The same problem is solved by slightly perturbing $Q = \left[\begin{smallmatrix} 1.6667 & -5 \\ 0 & 0.0001 \end{smallmatrix}\right]$ so that it becomes nonsingular, and by adding the terminal weight

$$P = \begin{bmatrix} -705.3939 & -454.5755 \\ 33.2772 & 354.7107 \end{bmatrix} \tag{4.22}$$

computed as shown in Proposition 4.1 ($\tilde{P} = \left[\begin{smallmatrix} -1.934 & -1.246 \\ 0.0912 & 0.972 \end{smallmatrix}\right]$, $H = \left[\begin{smallmatrix} 0.9345 & 0.0441 \\ -0.0441 & 0.9345 \end{smallmatrix}\right]$, $\rho = 7.8262$, $\sigma = 0.021448$). The explicit solution was computed in 80 s and consists of 44 regions.

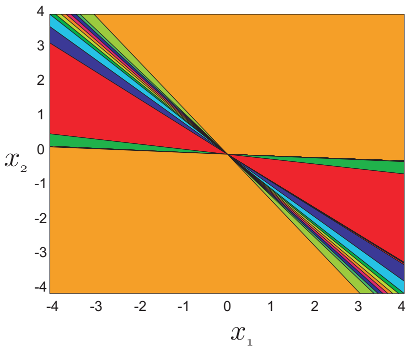


(a) Polyhedral partition of the state-space corresponding to the PPWA RHC (4.21)

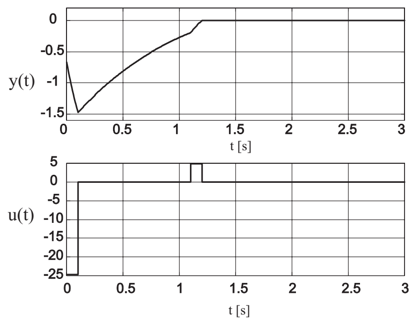


(b) Closed-loop RHC

Fig. 4.4. Example 4.2 with terminal weight $P = 0$



(a) Polyhedral partition of the state-space corresponding to the PPWA RHC in Example 4.2



(b) Closed-loop RHC

Fig. 4.5. Example 4.2 with terminal weight P as in (4.22)

In Figure 4.5(b) the closed-loop system is simulated from the initial state $x_0 = \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$.

Example 4.3. Consider the double integrator system (2.28). We want to regulate the system to the origin while minimizing at each time step t the performance measure

$$\sum_{k=0}^{1} \left\| \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k \right\|_\infty + |0.8 u_{k,t}| \tag{4.23}$$

subject to the input constraints

$$-1 \le u_{k,t} \le 1, \ k = 0, 1 \tag{4.24}$$

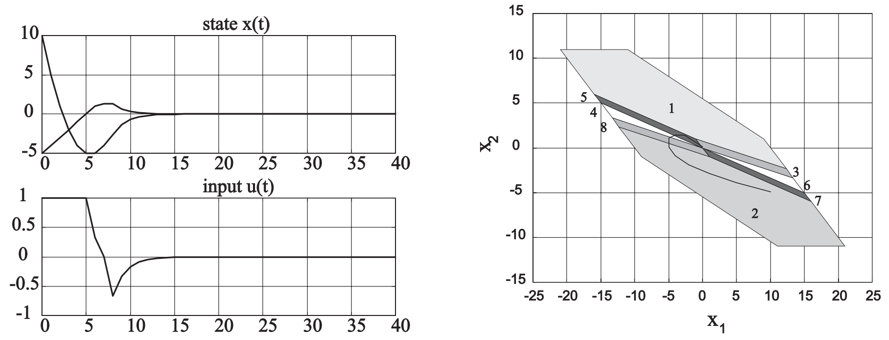
and the state constraints

$$-10 \le x_k \le 10, \ k = 1, 2. \tag{4.25}$$

This task is addressed by using the RHC algorithm (4.1)–(4.2) where $N = 2$, $Q = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.8$, $P = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, $\mathcal{X}_f = \mathbb{R}^2$. The solution of the mp-LP problem was computed in 13.57 s and the corresponding polyhedral partition of the state-space is depicted in Fig. 4.6(b). The resulting RHC law is

$$u = \begin{cases}
-1.00 & \text{if} \begin{bmatrix} 1.00 & 2.00 \\ 0.00 & 1.00 \\ -1.00 & -1.00 \\ -0.80 & -3.20 \\ 1.00 & 1.00 \\ -1.00 & -3.00 \end{bmatrix} x \le \begin{bmatrix} 11.00 \\ 11.00 \\ 10.00 \\ -2.40 \\ 10.00 \\ -2.00 \end{bmatrix} \quad \text{(Region \#1)} \\
1.00 & \text{if} \begin{bmatrix} 0.80 & 3.20 \\ -1.00 & -2.00 \\ -1.00 & -1.00 \\ 1.00 & 1.00 \\ 0.00 & -1.00 \\ 1.00 & 3.00 \end{bmatrix} x \le \begin{bmatrix} -2.40 \\ 11.00 \\ 10.00 \\ 10.00 \\ 11.00 \\ -2.00 \end{bmatrix} \quad \text{(Region \#2)} \\
\begin{bmatrix} -0.33 & -1.33 \end{bmatrix} x & \text{if} \begin{bmatrix} 0.53 & 2.13 \\ 0.67 & 0.67 \\ -1.00 & -1.00 \\ -0.33 & -1.33 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 1.00 \end{bmatrix} \quad \text{(Region \#3)} \\
0 & \text{if} \begin{bmatrix} -0.80 & -3.20 \\ 1.00 & 3.00 \\ -1.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \quad \text{(Region \#4)} \\
\begin{bmatrix} -0.50 & -1.50 \end{bmatrix} x & \text{if} \begin{bmatrix} -1.00 & -1.00 \\ 0.50 & 0.50 \\ -0.80 & -2.40 \\ 0.50 & 1.50 \end{bmatrix} x \le \begin{bmatrix} 10.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{bmatrix} \quad \text{(Region \#5)} \\
0 & \text{if} \begin{bmatrix} 0.80 & 3.20 \\ -1.00 & -3.00 \\ 1.00 & 1.00 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \quad \text{(Region \#6)} \\
\begin{bmatrix} -0.50 & -1.50 \end{bmatrix} x & \text{if} \begin{bmatrix} 1.00 & 1.00 \\ -0.50 & -0.50 \\ 0.80 & 2.40 \\ -0.50 & -1.50 \end{bmatrix} x \le \begin{bmatrix} 10.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{bmatrix} \quad \text{(Region \#7)} \\
\begin{bmatrix} -0.33 & -1.33 \end{bmatrix} x & \text{if} \begin{bmatrix} -0.53 & -2.13 \\ -0.67 & -0.67 \\ 1.00 & 1.00 \\ 0.33 & 1.33 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 1.00 \end{bmatrix} \quad \text{(Region \#8)}
\end{cases}$$

Note that region #1 and #2 correspond to the saturated controller, and regions #4 and #6 to idle control. The same example was solved for an increasing number of degrees of freedom N. The corresponding polyhedral partitions are reported in Fig. 4.7. Note that the white regions correspond to the saturated controller $u(t) = -1$ in the upper part, and $u(t) = -1$ in the lower part. The off-line computation times and number of regions $N^{\mathrm{oc}}{}_0$ in the RHC control law (4.20) are reported in Table 4.1.



(a) Closed-loop RHC

(b) Polyhedral partition of the state-space and closed-loop RHC trajectories

Fig. 4.6. Double integrator Example 4.3

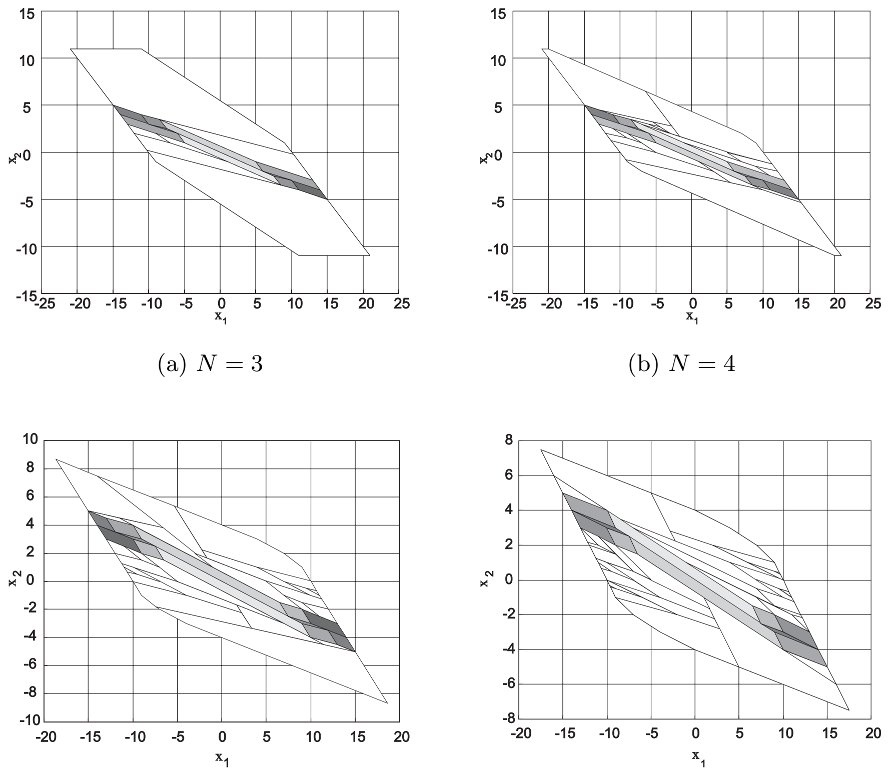
Free moves N	Computation time (s)	N. of regions $N^{oc}{}_0$
2	13.57	8
3	28.50	16
4	48.17	28
5	92.61	37
6	147.53	44

Table 4.1. Off-line computation times and number of regions $N^{oc}{}_0$ in the RHC control law (4.20) for the double integrator example

Example 4.4. Consider again the double integrator of Example 4.3, with the performance index

$$\min_{u_{0,t}} \left\| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_1 \right\|_\infty + |u_{0,t}| \tag{4.26}$$

and again subject to constraints (4.24)– (4.25). The associated mp-LP problem is



(a) $N = 3$ (b) $N = 4$

(c) $N = 5$ (d) $N = 6$

Fig. 4.7. Polyhedral partition of the state-space corresponding to the PPWA RHC controller in Example 4.3

$$\min_{\varepsilon_1,\varepsilon_2,u_t} \quad \varepsilon_1+\varepsilon_2$$

subj. to

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ u_t \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 10 \\ 10 \\ 10 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ -1 & -1 \\ 0 & -1 \\ 0 & -1 \\ -1 & -1 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} x(t) \tag{4.27}$$

The solution of (4.27) was computed in 0.5s and the corresponding polyhedral partition of the state-space is depicted in Fig. 4.8. The RHC law is

$$u = \begin{cases} \text{degenerate if } \begin{bmatrix} -1.00 & -2.00 \\ 1.00 & 0.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} & \text{(Region \#1)} \\ 0 \quad \text{if } \begin{bmatrix} 1.00 & 0.00 \\ 1.00 & 2.00 \\ -1.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} & \text{(Region \#2)} \\ \text{degenerate if } \begin{bmatrix} -1.00 & 0.00 \\ 1.00 & 2.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} & \text{(Region \#3)} \\ 0 \quad \text{if } \begin{bmatrix} -1.00 & -2.00 \\ -1.00 & 0.00 \\ 1.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} & \text{(Region \#4)} \end{cases}$$

Note the presence of idle control and multiple optima in region #2, #4 and #1, #3, respectively. The algorithm in [37] returns two possible sub-partitions of the degenerate regions #1, #3. Region #1 can be partitioned as

$$u_{1A} = \begin{cases} 0 \quad \text{if } \begin{bmatrix} -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} & \text{(Region \#1a)} \\ \begin{bmatrix} 0.00 & -1.00 \end{bmatrix} x + 10.00 \quad \text{if } \begin{bmatrix} 0.00 & -2.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} -20.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} & \text{(Region \#1b)} \end{cases}$$

or

$$u_{1B} = \begin{cases} -1.00 \quad \text{if } \begin{bmatrix} -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} -1.00 \\ -1.00 \\ 11.00 \end{bmatrix} & \text{(Region \#1a)} \\ 0 \quad \text{if } \begin{bmatrix} 1.00 & 2.00 \\ -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} & \text{(Region \#1b)} \\ \begin{bmatrix} 0.00 & -1.00 \end{bmatrix} x + 10.00 \quad \text{if } \begin{bmatrix} 1.00 & 2.00 \\ 0.00 & -2.00 \\ -1.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} & \text{(Region \#1c)} \\ 0 \quad \text{if } \begin{bmatrix} -1.00 & 0.00 \\ -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ -1.00 \\ 0.00 \\ 10.00 \end{bmatrix} & \text{(Region \#1d)} \\ \begin{bmatrix} 0.00 & -1.00 \end{bmatrix} x + 10.00 \quad \text{if } \begin{bmatrix} -1.00 & 0.00 \\ 0.00 & -2.00 \\ 1.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} & \text{(Region \#1e)} \end{cases}$$

Region #3 can be partitioned symmetrically as:

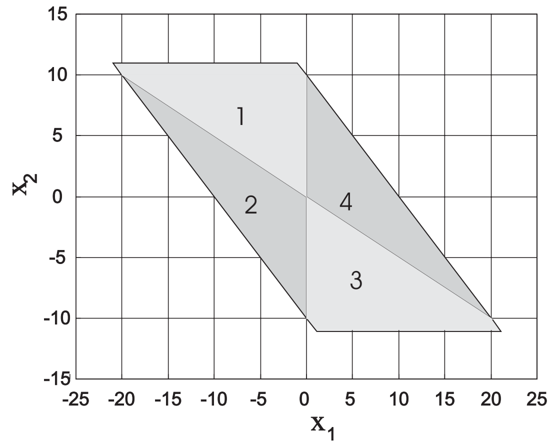


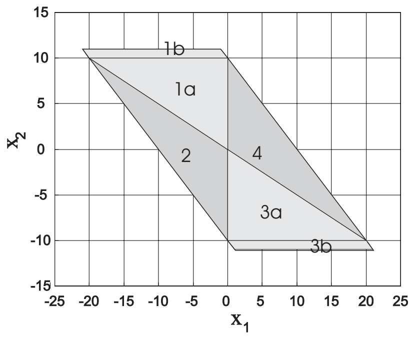
Fig. 4.8. Polyhedral partition of the state-space corresponding to the PPWA solution to problem (4.26)

$$u_{2A} = \begin{cases} 0 & \text{if } \begin{bmatrix} -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \quad \text{(Region \#3a)} \\ \begin{bmatrix} 0.00 & -1.00 \end{bmatrix} x - 10.00 & \text{if } \begin{bmatrix} 0.00 & 2.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} -20.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} \quad \text{(Region \#3b)} \end{cases}$$

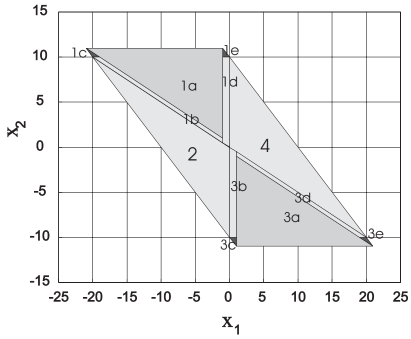
or

$$u_{2B} = \begin{cases} 1.00 & \text{if } \begin{bmatrix} -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} -1.00 \\ -1.00 \\ 11.00 \end{bmatrix} \quad \text{(Region \#3a)} \\ & \text{if } \begin{bmatrix} 1.00 & 0.00 \\ -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \quad \text{(Region \#3b)} \\ \begin{bmatrix} 0.00 & -1.00 \end{bmatrix} x - 10.00 & \text{if } \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 2.00 \\ -1.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} \quad \text{(Region \#3c)} \\ & \text{if } \begin{bmatrix} -1.00 & -2.00 \\ -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ -1.00 \\ 0.00 \\ 10.00 \end{bmatrix} \quad \text{(Region \#3d)} \\ \begin{bmatrix} 0.00 & -1.00 \end{bmatrix} x - 10.00 & \text{if } \begin{bmatrix} -1.00 & -2.00 \\ 0.00 & 2.00 \\ 1.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} \quad \text{(Region \#3e)} \end{cases}$$

Two possible explicit solutions to problem (4.26) are depicted in Figure 4.9. Note that controllers u_{1A} and u_{2A} are continuous in Regions 1 and 2, respectively, while controllers u_{1B} and u_{2B} are discontinuous in Regions 1 and 2, respectively.



(a) A possible solution of Example 4.4 obtained by choosing, for the degenerate regions 1 and 3, the control laws u_{1A} and u_{2A}



(b) A possible solution of Example 4.4 obtained by choosing, for the degenerate regions 1 and 3, the control laws u_{1B} and u_{2B}

Fig. 4.9. Double integrator in Example 4.3: example of degeneracy

4.5 On-Line Computation Time

The simplest way to implement the piecewise affine feedback laws (4.18)–(4.20) is to store the polyhedral cells $\{H^i x \leq K^i\}$, perform on-line a linear search through them to locate the one which contains $x(t)$ and then look up the corresponding feedback gain (F^i, g^i). This procedure can be easily parallelized (while for a QP or LP solver the parallelization is less obvious). However, more efficient on-line implementation techniques which avoid the storage and the evaluation of the polyhedral cells have been developed and will be presented in Chapter 5.

4.6 RHC Extensions

In order to down-sizes the optimization problem at the price of possibly reduced performance, the basic RHC formulation (4.1) is usually modified as follows

$$\begin{array}{ll}\min\limits_{U_t} & \left\{ \|Px'_{N_y}\|_p + \sum\limits_{k=0}^{N_y-1} [\|Qx_k\|_p + \|Ru_{k,t}\|_p] \right\} \\ \text{subj. to} & y_{\min} \leq y_k \leq y_{\max},\ k = 1, \ldots, N_c \\ & u_{\min} \leq u_{k,t} \leq u_{\max},\ k = 0, 1, \ldots, N_u \\ & x_0 = x(t) \\ & x_{k+1} = Ax_k + Bu_{k,t},\ k \geq 0 \\ & y_k = Cx_k,\ k \geq 0 \\ & u_{k,t} = Kx_k,\ N_u \leq k < N_y \end{array} \tag{4.28}$$

where K is some feedback gain, N_y, N_u, N_c are the output, input, and constraint horizons, respectively, with $N_u \le N_y$ and $N_c \le N_y - 1$.

Corollary 2.1 and Corollary 2.3 also apply to the extensions of RHC to reference tracking, disturbance rejection, soft constraints, variable constraints, and output feedback which will be discussed in the following.

Formulation (4.28) can be extended naturally to situations where the control task is more demanding. As long as the control task can be expressed as an mp-QP or mp-LP, a piecewise affine controller results which can be easily implemented. In this section we will mention only a few extensions to illustrate the potential. To our knowledge, these types of problems are difficult to formulate from the point of view of anti-windup or other techniques not related to RHC.

Reference Tracking

The controller can be extended to provide offset-free tracking of asymptotically constant reference signals. Future values of the reference trajectory can be taken into account by the controller explicitly so that the control action is optimal for the future trajectory in the presence of constraints.

Let the goal be to have the output vector $y(t)$ track $r(t)$, where $r(t) \in \mathbb{R}^p$ is the reference signal. To this aim, consider the RHC problem

$$\begin{array}{ll} \min\limits_{\Delta U_t \triangleq \{\delta u_{0,t},\ldots,\delta u_{N_u-1,t}\}} & \left\{\sum_{k=0}^{N_y-1} \|Q(y_k - r(t))\|_p + \|R\delta u_{k,t}\|_p\right\} \\ \text{subj. to} & y_{\min} \le y_k \le y_{\max},\ k = 1,\ldots,N_c \\ & u_{\min} \le u_{k,t} \le u_{\max},\ k = 0,1,\ldots,N_u \\ & \delta u_{\min} \le \delta u_{k,t} \le \delta u_{\max},\ k = 0,1,\ldots,N_u - 1 \\ & x_{k+1} = Ax_k + Bu_{k,t},\ k \ge 0 \\ & y_k = Cx_k,\ k \ge 0 \\ & u_{k,t} = u_{k-1,t} + \delta u_{k,t},\ k \ge 0 \\ & \delta u_{k,t} = 0,\ k \ge N_u \\ & x_0 = x(t) \end{array} \tag{4.29}$$

with

$$u(t) = \delta u^*_{0,t} + u(t-1). \tag{4.30}$$

Note that the δu-formulation (4.29) introduces m new states in the predictive model, namely the last input $u(t-1)$.

Just like the regulation problem, we can transform the tracking problem (4.29) into a multiparametric program. For example, for $p = 2$ the form

$$\begin{array}{ll} \min\limits_{\Delta U} & \frac{1}{2}\Delta U' H \Delta U + [x'(t)\ u'(t-1)\ r'(t)]\, F \Delta U \\ \text{subj. to} & G\Delta U \le W + E \begin{bmatrix} x(t) \\ u(t-1) \\ r(t) \end{bmatrix} \end{array} \tag{4.31}$$

results, where $r(t)$ lies in a given (possibly unbounded) polyhedral set. Thus, the same mp-QP algorithm can be used to obtain an explicit piecewise affine solution $\delta u(t) = F(x(t), u(t-1), r(t))$. In case the reference $r(t)$ is known in advance, one can replace $r(t)$ by $r(k)$ in (4.29) and similarly get a piecewise affine anticipative controller $\delta u(t) = F(x(t), u(t-1), r(t), \ldots, r(N_y - 1))$.

Disturbances

We distinguish between *measured* and *unmeasured* disturbances. Measured disturbances $v(t)$ can be included in the prediction model

$$x(k+1) = Ax(k) + Bu(k) + Vv(k) \tag{4.32}$$

where $v(k)$ is the prediction of the disturbance at time k based on the measured value $v(t)$. Usually $v(k)$ is a linear function of $v(t)$, for instance $v(k) \equiv v(t)$ where it is assumed that the disturbance is constant over the prediction horizon. Then $v(t)$ appears as a vector of additional parameters in the QP, and the piecewise affine control law becomes $u(t) = F(x(t), v(t))$. Alternatively, as for reference tracking, when $v(t)$ is known in advance one can replace $v(k)$ by the known $\bar{v}(k)$ in (4.32) and get an anticipative controller $\delta u(t) = F(x(t), u(t-1), \bar{v}(t), \ldots, \bar{v}(N_y - 1))$.

Unmeasured disturbances are usually modeled as the output of a linear system driven by white Gaussian noise. The state vector $x(t)$ of the linear prediction model (2.1) is augmented by including the state $x_n(t)$ of such a linear disturbance model, and the QP provides a control law of the form $\delta u(t) = F(x(t), x_n(t))$ within a certain range of states of the plant and of the disturbance model. Clearly, $x_n(t)$ must be estimated on line from output measurements by a linear observer.

Soft Constraints

State and output constraints can lead to feasibility problems. For example, a disturbance may push the output outside the feasible region where no allowed control input may exist which brings the output back in at the next time step. Therefore, in practice, the output constraints (2.2) are relaxed or softened [161] as $y_{\min} - M\varepsilon \le y(t) \le y_{\max} + M\varepsilon$, where $M \in \mathbb{R}^p$ is a constant vector ($M^i \ge 0$ is related to the "concern" for the violation of the i-th output constraint), and the term $\rho\varepsilon^2$ is added to the objective to penalize constraint violations (ρ is a suitably large scalar). The variable ϵ plays the role of an independent optimization variable in the QP and is adjoined to z. The solution $u(t) = F(x(t))$ is again a piecewise affine controller, which aims at keeping the states in the constrained region without ever running into feasibility problems.

Variable Constraints

The bounds $y_{\min}$, $y_{\max}$, $\delta u_{\min}$, $\delta u_{\max}$, $u_{\min}$, $u_{\max}$ may change depending on the operating conditions, or in the case of a stuck actuator the constraints become $\delta u_{\min} = \delta u_{\max} = 0$. This possibility can again be built into the control law. The bounds can be treated as parameters in the QP and added to the vector x. The control law will have the form $u(t) = F(x(t), y_{\min}, y_{\max}, \delta u_{\min}, \delta u_{\max}, u_{\min}, u_{\max})$.

5

Constrained Robust Optimal Control

For discrete-time uncertain linear systems with constraints on inputs and states, we develop an approach to determine the state feedback controller based on a min-max optimal control formulation based on a linear performance index. Robustness is achieved against additive norm-bounded input disturbances and/or polyhedral parametric uncertainties in the state-space matrices. We show that the robust optimal control law over a finite horizon is a continuous piecewise affine function of the state vector. Thus, when the optimal control law is implemented in a moving horizon scheme, the on-line computation of the resulting controller requires the evaluation of a piecewise affine function only.

5.1 Introduction

A control system is robust when stability is preserved and the performance specifications are met for a specified range of model variations and a class of noise signals (uncertainty range). Although a rich theory has been developed for the robust control of *linear systems*, very little is known about the robust control of *linear systems with constraints*. This type of problem has been addressed in the context of constrained optimal control, and in particular in the context of robust model predictive control (MPC) (see e.g. [112, 107]). A typical robust RHC/MPC strategy consists of solving a min-max problem to optimize robust performance (the minimum over the control input of the maximum over the disturbance) while enforcing input and state constraints for all possible disturbances. Min-max robust constrained optimal control was originally proposed by Witsenhausen [159]. In the context of robust MPC, the problem was tackled by Campo and Morari [45], and further developed in [3] for SISO FIR plants. Kothare *et al.* [101] optimize robust performance for polytopic/multi-model and linear fractional uncertainty, Scokaert and Mayne [133] for additive disturbances, and Lee and Yu [104] for linear time-varying and time-invariant state-space models depending on a vector of parameters $\theta \in \Theta$, where Θ is either an ellipsoid or a polyhedron. In all cases, the resulting min-max optimization problems turn out to be computationally demanding, a serious drawback for on-line receding horizon implementation.

In this chapter we show how state feedback solutions to min-max robust constrained optimal control problems based on a linear performance index can be computed off-line for systems affected by additive norm-bounded input disturbances and/or polyhedral parametric uncertainty. We show that the resulting optimal state feedback control law is affine so that on-line computation involves a simple function evaluation. The approach of this chapter relies on multiparametric solvers, and follows the ideas proposed in the previous chapters for the optimal control of linear systems without uncertainty.

5.2 Problem Formulation

Consider the following discrete-time linear uncertain system

$$x(t+1) = A(w(t))x(t) + B(w(t))u(t) + Ev(t) \tag{5.1}$$

subject to constraints

$$Fx(t) + Gu(t) \le f, \tag{5.2}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^{n_u}$ are the state and input vector, respectively. Vectors $v(t) \in \mathbb{R}^{n_v}$ and $w(t) \in \mathbb{R}^{n_w}$ are unknown exogenous disturbances and parametric uncertainties, respectively, and we assume that only bounds on $v(t)$ and $w(t)$ are known, namely that $v(t) \in \mathcal{V}$, where $\mathcal{V} \subset \mathbb{R}^{n_v}$ is a given polytope containing the origin , $\mathcal{V} = \{v : Lv \le \ell\}$, and that $w(t) \in \mathcal{W} = \{w : Mw \le m\}$, where $\mathcal{W}$ is a polytope in $\mathbb{R}^{n_w}$. We also assume that $A(\cdot)$, $B(\cdot)$ are affine functions of w, $A(w) = A_0 + \sum_{i=1}^q A_i w_i$, $B(w) = B_0 + \sum_{i=1}^q B_i w_i$, a rather general time-domain description of uncertainty (see e.g. [31]), which includes uncertain FIR models [45].

The following min-max optimal control problem will be referred to as Open-Loop Constrained Robust Optimal Control (OL-CROC) problem:

$$J_N^*(x_0) \triangleq \min_{u_0,\dots,u_{N-1}} J(x_0, U) \tag{5.3}$$

$$\text{subj. to} \quad \left\{ \begin{array}{l} Fx_k + Gu_k \le f \\ x_{k+1} = A(w_k)x_k + B(w_k)u_k + Ev_k \\ x_N \in \mathcal{X}_f \\ k = 0, \dots, N-1 \end{array} \right\} \begin{array}{l} \forall v_k \in \mathcal{V}, w_k \in \mathcal{W} \\ \forall k = 0, \dots, N-1 \end{array} \tag{5.4}$$

$$J(x_0, U) \triangleq \max_{\substack{v_0, \dots, v_{N-1} \\ w_0, \dots, w_{N-1}}} \left\{ \sum_{k=0}^{N-1} \|Qx_k\|_p + \|Ru_k\|_p + \|Px_N\|_p \right\} \tag{5.5}$$

$$\text{subj. to} \quad \left\{ \begin{array}{l} x_{k+1} = A(w_k)x_k + B(w_k)u_k + Ev_k \\ v_k \in \mathcal{V} \\ w_k \in \mathcal{W}, \\ k = 0, \dots, N-1 \end{array} \right. \tag{5.6}$$

where x_k denotes the state vector at time k, obtained by starting from the state $x_0 \triangleq x(0)$ and applying to model (5.1) the input sequence $U \triangleq \{u_0, \dots, u_{N-1}\}$ and the sequences $V \triangleq \{v_0, \dots, v_{N-1}\}$, $W \triangleq \{w_0, \dots, w_{N-1}\}$; $p = 1$ or $p = +\infty$, $\|x\|_\infty$ and $\|x\|_1$ are the standard ∞-norm and 1-norm in $\mathbb{R}^n$, $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{n_u \times n_u}$ are non-singular matrices, $P \in \mathbb{R}^{m \times n}$, and the constraint $x_N \in \mathcal{X}_f$ forces the final state x_N to belong to the polyhedral set

$$\mathcal{X}_f \triangleq \{x \in \mathbb{R}^n : \ F_N x \le f_N\}. \tag{5.7}$$

The choice of $\mathcal{X}_f$ is typically dictated by stability and feasibility requirements when (5.3)–(5.6) is implemented in a receding horizon fashion, as will be discussed later in Section 5.4. Problem (5.5)–(5.6) looks for the worst value $J(x_0, U)$ of the performance index and the corresponding worst sequences V, W as a function of x_0 and U, while problem (5.3)–(5.4) minimizes the worst performance subject to the constraint that the input sequence must be feasible *for all* possible disturbance realizations. In other words, worst-case performance is minimized under constraint fulfillment against all possible realizations of V, W.

In the sequel, we denote by $U^* = \{u_0^*, \ldots, u_{N-1}^*\}$ the optimal solution to (5.3)–(5.6) where $u_j^* : \mathbb{R}^n \mapsto \mathbb{R}^{n_u}$, $j = 0, \ldots, N-1$, and by $\mathcal{X}_0$ the set of initial states x_0 for which (5.3)–(5.6) is feasible.

The min-max formulation (5.3)–(5.6) is based on an *open-loop* prediction, in contrast to the *closed-loop* prediction schemes of [133, 104, 101, 102, 14]. In [102, 14] $u_k = Fx_k + \bar{u}_k$, where F is a constant linear feedback law, and $\bar{u}_k$ are new degrees of freedom optimized on-line. In [101] $u_k = Fx_k$, and F is optimized on-line via linear matrix inequalities. In [14] $u_k = Fx_k + \bar{u}_k$, where $\bar{u}_k$ and F are optimized on line (for implementation, F is restricted to belong to a finite set of LQR gains). In [133, 104] the optimization is over general feedback laws.

The benefits of closed-loop prediction can be understood by viewing the optimal control problem as a dynamic game between the disturbance and the input. In open-loop prediction the whole disturbance sequence plays first, then the input sequence is left with the duty of counteracting the worst disturbance realization. By letting the whole disturbance sequence play first, the effect of the uncertainty may grow over the prediction horizon and may easily lead to infeasibility of the min problem (5.3)–(5.4). On the contrary, in closed-loop prediction schemes the disturbance and the input play one move at a time, which makes the effect of the disturbance more easily mitigable [14].

In order not to impose any predefined structure on the closed-loop controller, we define the following Closed-Loop Constrained Robust Optimal Control (CL-CROC) problem [159, 111, 29, 104]:

$$J_j^*(x_j) \triangleq \min_{u_j} J_j(x_j, u_j) \tag{5.8}$$

$$\text{subj. to} \quad \left\{ \begin{array}{l} Fx_j + Gu_j \le f \\ A(w_j)x_j + B(w_j)u_j + Ev_j \in \mathcal{X}_{i+1} \end{array} \right\} \forall v_j \in \mathcal{V}, w_j \in \mathcal{W} \tag{5.9}$$

$$J_j(x_j, u_j) \triangleq \max_{v_j \in \mathcal{V},\ w_j \in \mathcal{W}} \left\{ \|Qx_j\|_p + \|Ru_j\|_p + J_{j+1}^*(A(w_j)x_j + B(w_j)u_j + Ev_j) \right\} \tag{5.10}$$

for $j = 0, \ldots, N-1$ and with boundary conditions

$$J_N^*(x_N) = \|Px_N\|_p \tag{5.11}$$

$$\mathcal{X}_N = \mathcal{X}_f, \tag{5.12}$$

where $\mathcal{X}_j$ denotes the set of states x for which (5.8)–(5.10) is feasible

$$\begin{aligned}\mathcal{X}_j = \{x \in \mathbb{R}^n | \; \exists u, \; (Fx + Gu \leq f, \text{ and} \\ \text{and } A(w)x + B(w)u + Ev \in \mathcal{X}_{j+1} \; \forall v \in \mathcal{V}, w \in \mathcal{W})\}.\end{aligned} \tag{5.13}$$

The reason for including constraints (5.9) in the minimization problem and not in the maximization problem is that in (5.10) v_j is free to act regardless of the state constraints. On the other hand, the input u_j has the duty of keeping the state within the constraints (5.9) for all possible disturbance realization.

We will consider different ways of solving OL-CROC and CL-CROC problems in the following sections. First we will briefly review other algorithms that were proposed in the literature.

For models affected by additive norm-bounded disturbances and parametric uncertainties on the impulse response coefficients, Campo and Morari [45] show how to solve the OL-CROC problem via linear programming. The idea can be summarized as follows. First, the minimization of the objective function (5.3) is replaced by the minimization of an upper-bound μ on the objective function subject to the constraint that μ is indeed an upper bound for all sequences $V \in \Omega^{\mathcal{V}}$ (although μ is an upper bound, at the optimum it coincides with the optimal value of the original problem). Then, by exploiting the convexity of the objective function (5.3) with respect to V, such a continuum of constraints is replaced by a finite number, namely one for each vertex of the set $\Omega^{\mathcal{V}}$. As a result, for a given value of the initial state $x(0)$, the OL-CROC problem is recast as a linear program (LP).

A solution to the CL-CROC problem was given in [133] using a similar convexity and vertex enumeration argument. The idea there is to augment the number of free inputs by allowing one free sequence U_i for each vertex i of the set $\mathcal{V} \times \mathcal{V} \times \ldots \times \mathcal{V}$, i.e. $N \cdot N_{\mathcal{V}}^N$ free control moves, where $N_{\mathcal{V}}$ is the number of vertices of the set $\mathcal{V}$. By using a causality argument, the number of such free control moves is decreased to $(N_{\mathcal{V}}^N - 1)/(N_{\mathcal{V}} - 1)$. Again, using the minimization of an upper-bound for all the vertices of $\mathcal{V} \times \mathcal{V} \times \ldots \times \mathcal{V}$, the problem is recast as a finite dimensional convex optimization problem, which in the case of ∞-norms or 1-norms, can be handled via linear programming as in [45] (see [98] for details). By reducing the number of degrees of freedom in the choice of the optimal input moves, other suboptimal CL-CROC strategies have been proposed, e.g., in [101, 14, 102].

5.3 State Feedback Solution to CROC Problems

In Section 5.2 we reviewed different approaches to compute the optimal input sequence solving the CROC problems for a given value of the initial state x_0. For a very general parameterization of the uncertainty description, in [104] the authors propose to solve CL-CROC in state feedback form via dynamic programming by discretizing the state-space. Therefore the technique is limited to simple low-dimensional prediction models. In this chapter we want to find a *state feedback* solution to CROC problems, namely an explicit function $u_k^*(x_k)$ mapping the state x_k to its corresponding optimal input u_k^*, $\forall k = 0, \dots, N-1$. We aim at finding the exact solution to CROC problems via multiparametric programming [68, 57, 25, 37], and in addition, for the CL-CROC problem, by using dynamic programming.

5.3.1 Preliminaries on Multiparametric Programming

Consider the multiparametric mixed-integer linear program (mp-MILP)

$$\begin{array}{rl} J^*(x) = \min_z & g'z \\ \text{s.t.} & Cz \leq c + Sx, \end{array} \tag{5.14}$$

where $z \triangleq [z_c, z_d]$, $z_c \in \mathbb{R}^{n_c}$, $z_d \in \{0,1\}^{n_d}$, $n_z \triangleq n_c + n_d$ is the optimization vector, $x \in \mathbb{R}^n$ is the vector of parameters, and $g \in \mathbb{R}^{n_z}$, $C \in \mathbb{R}^{q \times n_z}$, $c \in \mathbb{R}^q$, $S \in \mathbb{R}^{q \times n}$ are constant matrices. If in problem (5.14) there are no integer variables, i..e. $n_d = 0$, then (5.14) is a multiparametric linear program (mp-LP) (see Chapter 1) .

For a given polyhedral set $X \subseteq \mathbb{R}^n$ of parameters, solving (5.14) amounts to determining the set $X_f \subseteq X$ of parameters for which (5.14) is feasible, the value function $J : X_f \to \mathbb{R}$, and the optimizer function[1] $z^* : X_f \to \mathbb{R}^{n_z}$.

The properties of $J^*(\cdot)$ and $z^*(\cdot)$ have been analyzed in Chapter 1 and summarized in Theorems 1.8 and 1.16. Below we give some results based on these theorems.

Lemma 5.1. *Let $J : \mathbb{R}^{n_z} \times \mathbb{R}^n \to \mathbb{R}$ be a continuous piecewise affine (possibly nonconvex) function of (z, x),*

$$J(z, x) = L_i z + H_i x + K_i \;\; \text{for} \;\; \left[\begin{smallmatrix} z \\ x \end{smallmatrix}\right] \in \mathcal{R}_i, \tag{5.15}$$

where $\{\mathcal{R}_i\}_{i=1}^s$ are polyhedral sets with disjoint interiors, $\mathcal{R} \triangleq \bigcup_{i=1}^s \mathcal{R}_i$ is a (possibly non-convex) polyhedral set and L_i, H_i and K_i are matrices of suitable dimensions. Then the multiparametric optimization problem

$$\begin{array}{rl} J^*(x) \triangleq \min_z & J(z, x) \\ \text{subj. to} & Cz \leq c + Sx. \end{array} \tag{5.16}$$

is an mp-MILP.

Proof: By following the approach of [23] to transform piecewise affine functions into a set of mixed-integer linear inequalities, introduce the auxiliary binary optimization variables $\delta_i \in \{0, 1\}$, defined as

$$[\delta_i = 1] \;\leftrightarrow\; \left[\left[\begin{smallmatrix} z \\ x \end{smallmatrix}\right] \in \mathcal{R}_i\right], \tag{5.17}$$

[1] In case of multiple solutions, we define $z^*(x)$ as one of the optimizers.

where $\delta_i,\ i = 1,\dots,s$, satisfy the exclusive-or condition $\sum_{i=1}^{s} \delta_i = 1$, and set

$$J(z,x) = \sum_{i=1}^{s} q_i \tag{5.18}$$

$$q_i \triangleq [L_i z + H_i x + K_i]\delta_i \tag{5.19}$$

where q_i are auxiliary continuous optimization vectors. By transforming (5.17)–(5.19) into mixed-integer linear inequalities [23], it is easy to rewrite (5.16) as a multiparametric MILP. □

The following lemma deals with the special case where J is a convex function of z and x (i.e., $\mathcal{R} \triangleq \cup_{i=1}^{s} \mathcal{R}_i$ is a convex set and J is convex over $\mathcal{R}$).

Lemma 5.2. *Let $J : \mathbb{R}^{n_z} \times \mathbb{R}^n \to \mathbb{R}$ be a convex piecewise affine function of (z,x). Then the multiparametric optimization problem (5.16) is an mp-LP.*

Proof: As J is a convex piecewise affine function, it follows that $J(z,x) = \max_{i=1,\dots,s} \{L_i z + H_i x + K_i\}$ [132]. Then, it is easy to show that (5.16) is equivalent to the following mp-LP: $\min_{z,\varepsilon} \varepsilon$ subject to $Cz \le c + Sx$, $L_i z + H_i x + K_i \le \varepsilon$, $i = 1,\dots,s$. □

Lemma 5.3. *Let $f : \mathbb{R}^{n_z} \times \mathbb{R}^n \times \mathbb{R}^{n_d} \to \mathbb{R}$ and $g : \mathbb{R}^{n_z} \times \mathbb{R}^n \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_g}$ be functions of (z,x,d) convex in d for each (z,x). Assume that the variable d belongs to the polyhedron $\mathcal{D}$ with vertices $\{\bar{d}_i\}_{i=1}^{N_\mathcal{D}}$. Then, the min-max multiparametric problem*

$$\begin{array}{rl} J^*(x) = \min_z & \max_{d\in\mathcal{D}} f(z,x,d) \\ & \textit{subj. to } g(z,x,d) \le 0\ \forall d \in \mathcal{D} \end{array} \tag{5.20}$$

is equivalent to the multiparametric optimization problem

$$\begin{array}{rl} J^*(x) = \min_{\mu,z} & \mu \\ & \textit{subj. to } \mu \ge f(z,x,\bar{d}_i),\ i = 1,\dots,N_\mathcal{D} \\ & g(z,x,\bar{d}_i) \le 0,\ i = 1,\dots,N_\mathcal{D}. \end{array} \tag{5.21}$$

Proof: Easily follows by the fact that the maximum of a convex function over a convex set is attained at an extreme point of the set, cf. also [133]. □
□

Corollary 5.1. *If f is also convex and piecewise affine in (z,x), i.e. $f(z,x,d) = \max_{i=1,\dots,s} \{L_i(d)z + H_i(d)x + K_i(d)\}$ and g is linear in (z,x) for all $d \in \mathcal{D}$, $g(z,x,d) = K_g(d) + L_g(d)x + H_g(d)z$ (with $K_g(\cdot)$, $L_g(\cdot)$, $H_g(\cdot)$, $L_i(\cdot)$, $H_i(\cdot)$, $K_i(\cdot)$, $i = 1,\dots,s$, convex functions), then the min-max multiparametric problem (5.20) is equivalent to the mp-LP problem*

$$\begin{array}{rl} J^*(x) = min_{\mu,z} & \mu \\ & \textit{subj. to } \mu \ge K_j(\bar{d}_i) + L_j(\bar{d}_i)z + H_j(\bar{d}_i)x,\ i = 1,\dots,N_\mathcal{D},\ j = 1,\dots,s \\ & L_g(\bar{d}_i)x + H_g(\bar{d}_i)z \le -K_g(\bar{d}_i),\ i = 1,\dots,N_\mathcal{D} \end{array} \tag{5.22}$$

Remark 5.1. In case $g(x,z,d) = g_1(x,z)+g_2(d)$, the second constraint in (5.21) can be replaced by $g(z,x) \leq -\bar{g}$, where $\bar{g} \triangleq \left[\bar{g}_1,\dots,\bar{g}_{n_g}\right]'$ is a vector whose i-th component is

$$\bar{g}_i = \max_{d\in\mathcal{D}} \; g_2^i(d), \tag{5.23}$$

and $g_2^i(d)$ denotes the i-th component of $g_2(d)$. Similarly, if $f(x,z,d) = f_1(x,z)+f_2(d)$, the first constraint in (5.21) can be replaced by $f(z,x) \leq -\bar{f}$, where

$$\bar{f}_i = \max_{d\in\mathcal{D}} \; f_2^i(d). \tag{5.24}$$

Clearly, this has the advantage of reducing the number of constraints in the multiparametric program from $N_{\mathcal{D}}n_g$ to n_g for the second constraint in (5.21) and from $N_{\mathcal{D}}s$ to s for the first constraint in (5.21).

In the following subsections we propose two different approaches to solve CROC problems in state feedback form, the first one is based on multiparametric linear programming and the second one based on multiparametric mixed-integer linear programming.

5.3.2 Closed Loop CROC

Theorem 5.1. *By solving N mp-LPs, the solution of CL-CROC (5.8)–(5.12) is obtained in state feedback piecewise affine form*

$$u_k^*(x(k)) = F_k^i x(k) + g_k^i, \;\; \text{if } x(k) \in CR_k^i, \;\; i = 1,\dots,N_k^r \tag{5.25}$$

where the sets CR_k^i $i = 1,\dots,N_k^r$ are a polyhedral partition of the set of feasible states $\mathcal{X}_k$ at time k.

Proof: Consider the first step $j = N-1$ of dynamic programming applied to the CL-CROC problem (5.8)–(5.10)

$$J_{N-1}^*(x_{N-1}) \triangleq \min_{u_{N-1}} J_{N-1}(x_{N-1},u_{N-1}) \tag{5.26}$$

$$\text{subj. to} \; \begin{cases} Fx_{N-1} + Gu_{N-1} \leq f \\ A(w_{N-1})x_{N-1} + B(w_{N-1})u_{N-1} + Ev_{N-1} \in \mathcal{X}_f \\ \forall v_{N-1} \in \mathcal{V}, w_{N-1} \in \mathcal{W} \end{cases} \tag{5.27}$$

$$J_{N-1}(x_{N-1},u_{N-1}) \triangleq \max_{v_{N-1}\in\mathcal{V},\; w_{N-1}\in\mathcal{W}} \left\{ \begin{array}{l} \|Qx_{N-1}\|_p + \|Ru_{N-1}\|_p + \\ +\|P(A(w_{N-1})x_{N-1} + \\ +B(w_{N-1})u_{N-1} + Ev_{N-1})\|_p \end{array} \right\}. \tag{5.28}$$

The cost function in the maximization problem (5.28) is piecewise affine and convex with respect to the optimization vector v_{N-1}, w_{N-1} and the parameters u_{N-1}, x_{N-1}. Moreover, the constraints in the minimization problem

(5.27) are linear in (u_{N-1}, x_{N-1}) for all vectors $v_{N-1},\ w_{N-1}$. Therefore, by Corollary 5.1, $J^*_{N-1}(x_{N-1})$, $u^*_{N-1}(x_{N-1})$ and $\mathcal{X}_{N-1}$ are computable via the mp-LP:

$$J^*_{N-1}(x_{N-1}) \triangleq \min_{\mu, u_{N-1}} \ \mu \tag{5.29a}$$

$$\text{s.t.} \ \ \mu \geq \|Qx_{N-1}\|_p + \|Ru_{N-1}\|_p + + \|P(A(\bar{w}_h)x_{N-1} + B(\bar{w}_h)u_{N-1} + E\bar{v}_i)\|_p \tag{5.29b}$$

$$Fx_{N-1} + Gu_{N-1} \leq f \tag{5.29c}$$

$$A(\bar{w}_h)x_{N-1} + B(\bar{w}_h)u_{N-1} + E\bar{v}_i \in \mathcal{X}_N \tag{5.29d}$$

$$\forall i = 1, \ldots, N_{\mathcal{V}},\ \forall h = 1, \ldots, N_{\mathcal{W}}.$$

where $\{\bar{v}_i\}_{i=1}^{N_{\mathcal{V}}}$ and $\{\bar{w}_h\}_{h=1}^{N_{\mathcal{W}}}$ are the vertices of the disturbance sets $\mathcal{V}$ and $\mathcal{W}$, respectively. By Theorem 1.8, J^*_{N-1} is a convex and piecewise affine function of x_{N-1}, the corresponding optimizer u^*_{N-1} is piecewise affine and continuous, and the feasible set $\mathcal{X}_{N-1}$ is a convex polyhedron. Therefore, the convexity and linearity arguments still hold for $j = N-2, \ldots, 0$ and the procedure can be iterated backwards in time j, proving the theorem. □

Remark 5.2. Let n_a and n_b be the number of inequalities in (5.29b) and (5.29d), respectively, for any i and h. In case of additive disturbances only ($w(t) = 0$) the total number of constraints in (5.29b) and (5.29d) for all i and h can be reduced from $(n_a + n_b)N_{\mathcal{V}}N_{\mathcal{W}}$ to $n_a + n_b$ as shown in Remark 5.1.

The following corollary is an immediately consequence of the continuity properties of the mp-LP recalled in Theorem 1.8, and from Theorem 5.1:

Corollary 5.2. *The piecewise affine solution $u^*_k : \mathbb{R}^n \to \mathbb{R}^{n_u}$ to the CL-CROC problem is a continuous function of x_k, $\forall k = 0, \ldots, N-1$.*

5.3.3 Open Loop CROC

Theorem 5.2. *The solution $U^* : \mathcal{X}_0 \to \mathbb{R}^{Nn_u}$ to OL-CROC with parametric uncertainties in the B matrix only ($A(w) \equiv A$), is a piecewise affine function of $x_0 \in \mathcal{X}_0$, where $\mathcal{X}_0$ is the set of initial states for which a solution to (5.3)–(5.6) exists. It can be found by solving an mp-LP.*

Proof: Since $x_k = A^k x_0 + \sum_{k=0}^{k-1} A^i[B(w)u_{k-1-i} + Ev_{k-1-i}]$ is a linear function of the disturbances W, V for any given U and x_0, the cost function in the maximization problem (5.5) is convex and piecewise affine with respect to the optimization vectors V, W and the parameters U, x_0. The constraints in (5.3) are linear in U and x_0, for any V and W. Therefore, by Lemma 5.3, problem (5.3)–(5.6) can be solved by solving an mp-LP through the enumeration of all the vertices of the sets $\mathcal{V} \times \mathcal{V} \times \ldots \times \mathcal{V}$ and $\mathcal{W} \times \mathcal{W} \times \ldots \times \mathcal{W}$. □

We remark that Theorem 5.2 covers a rather broad class of uncertainty descriptions, including uncertainty on the coefficients of the impulse and step response [45]. In case of OL-CROC with additive disturbances only ($w(t) = 0$) the number of constraints in (5.4) can be reduced as explained in Remark 5.1.

Remark 5.3. While in the mp-LP problem for OL-CROC given by Theorem 5.2 the number of optimization variables is proportional to N, in the mp-LP problem resulting from the CL-CROC approach of [133, 98] it is proportional to $N_{\mathcal{V}} \cdot \max\{N, n_v^N\}$. The larger number of degrees of freedom in CL-CROC is another way of explaining its superior performance with respect to OL-CROC.

The following is a corollary of the continuity properties of mp-LP recalled in Theorem 1.8 and of Theorem 5.2:

Corollary 5.3. *The piecewise affine solution $U^* : \mathcal{X}_0 \to \mathbb{R}^{Nn_u}$ to the OL-CROC problem with additive disturbances and uncertainty in the B matrix only ($A(w) \equiv A$) is a continuous function of x_0.*

5.3.4 Solution to CL-CROC and OL-CROC via mp-MILP

Theorem 5.3. *By solving 2N mp-MILPs, the solution of the CL-CROC (5.8)–(5.12) problem with additive disturbances only ($w(t) = 0$) can be obtained in state feedback piecewise affine form (5.25)*

Proof: Consider the first step $j = N - 1$ of the dynamic programming solution (5.8)–(5.10) to CL-CROC. From the terminal conditions (5.12) it follows that the cost function in the maximization problem is piecewise affine with respect to both the optimization vector v_{N-1} and the parameters u_{N-1}, x_{N-1}. By Lemma 5.1, (5.8)-(5.10) can be computed via mp-MILP, and, by Theorem 1.16, it turns out that $J_{N-1}(u_{N-1}, x_{N-1})$ is a piecewise affine and continuous function. Then, since constraints (5.27) are linear with respect to v_{N-1} for each u_{N-1}, x_{N-1}, we can apply Lemma 5.3 by solving LPs of the form (5.23). Then, by Lemma 5.1, $J^*_{N-1}(x_{N-1})$ is again computable via mp-MILP, and by Theorem 1.16 it is a piecewise affine and continuous function of x_{N-1}. By virtue of Theorem 1.16, $\mathcal{X}_{N-1}$ is a (possible non-convex) polyhedral set and therefore the above maximization and minimization procedures can be iterated to compute the solution (5.25) to the CL-CROC problem. □

Theorem 5.4. *By solving two mp-MILPs, the solution $U^*(x_0) : \mathcal{X}_0 \to \mathbb{R}^{Nn_u}$ to OL-CROC with additive disturbances only ($w(t) = 0$) can be computed in explicit piecewise affine form, where $\mathcal{X}_0$ is the set of states for which a solution to (5.3)–(5.6) exists.*

Proof: The objective function in the maximization problem (5.5) is convex and piecewise affine with respect to the optimization vector $V = \{v_0, \ldots, v_{N-1}\}$ and the parameters $U = \{u_0, \ldots, u_{N-1}\}$, x_0. By Lemma 5.1, it can be solved via mp-MILP. By Theorem 1.16, the value function J is a piecewise affine function of U and x_0 and the constraints in (5.3) are a linear function of the disturbance V for any given U and x_0. Then, by Lemma 5.1 and Lemma 5.3 the minimization problem is again solvable via mp-MILP, and the optimizer $U^* = \{u_0^*, \ldots, u_{N-1}^*\}$ is a piecewise affine function of x_0. □

Remark 5.4. Theorems 5.3, 5.4 and Theorems 5.1, 5.2 propose two different ways of finding the PPWA solution to constrained robust optimal control by using dynamic programming. The solution approach of Theorems 5.3, 5.4 is more general than the one of Theorems 5.1, 5.2 as it does not exploit convexity, so that it may be also used in other contexts, for instance in CL-CROC of hybrid systems.

5.4 Robust Receding Horizon Control

A robust receding horizon controller for system (5.1) which enforces the constraints (5.2) at each time t in spite of additive and parametric uncertainties can be obtained immediately by setting

$$u(t) = u_0^*(x(t)), \tag{5.30}$$

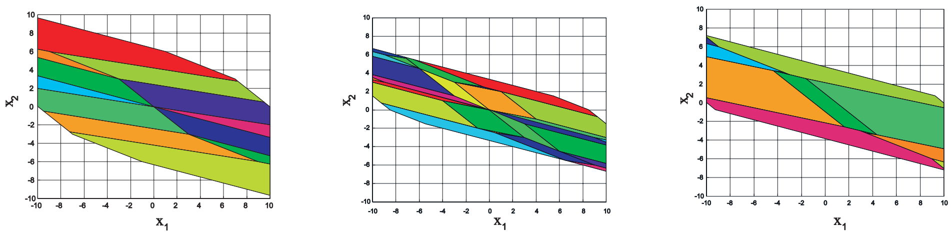
where $u_0^*(x_0) : \mathbb{R}^n \to \mathbb{R}^{n_u}$ is the piecewise affine solution to the OL-CROC or CL-CROC problems developed in the previous sections. In this way we obtain a state feedback strategy defined at all time steps $t = 0, 1, \ldots$, from the associated finite time CROC problem.

For stability and feasibility of the closed-loop system (5.1), (5.30) we refer the reader to previously published results on robust MPC, see e.g. [24, 107, 112], although general stability criteria for the case of parametric uncertainties are still an open problem.

When the optimal control law is implemented in a moving horizon scheme, the on-line computation consists of a simple function evaluation. However, when the number of constraints involved in the optimization problem increases, the number of regions associated with the piecewise affine control mapping may increase exponentially. In [35, 149] efficient algorithms for the on-line evaluation of the piecewise affine optimal control law were presented, where efficiency is in terms of storage demand and computational complexity.

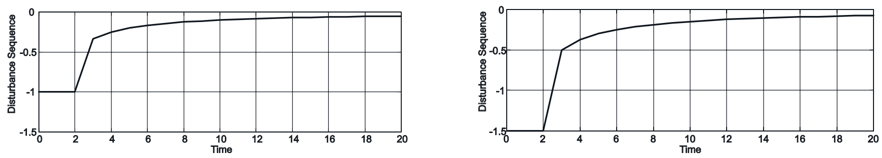
5.5 Examples

In [17] we compared the state feedback solutions to nominal MPC [18], open-loop robust MPC, and closed-loop robust MPC for the example considered in [133], using infinity norms instead of quadratic norms in the objective function. For closed-loop robust MPC, the off-line computation time in Matlab 5.3 on a Pentium III 800 was about 55 s by using the approach of Theorem 5.3 (mp-MILP), and 1.27 s by using Theorem 5.1 (mp-LP). Below we consider another example.



(a) Nominal Optimal Control (b) OL-CROC (c) CL-CROC

Fig. 5.1. Polyhedral partition of the state-space corresponding to the explicit solution of nominal optimal control, OL-CROC and CL-CROC



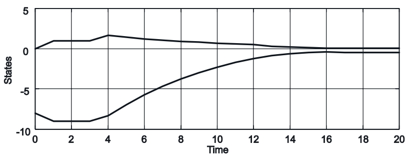
(a) Disturbance profile #1 (b) Disturbance profile #2

Fig. 5.2. Disturbances profiles

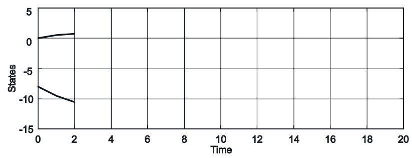
Example 5.1. Consider the problem of robustly regulating to the origin the system $x(t+1) = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right] x(t) + \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right] u(t) + \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right] v(t)$. We consider the performance measure $\|Px_N\|_\infty + \sum_{k=0}^{N-1}(\|Qx_k\|_\infty + |Ru_k|)$, where $N = 4$, $P = Q = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 1.8$, and $U = \{u_0, \ldots, u_3\}$, subject to the input constraints $-3 \leq u_k \leq 3$, $k = 0, \ldots, 3$, and the state constraints $-10 \leq x_k \leq 10$, $k = 0, \ldots, 3$ ($\mathcal{X}_f = \mathbb{R}^2$). The two-dimensional disturbance v is restricted to the set $\mathcal{V} = \{v : \|v\|_\infty \leq 1.5\}$.

We compare the control law (5.30) for the nominal case, OL-CROC, and CL-CROC.

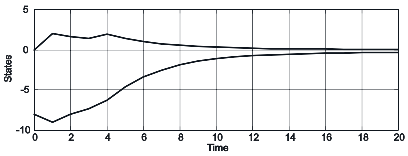
Nominal case. We ignore the disturbance $v(t)$, and solve the resulting multiparametric linear program by using the approach of [18]. The state feedback piecewise affine control law is computed in 23 s, and the corresponding polyhedral partition is depicted in Figure 5.1(a) (for lack of space, we do not report here the different affine gains for each region). The closed-loop system is simulated from the initial state $x(0) = [-8, 0]$ by applying two different disturbances profiles shown in Figure 5.2. Figures 5.3(a)-5.3(b) report the corresponding evolutions of the state vector. Note that the second disturbance profile leads to infeasibility at step 3.



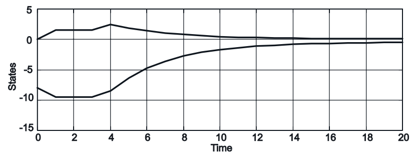
(a) nominal MPC, disturbance profile #1



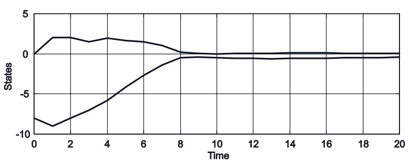
(b) nominal MPC, disturbance profile #2



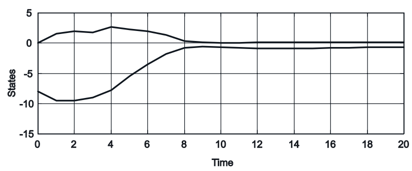
(c) Open-Loop Robust MPC, disturbance profile #1



(d) Open-Loop Robust MPC, disturbance profile #2



(e) Closed-Loop Robust MPC, disturbance profile #1



(f) Closed-Loop Robust MPC, disturbance profile #2

Fig. 5.3. Closed-loop simulations of nominal optimal control, open-loop robust MPC and closed-loop robust MPC

OL-CROC. The min-max problem is formulated as in (5.3)–(5.6) and solved off-line in 582 s. The resulting polyhedral partition is depicted in Figure 5.1(b). In Figures 5.3(c)- 5.3(d) the closed-loop system is simulated from the initial state $x(0) = [-8, 0]$ by applying the disturbances profiles in Figure 5.2.

CL-CROC. The min-max problem is formulated as in (5.8)–(5.10) and solved in 53 s using the approach of Theorem 5.1. The resulting polyhedral partition is depicted in Figure 5.1(c). In Figures 5.3(e)-5.3(f) the closed-loop system is again simulated from the initial state $x(0) = [-8, 0]$ by applying the disturbances profiles in Figure 5.2.

Remark 5.5. As shown in [98], the approach of [133] to solve CL-CROC, requires the solution of one mp-LP where the number of constraints is proportional to the number $N_{\mathcal{V}}^N$ of extreme points of the set $\mathcal{V} \times \mathcal{V} \times \ldots \times \mathcal{V} \subset \mathbb{R}^{Nn_v}$

of disturbance sequences, and the number of optimization variables is proportional to $N_{\mathcal{V}} \cdot \max\{N, n_v^N\}$, where $N_{\mathcal{V}}$ is the number of vertices of $\mathcal{V}$.

Let $n_{J_i^*}$ and $n_{\mathcal{X}_i}$ be the number of the affine gains of the cost-to-go function J_i^* and the number of constraints defining $\mathcal{X}_i$, respectively. The dynamic programming approach of Theorem 5.1 requires N mp-LPs where at step i the number of optimization variables is $n_u + 1$ and the number of constraints is equal to a quantity proportional to $(n_{J_i^*} + n_{\mathcal{X}_i})$. Usually $n_{J_i^*}$ and $n_{\mathcal{X}_i}$ do not increase exponentially during the recursion $i = N-1, \ldots, 0$ (even if, in the worst case, they could). For instance in Example 5.1, we have at step 0 $n_{J_0^*} = 34$ and $n_{\mathcal{X}_0} = 4$ while $N_{\mathcal{V}} = 4$ and $N_{\mathcal{V}}^N = 256$. As the complexity of an mp-LP depends mostly (in general combinatorially) on the number of constraints, one can expect that the approach presented here is numerically more efficient than the approach of [133, 98].

We remark that the off-line computational time of CL-CROC is about ten times smaller than the one of OL-CROC, where the vertex enumeration would lead to a problem with 12288 constraints, reduced to 38 after removing redundant inequalities in the extended space of variables and parameters. We finally remark that by enlarging the disturbance v to the set $\tilde{\mathcal{V}} = \{v : \|v\|_\infty \leq 2\}$ the OL-RMPC problem becomes infeasible for all the initial states, while the CL-RMPC problem is still feasible for a certain set of initial states.

Example 5.2. We consider here the problem of robustly regulating to the origin the active suspension system [89] $x(t+1) = \begin{bmatrix} 0.809 & 0.009 & 0 & 0 \\ -36.93 & 0.80 & 0 & 0 \\ 0.191 & -0.009 & 1 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0.0005 \\ 0.0935 \\ -0.005 \\ -0.0100 \end{bmatrix} u(t) + \begin{bmatrix} -0.009 \\ 0.191 \\ -0.0006 \\ 0 \end{bmatrix} v(t)$ where the input disturbance $v(t)$ represents the vertical ground velocity of the road profile and $u(t)$ the vertical acceleration. We solved the CL-CROC (5.8)–(5.10) with $N = 4$, $P = Q = \text{diag}\{5000, 0.1, 400, 0.1\}$, $\mathcal{X}^f = \mathbb{R}^4$, and $R = 1.8$, with input constraints $-5 \leq u \leq 5$, and the state constraints $\begin{bmatrix} -0.02 \\ -\infty \\ -0.05 \\ -\infty \end{bmatrix} \leq x \leq \begin{bmatrix} 0.02 \\ +\infty \\ 0.05 \\ +\infty \end{bmatrix}$. The disturbance v is restricted to the set $-0.4 \leq v \leq 0.4$. The problem was solved in less then 5 minutes for the subset $X = \left\{ x \in \mathbb{R}^4 | \begin{bmatrix} -0.02 \\ -1 \\ -0.05 \\ -0.5 \end{bmatrix} \leq x \leq \begin{bmatrix} 0.02 \\ 1 \\ 0.50 \\ 0.5 \end{bmatrix} \right\}$ of states, and the resulting piecewise-affine robust optimal control law is defined over 390 polyhedral regions.

6

Reducing On-line Complexity

By exploiting the properties of the value function and the piecewise affine optimal control law of constrained finite time optimal control (CFTOC) problem, we propose two new algorithms that avoid storing the polyhedral regions. The new algorithms significantly reduce the on-line storage demands and computational complexity during evaluation of the PPWA feedback control law resulting from the CFTOC problem.

6.1 Introduction

In Chapter 2 we have shown how to compute the solution to the constrained finite-time optimal control (CFTOC) problem as an explicit piecewise affine function of the initial state. Such a function is computed off-line by using a multiparametric program solver [25, 37], which divides the state space into polyhedral regions, and for each region determines the linear gain and offset which produces the optimal control action.

This method reveals its effectiveness when applied to *Receding Horizon Control* (RHC). Having a precomputed solution as an explicit piecewise affine on polyhedra (PPWA) function of the state vector reduces the on-line computation of the MPC control law to a function evaluation, therefore avoiding the on-line solution of a quadratic or linear program.

The only drawback of such explicit optimal control law is that the number of polyhedral regions could grow dramatically with the number of constraints in the optimal control problem. In this chapter we focus on efficient on-line methods for the evaluation of such a piecewise affine control law. The simplest algorithm would require: (i) the storage of the list of polyhedral regions and of the corresponding affine control laws, (ii) a sequential search through the list of polyhedra for the i-th polyhedron that contains the current state in order to implement the i-th control law.

By exploiting the properties of the value function and the optimal control law for CFTOC problem with $1,\infty$-norm and 2-norm we propose two new algorithms that avoid storing the polyhedral regions. The new algorithms significantly reduce the on-line storage demands and computational complexity during evaluation of the explicit solution of CFTOC problem.

6.2 Efficient On-Line Algorithms

Let the explicit optimal control law be:

$$u^*(x) = F^i x + g^i, \quad \forall x \in \mathcal{P}_i, \quad i = 1, \dots, N^r \tag{6.1}$$

where $F^i \in \mathbb{R}^{m\times n}$, $g^i \in \mathbb{R}^m$, and $\mathcal{P}_i = \left\{ x \in \mathbb{R}^n \mid H^i x \le K^i,\ H^i \in \mathbb{R}^{N_c^i \times n},\ K^i \in \mathbb{R}^{N_c^i} \right\},\ i = 1, \dots, N^r$ is a polyhedral partition of $\mathcal{X}$. In the following H_j^i denotes the j-row of the matrix

H^i and N_c^i is the numbers of constraints defining the i-th polyhedron. The on-line implementation of the control law (6.1) is simply executed according to the following steps:

Algorithm 6.2.1

```
1   Measure the current state x
2   Search for the j-th polyhedron that contains x, (H^j x ≤ K^j)
3   Implement the j-th control law (u(t) = F^j x + g^j)
```

In Algorithm 6.2.1, step (2) is critical and it is the only step whose efficiency can be improved. A simple implementation of step (2) would consist of searching for the polyhedral region that contains the state x as in the following algorithm:

Algorithm 6.2.2

Input: Current state x and polyhedral partion $\{\mathcal{P}_i\}_{i=1}^{N^r}$ of the control law (6.1)

Output: Index j of the polyhedron $\mathcal{P}_j$ in the control law (6.1) containing the current state x

```
i = 0, notfound=1;
while i ≤ N^r and notfound
    j = 0, stillfeasible=1
    while j ≤ N_c^i and stillfeasible
        if H_j^i x > K_j^i then stillfeasible=0
        else j = j + 1
    end
    if stillfeasible=1 then notfound=0
end
```

In Algorithm 6.2.2 H_j^i denotes the j-row of the matrix H^i, K_j^i denotes the j-th element of the vector K^i and N_c^i is the number of constraints defining the i-th polyhedron $\mathcal{P}_i$. Algorithm 6.2.2 requires the storage of all polyhedra $\mathcal{P}_i$, i.e., $(n+1)N_C$ real numbers (n numbers for each row of the matrix H^i plus one number for the corresponding element in the matrix K^i), $N_C \triangleq \sum_{i=1}^{N^r} N_c^i$, and in the worst case (the state is contained in the last region of the list) it will give a solution after nN_C multiplications, $(n-1)N_C$ sums and N_C comparisons.

Remark 6.1. Note that Algorithm 6.2.2 can also deduce if the point x is not inside of the feasible set $\mathcal{X}_0 = \cup_{i=1}^{N^r}\mathcal{P}_i$. In the following sections we implicitly assume that x belongs to $\mathcal{X}_0$. If this (reasonable) assumptions does not hold, it is always possible to include set of *boundaries* of feasible parameter space

$\mathcal{X}_0$. Then, before using any of proposed algorithms, we should first check if the point x is inside the boundaries of $\mathcal{X}_0$.

By using the properties of the value function, we will show how Algorithm 6.2.2 can be replaced by more efficient algorithms that have a smaller computational complexity and that *avoid storing the polyhedral regions* $\mathcal{P}_i,\ i = 1, \ldots, N^r$, therefore reducing significantly the storage demand.

In the following we will distinguish between optimal control based on LP and optimal control based on QP.

6.2.1 Efficient Implementation, $1, \infty$-Norm Case

From Theorem 2.4, the value function $J^*(x)$ corresponding to the solution of the CFTOC problem (4.28) with $1, \infty$-norm is convex and PWA:

$$J^*(x) = T^{i\prime} x + V^i, \quad \forall x \in \mathcal{P}_i, \quad i = 1, \ldots, N^r. \tag{6.2}$$

By exploiting the convexity of the value function the storage of the polyhedral regions $\mathcal{P}_i$ can be avoided. From the equivalence of the representations of PWA convex functions [132], the function $J^*(x)$ in equation (6.2) can be represented alternatively as

$$J^*(x) = \max \left\{ T^{i\prime} x + V^i,\ i = 1, \ldots, N^r \right\} \text{ for } x \in \mathcal{X}_0 = \cup_{i=1}^{N^r} \mathcal{P}_i. \tag{6.3}$$

Thus, the polyhedral region P_j containing x can be simply identified by searching for the maximum number in the list $\{T^{i\prime} x + V^i\}_{i=1}^{N^r}$:

$$x \in \mathcal{P}_j \Leftrightarrow T^{j\prime} x + V^j = \max \left\{ T^{i\prime} x + V^i,\ i = 1, \ldots, N^r \right\}. \tag{6.4}$$

Therefore, instead of searching for the polyhedron j that contains the point x via Algorithm 6.2.2, we can just store the value function and identify region j by searching for the maximum in the list of numbers composed of the single affine function $T^{i\prime} x + V^i$ evaluated at x (see Figure 6.1):

Algorithm 6.2.3

Input: Current state x and value function 6.2
Output: Index j of the polyhedron $\mathcal{P}_j$ containing the current state x in the control law (6.1)

1 Compute the list $\mathcal{L} = \{n_i \triangleq T^{i\prime} x + V^i,\ i = 1, \ldots, N^r\}$
2 Find j such that $n_j = \max_{n_i \in \mathcal{L}} n_i$

Algorithm 6.2.3 requires the storage of $(n + 1)N^r$ real numbers and will give a solution after nN^r multiplications, $(n - 1)N^r$ sums, and $N^r - 1$ comparisons. In Table 6.1 we compare the complexity of Algorithm 6.2.3 against Algorithm 6.2.2 in terms of storage demand and number of flops.

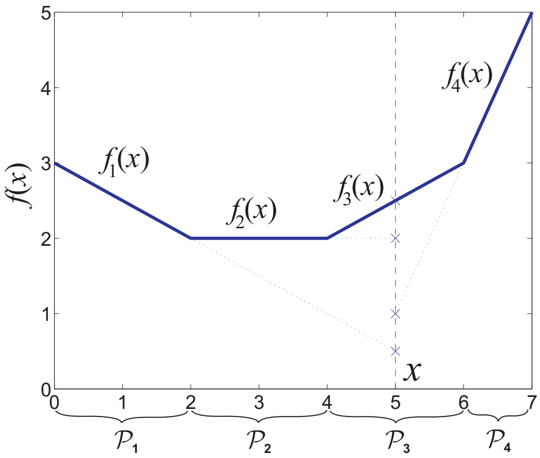


Fig. 6.1. Example for Algorithm 6.2.3 in one dimension: For a given point $x \in \mathcal{P}_3$ $(x = 5)$ we have $f_3(x) = \max(f_1(x), f_2(x), f_3(x), f_4(x))$.

Table 6.1. Complexity comparison of Algorithm 6.2.2 and Algorithm 6.2.3

	Algorithm 6.2.2	Algorithm 6.2.3
Storage demand (real numbers)	$(n+1)N_C$	$(n+1)N^r$
Number of flops (worst case)	$2nN_C$	$2nN^r$

Remark 6.2. Algorithm 6.2.3 will outperform Algorithm 6.2.2 since typically $N_C \gg N^r$.

6.2.2 Efficient Implementation, 2-Norm Case

Consider the state feedback solution (2.16) of the CFTOC problem (2.4) with $p = 2$. Theorem 2.2 states that the value function $J^*(x)$ is convex and piecewise quadratic on polyhedra and the simple Algorithm 6.2.3 described in the previous subsection cannot be used here. Instead, a different approach is described below. It uses a surrogate of the value function to uniquely characterize the polyhedral partition of the optimal control law.

We will first establish the following general result: given a general polyhedral partition of the state space, we can locate where the state lies (i.e., in which polyhedron) by using a search procedure based on the information provided by an "appropriate" PWA continuous function defined over the same polyhedral partition. We will refer to such an "appropriate" PWA function as a *PWA descriptor function*. First we outline the properties of the PWA descriptor function and then we describe the search procedure itself.

Let $\{\mathcal{P}_i\}_{i=1}^{N^r}$ be the polyhedral partition obtained by solving the mp-QP (2.18) and denote by $C_i = \{j \mid \mathcal{P}_j$ is a neighbor of $\mathcal{P}_i,\ j = 1, \ldots, N^r, j \neq i\}$ the list of all neighboring polyhedra of $\mathcal{P}_i$. The list C_i has N_c^i elements and we denote by $C_i(k)$ its k-th element. We give the following definition of PWA descriptor function.

Definition 6.1 (PWA descriptor function). *A scalar continuous real-valued PWA function*

$$f(x) = f_i(x) \triangleq A^{i\prime} x + B^i, \text{ if } x \in \mathcal{P}_i, \tag{6.5}$$

is called descriptor function if

$$A^i \neq A^j,\ \forall j \in C_i,\ i = 1, \ldots, N_r. \tag{6.6}$$

Theorem 6.1. *Let $f(x)$ be a PWA descriptor function on the polyhedral partition $\{\mathcal{P}_i\}_{i=1}^{N_r}$.*

Let $O^i(x) \in \mathbb{R}^{N_c^i}$ be a vector associated with region $\mathcal{P}_i$, and let the j-th element of $O^i(x)$ be defined as

$$O_j^i(x) = \begin{cases} +1 & f_i(x) \geqslant f_{C_i(j)}(x) \\ -1 & f_i(x) < f_{C_i(j)}(x) \end{cases} \tag{6.7}$$

Then $O^i(x)$ has the following properties:

(i) $O^i(x) = S^i = const, \quad \forall x \in \mathcal{P}_i$,
(ii)$O^i(x) \neq S^i, \quad \forall x \notin \mathcal{P}_i$.

Proof: Let $\mathcal{F} = \mathcal{P}_i \cap \mathcal{P}_{C_i(j)}$ be the common facet of $\mathcal{P}_i$ and $\mathcal{P}_{C_i(j)}$. Define the linear function

$$z_j^i(x) = f_i(x) - f_{C_i(j)}(x). \tag{6.8}$$

From the continuity of $f(x)$ it follows that $z_j^i(x) = 0, \forall x \in \mathcal{F}$. As $\mathcal{P}_i$ and $\mathcal{P}_{C_i(j)}$ are disjoint convex polyhedra and $A^i \neq A^{C_i(j)}$ it follows that $z_j^i(\xi_i) > 0$ (or $z_j^i(\xi_i) < 0$, but not both) for any interior point ξ_i of $\mathcal{P}_i$. Similarly for any interior point $\xi_{C_i(j)}$ of $\mathcal{P}_{C_i(j)}$ we have $z_j^i(\xi_{C_i(j)}) < 0$ (or $z_i^j(\xi_i) > 0$, but not both). Consequently, $z_j^i(x) = 0$ is the separating hyperplane between $\mathcal{P}_i$ and $\mathcal{P}_{C_i(j)}$.

(i) Because $z_j^i(x) = 0$ is a separating hyperplane, the function $z_j^i(x)$ does not change its sign for all $x \in \mathcal{P}_i$, i.e., $O_j^i(x) = s_j^i, \forall x \in \mathcal{P}_i$ with $s_j^i = +1$ or $s_j^i = -1$. The same reasoning can be applied to all neighbors of $\mathcal{P}_i$ to get the vector $S^i = \{s_j^i\} \in \mathbb{R}^{N_c^i}$.

(ii) $\forall x \notin \mathcal{P}_i$, $\exists j \in C_i$ such that $H_j^i x > K_j^i$. Since $z_j^i(x) = 0$ is a separating hyperplane then $O_j^i(x) = -s_j^i$. ■

Equivalently, Theorem 6.1 states that

$$x \in \mathcal{P}_i \Leftrightarrow O^i(x) = S^i, \tag{6.9}$$

which means that the function $O^i(x)$ and the vector S^i uniquely characterize $\mathcal{P}_i$. Therefore, to check on-line if the polyhedral region i contains the state x it is sufficient to compute the binary vector $O^i(x)$ and compare it with S^i. Vectors S^i are calculated off-line for all $i = 1, \ldots, N^r$, by comparing the values of $f_i(x)$ and $f_{C_i(j)}(x)$ for $j = 1, \ldots, N_c^i$, for a point x belonging to $\mathcal{P}_i$, for instance, the Chebychev center of $\mathcal{P}_i$.

In Figure 6.2 a one dimensional example illustrates the procedure with $N^r = 4$ regions. The list of neighboring regions C_i and the vector S^i can be constructed by simply looking at the figure: $C_1 = \{2\}$, $C_2 = \{1, 3\}$, $C_3 = \{2, 4\}$, $C_4 = \{3\}$, $S^1 = -1$, $S^2 = [-1\ 1]'$, $S^3 = [1\ \ -1]'$, $S^4 = -1$. The point $x = 4$ is in region 2 and we have $O^2(x) = [-1\ 1]' = S^2$, while $O^3(x) = [-1\ \ -1]' \neq S^3$, $O^1(x) = 1 \neq S^1$, $O^4(x) = 1 \neq S^4$. The failure of a match $O^i(x) = S^i$ provides information on a good search direction(s). The solution can be found by searching in the direction where a constraint is violated, i.e., one should check the neighboring region $\mathcal{P}_j$ for which $O_j^i(x) \neq s_j^i$.

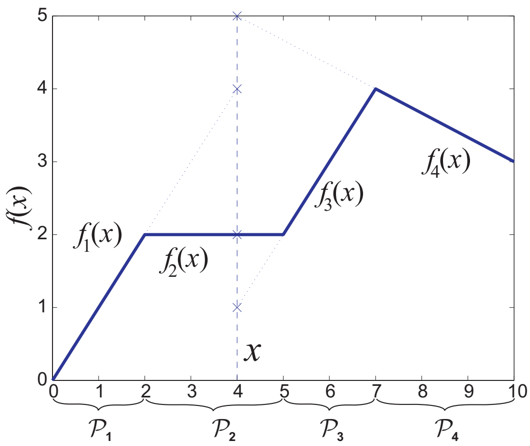


Fig. 6.2. Example for Algorithm 6.2.4 in one dimension: For a given point $x \in \mathcal{P}_2$ $(x = 4)$ we have $O_2(x) = [-1\ 1]' = S_2$, while $O_1(x) = 1 \neq S_1 = -1$, $O_3(x) = [-1\ \ -1]' \neq S_3 = [1\ \ -1]'$, $O_4(x) = 1 \neq S_4 = -1$.

The overall procedure is composed of two parts:

1. *(off-line)* Construction of the PWA function $f(x)$ in (6.5) satisfying (6.6) and computation of the list of neighbors C_i and the vector S^i,
2. *(on-line)* Execution of the following algorithm

Algorithm 6.2.4

Input: Current state x, the list of neighboring regions C_i and the vectors S^i

Output: Index i of the polyhedron $\mathcal{P}_j$ containing the current state x in the control law (6.1)

$i = 1$, notfound=1;
while notfound
  compute $O^i(x)$
  if $O^i(x) = S^i$ **then** notfound=0
  else $i = C_i(q)$, where $O^i_q(x) \neq s^i_q$.
end

Algorithm 6.2.4 does not require the storage of the polyhedra $\mathcal{P}_i$, but only the storage of one linear function $f_i(x)$ per polyhedron, i.e., $N^r(n+1)$ real numbers and the list of neighbors C_i which requires N_C integers. In the worst case, Algorithm 6.2.4 terminates after $N^r n$ multiplications, $N^r(n-1)$ sums and N_C comparisons.

In Table 6.2 we compare the complexity of Algorithm 6.2.4 against the standard Algorithm 6.2.2 in terms of storage demand and number of flops.

Remark 6.3. Note that the computation of $O^i(x)$ in Algorithm 6.2.4 requires the evaluation of N^i_c linear functions, but the overall computation never exceeds N^r linear function evaluations. Consequently, Algorithm 6.2.4 will outperform Algorithm 6.2.2, since typically $N_C \gg N^r$.

Table 6.2. Complexity comparison of Algorithm 6.2.2 and Algorithm 6.2.4

	Algorithm 6.2.2	Algorithm 6.2.4
Storage demand (real numbers)	$(n+1)N_C$	$(n+1)N^r$
Number of flops (worst case)	$2nN_C$	$(2n-1)N^r + N_C$

Now that we have shown how to locate the polyhedron in which the state lies by using a PWA descriptor function, we need a procedure for the construction of such a function.

The image of the descriptor function is the set of real numbers $\mathbb{R}$. In the following we will show how a descriptor function can be generated from a vector valued function $m : \mathbb{R}^n \to \mathbb{R}^s$. This general result will be used in the next subsections.

Definition 6.2 (Vector valued PWA descriptor function). *A continuous vector valued PWA function*

$$m(x) = \bar{A}^i x + \bar{B}^i, \ \text{if } x \in \mathcal{P}_i, \tag{6.10}$$

is called a vector valued PWA descriptor function if

$$\bar{A}^i \neq \bar{A}^j, \ \forall j \in C_i, \ i = 1, \ldots, N_r. \tag{6.11}$$

where $\bar{A}^i \in \mathbb{R}^{s \times n}$, $\bar{B}^i \in \mathbb{R}^s$.

Theorem 6.2. *Given a vector valued PWA descriptor function* $m(x)$ *defined over a polyhedral partition* $\{\mathcal{P}_i\}_{i=1}^{N_r}$ *it is possible to construct a PWA descriptor function* $f(x)$ *over the same polyhedral partition.*

Proof: Let $\mathcal{N}_{i,j}$ be the null-space of $(\bar{A}^i - \bar{A}^j)'$. Since by definition $\bar{A}^i - \bar{A}^j \neq \mathbf{0}$ it follows that $\mathcal{N}_{i,j}$ is not full dimensional, i.e., $\mathcal{N}_{i,j} \subseteq \mathbb{R}^{s-1}$. Consequently, it is always possible to find a vector $w \in \mathbb{R}^s$ such that $w'(\bar{A}^i - \bar{A}^j) \neq \mathbf{0}$ holds for all $i = 1, \ldots, N_r$ and $\forall j \in C_i$. Clearly, $f(x) = w'm(x)$ is then a valid PWA descriptor function. ■

As shown in the proof of Theorem 6.2, once we have vector valued PWA descriptor function, practically any randomly chosen vector $w \in \mathbb{R}^s$ is likely to be satisfactory for the construction of a PWA descriptor function. From a numerical point of view however, we would like to obtain w that is as far away as possible from the null-spaces $\mathcal{N}_{i,j}$. We show one algorithm for finding such a vector w.

For a given vector valued PWA descriptor function we form set of vectors $a_k \in \mathbb{R}^s$, $\|a_k\| = 1$, $k = 1, \ldots, N_C/2$, by taking and normalizing one (and only one) nonzero column from each matrix $(\bar{A}^i - \bar{A}^j)$, $\forall j \in C_i, \ i = 1, \ldots, N_r$. The vector $w \in \mathbb{R}^s$ satisfying the set of equations $w'a_k \neq 0, \ k = 1, \ldots, N_C/2$, can then be constructed by using the following algorithm[1]

Algorithm 6.2.5

Input: Vectors $a_i \in \mathbb{R}^s, \ i = 1, \ldots, N$.
Output: The vector $w \in \mathbb{R}^s$ satisfying the set of equations $w'a_i \neq 0, \ i = 1, \ldots, N$

$w \leftarrow [1, \ldots, 1]'$, $R \leftarrow 1$
while $k \leq N_C/2$
  $d \leftarrow w'a_k$
  if $0 \leq d \leq R$ **then** $w \leftarrow w + \frac{1}{2}(R-d)a_k$, $R \leftarrow \frac{1}{2}(R+d)$
  if $-R \leq d < 0$ **then** $w \leftarrow w - \frac{1}{2}(R+d)a_k$, $R \leftarrow \frac{1}{2}(R-d)$
end

Algorithm 6.2.5 is based on the construction of a sequence of balls $\mathcal{B} = \{x \mid x = w + r, \ \|r\|_2 \leq R\}$. As depicted in Figure 6.3, Algorithm 6.2.5 starts

[1] Index k goes to $N_C/2$ since the term $(\bar{A}^j - \bar{A}^i)$ is the same as $(\bar{A}^i - \bar{A}^j)$ and thus there is no need to consider it twice.

with the initial ball of radius $R = 1$, centered at $w = [1, \dots, 1]'$. Iteratively one hyperplane $a_k' x = 0$ at a time is introduced and the largest ball $\mathcal{B}' \subseteq \mathcal{B}$ that does not intersect this hyperplane is designed. The center w of the final ball is the vector w we want to construct, while R provides an information about the degree of non-orthogonality: $|w' a_k| \geq R, \forall k$.

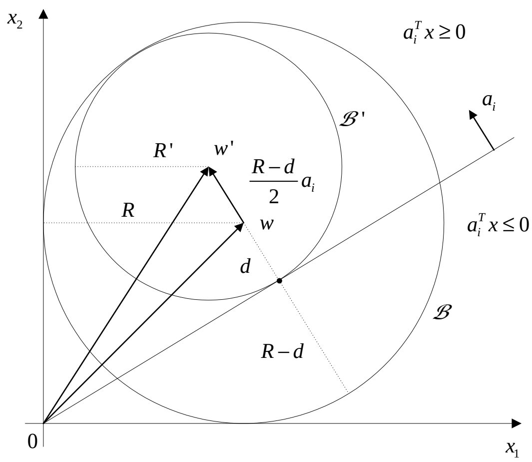


Fig. 6.3. Illustration for Algorithm 6.2.5 in two dimensions.

In the following subsections we will show that the gradient of the value function, and the optimizer, are vector valued PWA descriptor functions and thus we can use Algorithm 6.2.5 for the construction of the PWA descriptor function.

Generating a PWA descriptor function from the value function

Let $J^*(x)$ be the convex and piecewise quadratic (CPWQ) value function obtained as a solution of the CFTOC (2.4) problem for $p = 2$:

$$J^*(x) = q_i(x) \triangleq x' Q^i x + T^{i'} x + V^i, \quad \text{if } x \in \mathcal{P}_i, \ i = 1, \dots, N^r. \tag{6.12}$$

In Section 1.4.4 we have proven that for non-degenerate problems the value function $J^*(x)$ is a $C^{(1)}$ function. We can obtain a vector valued PWA descriptor function by differentiating $J^*(x)$.

Theorem 6.3. *Consider the value function $J^*(x)$ in (6.12) and assume that the CFTOC (2.4) problem leads to a non-degenerate mp-QP (2.20). Then the gradient $m(x) \triangleq \nabla J^*(x)$, is a vector valued PWA descriptor function.*

Proof: From Theorem 1.15 we see that $m(x)$ is continuous vector valued PWA function, while from equation (1.46) we get

$$m(x) \triangleq \nabla J^*(x) = 2Q_i x + T_i \tag{6.13}$$

Since from Theorem 1.13 we know that $Q_i \neq Q_j$ for all neighboring polyhedra, it follows that $m(x)$ satisfies all conditions for a vector valued PWA descriptor function. ■

Combining results of Theorem 6.3 and Theorem 6.2 it follows that by using Algorithm 6.2.5 we can construct a PWA descriptor function from the gradient of the value function $J^*(x)$.

Generating a PWA descriptor function from the optimizer

Another way to construct descriptor function $f(x)$ emerges naturally if we look at the properties of the optimizer $U_N^*(x)$ corresponding to the state feedback solution of the CFTOC problem (2.4). From Theorem 1.12 it follows that the optimizer $U_N^*(x)$ is continuous in x and piecewise affine on polyhedra:

$$U_N^*(x) = l_i(x) \triangleq F^i x + g^i, \text{ if } x \in \mathcal{P}_i, \ i = 1, \dots, N^r, \tag{6.14}$$

where $F^i \in \mathbb{R}^{s \times n}$ and $g^i \in \mathbb{R}^s$. We will assume that $\mathcal{P}_i$ are critical regions in the sense of Definition 1.37. All we need to show is the following lemma.

Lemma 6.1. *Consider the state feedback solution (6.14) of the CFTOC problem (2.4) and assume that the CFTOC (2.4) leads to a non-degenerate mp-QP (2.20). Let $\mathcal{P}_i$, $\mathcal{P}_j$ be two neighboring polyhedra, then $F^i \neq F^j$.*

Proof: The proof is a simple consequence of Theorem 1.13. As in Theorem 1.13, without loss of generality we can assume that the set of active constraints $\mathcal{A}_i$ associated with the critical region $\mathcal{P}_i$ is empty, i.e., $\mathcal{A}_i = \emptyset$. Assume that the optimizer is the same for both polyhedra, i.e., $[F^i \ g^i] = [F^j \ g^j]$. Then, the cost functions $q_i(x)$ and $q_j(x)$ are also equal. From the proof of Theorem 1.13 this implies that $\mathcal{P}_i = \mathcal{P}_j$, which is a contradiction. Thus we have $[F^i \ g^i] \neq [F^j \ g^j]$. Note that $F^i = F^j$ cannot happen since, from the continuity of $U_N^*(x)$, this would imply $g^i = g^j$. Consequently we have $F^i \neq F^j$. □

From Lemma 6.1 and Theorem 6.2 it follows that an appropriate PWA descriptor function $f(x)$ can be calculated from the optimizer $U^*(x)$ by using Algorithm 6.2.5.

Remark 6.4. Note that even if we are implementing a receding horizon control strategy, the construction of the PWA descriptor function is based on the full optimization vector $U^*(x)$ and the corresponding matrices $\bar{F}^i$ and $\bar{g}^i$.

Remark 6.5. In some cases the use of the optimal control profile $U^*(x)$ for the construction of a descriptor function $f(x)$ can be extremely simple. If there is a row r, $r \leq m$ (m is the dimension of u) for which $(F^i)^r \neq (F^j)^r$, $\forall i =$

$1 \dots, N^r$, $\forall j \in C_i$, it is enough to set $A^{i'} = (F^i)^r$ and $B^i = (g^i)^r$, where $(F^i)^r$ and $(g^i)^r$ denote the r-th row of the matrices F^i and g^i, respectively. In this way we *avoid* the storage of the descriptor function altogether, since it is equal to one component of the control law, which is stored anyway.

6.3 Example

As an example, we compare the performance of Algorithms 6.2.2, 6.2.3 and 6.2.4 on CFTOC problem for the discrete-time system

$$\begin{cases} x(t+1) = \begin{bmatrix} 4 & -1.5 & 0.5 & -0.25 \\ 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0.083 & 0.22 & 0.11 & 0.02 \end{bmatrix} x(t) \end{cases} \tag{6.15}$$

resulting from the linear system

$$y = \frac{1}{s^4} u \tag{6.16}$$

sampled at $T_s = 1$, subject to the input constraint

$$-1 \leq u(t) \leq 1 \tag{6.17}$$

and the output constraint

$$-10 \leq y(t) \leq 10. \tag{6.18}$$

6.3.1 CFTOC based on LP

To regulate (6.15), we design a receding horizon controller based on the optimization problem (4.28) where $p = \infty$, $N = 2$, $Q = \text{diag}\{5, 10, 10, 10\}$, $R = 0.8$, $P = 0$. The PPWA solution of the mp-LP problem was computed in 240 s on a Pentium III 900 MHz machine running Matlab 6.0. The corresponding polyhedral partition of the state-space consists of 136 regions. In Table 6.3 we report the comparison between the complexity of Algorithm 6.2.2 and Algorithm 6.2.3 for this example.

The average on-line evaluation of the PPWA solution for a set of 1000 random points in the state space is 2259 flops (Algorithm 6.2.2), and 1088 flops (Algorithm 6.2.3). We point out that the solution using Matlab's LP solver (function linprog.m with interior point algorithm and LargeScale set to 'off') takes 25459 flops on average.

Table 6.3. Complexity comparison of Algorithm 6.2.2 and Algorithm 6.2.3 for the example in Section 6.3.1

	Algorithm 6.2.2	Algorithm 6.2.3
Storage demand (real numbers)	5690	680
Number of flops (worst case)	9104	1088
Number of flops (average for 1000 random points)	2259	1088

6.3.2 CFTOC based on QP

To regulate (6.15), we design a receding horizon controller based on the optimization problem (4.28) where $p = 2$, $N = 7$, $Q = I$, $R = 0.01$, $P = 0$. The PPWA solution of the mp-QP problem was computed in 560 s on a Pentium III 900 MHz machine running Matlab 6.0. The corresponding polyhedral partition of the state-space consists of 213 regions. For this example the choice of $w = [1\ 0\ 0\ 0]'$ is satisfactory to obtain a descriptor function from the value function. In Table 6.4 we report the comparison between the complexity of Algorithm 6.2.2 and Algorithm 6.2.4 for this example.

The average computation of the PPWA solution for a set of 1000 random points in the state space is 2114 flops (Algorithm 6.2.2), and 175 flops (Algorithm 6.2.4). The solution of the corresponding quadratic program with Matlab's QP solver (function quadprog.m and LargeScale set to 'off') takes 25221 flops on average.

Table 6.4. Complexity comparison of Algorithm 6.2.2 and Algorithm 6.2.4 for the example in Section 6.3.2

	Algorithm 6.2.2	Algorithm 6.2.4
Storage demand (real numbers)	9740	1065
Number of flops (worst case)	15584	3439
Number of flops (average for 1000 random points)	2114	175

Part III

Optimal Control of Hybrid Systems

7

Hybrid Systems

In this chapter we give a short overview of discrete time linear hybrid systems, i.e., the class of systems that explicitly takes into account continuous valued and discrete valued variables as well as their interaction in one common framework. A more detailed and comprehensive review of the topic following the same structure and argumentation can be found in [115].

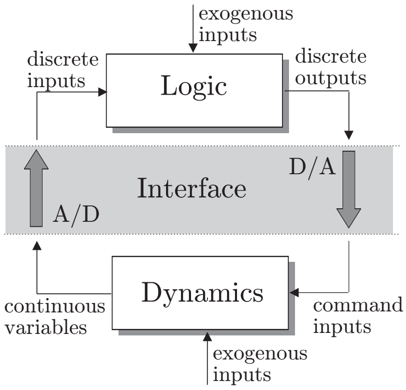
7.1 Introduction

The mathematical *model* of a system is traditionally associated with differential or difference equations, typically derived from physical laws governing the *dynamics* of the system under consideration. Consequently, most of the control theory and tools have been developed for such systems, in particular, for systems whose evolution is described by smooth linear or nonlinear state transition functions. On the other hand, in many applications the system to be controlled comprises also parts described by *logic*, such as for instance on/off switches or valves, gears or speed selectors, and evolutions dependent on if-then-else rules. Often, the control of these systems is left to schemes based on heuristic rules inferred from practical plant operation.

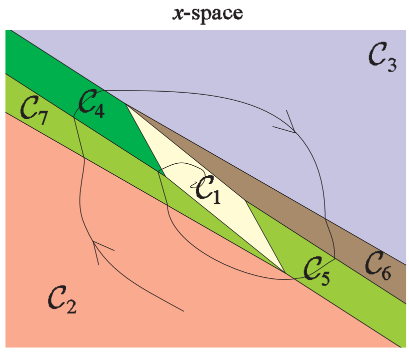
Recently, researchers started dealing with *hybrid systems*, namely processes which evolve according to dynamic equations and logic rules. There are many examples of hybrid systems [6]. Hierarchical systems constituted by dynamical components at the lower level, governed by upper level logical/discrete components [6, 130, 41, 103, 39] are hybrid systems. Sometimes, in the control field a continuous time linear plant described by linear differential equations and controlled by a discrete-time system described by difference equations, is also considered a hybrid system. This class is usually referred to as *hybrid control systems* [5, 7, 23, 105, 106] (Fig. 7.1(a)) and it is not considered in this book. Another familiar example of hybrid system is a switching systems where the dynamic behaviour of the system is described by a finite number of dynamical models, that are typically sets of differential or difference equations, together with a set of rules for switching among these models. The switching rules are described by logic expressions or a discrete event system with a finite automaton or a Petri net representation. *Switched systems* [40, 92, 136] belong to this class of systems (Fig. 7.1(b)).

The interest in hybrid systems has grown over the last few years not only because of the theoretical challenges, but also because of their impact on applications. Hybrid systems arise in a large number of application areas and are attracting increasing attention in both academic theory-oriented circles as well as in industry, for instance the automotive industry [9, 93].

In this chapter we briefly discuss a framework for modeling, analyzing and controlling hybrid systems. We will focus exclusively on *discrete time* models. We note, however, that interesting mathematical phenomena occurring in hybrid systems, such as Zeno behaviors [94] do not exist in discrete-time. On



(a) Hybrid control systems. Finite state machines and continuous dynamics interact through analog-to-digital (A/D) and D/A interfaces



(b) Switched systems

Fig. 7.1. Hybrid models

the other hand, most of these phenomena are usually a consequence of the continuous time switching model, rather than the real natural behaviour.

Among various modeling frameworks used for discrete-time hybrid systems, *piecewise affine* (PWA) systems [136] are the most studied. PWA systems are defined by partitioning the state and input space into polyhedral regions, and associating with each region a different affine state-update equation

$$x(t+1) = A^i x(t) + B^i u(t) + f^i \quad \text{if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{C}^i \tag{7.1}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$, $\{\mathcal{C}_i\}_{i=0}^{s-1}$ is a polyhedral partition of the sets of state and input space $\mathbb{R}^{n+m}$, $n \triangleq n_c + n_\ell$, $m \triangleq m_c + m_\ell$, the matrices A^i, B^i, f^i are constant and have suitable dimensions. We will give the following definition of continuous PWA system:

Definition 7.1. *We say that the PWA system (7.1) is* continuous *if the mapping* $(x_c(t), u_c(t)) \mapsto x_c(t+1)$ *is continuous and* $n_\ell = m_\ell = 0$.

Definition 7.2. *We say that a PWA system (7.1) is* continuous in the input space *if the mapping* $(x_c(t), u_c(t)) \mapsto x_c(t+1)$ *is continuous in the* u_c *space and* $n_\ell = m_\ell = 0$.

Definition 7.3. *We say that a PWA system (7.1) is* continuous in the real input space *if the mapping* $(x_c(t), u_c(t)) \mapsto x_c(t+1)$ *is continuous in the* u_c *space.*

PWA systems are equivalent to interconnections of linear systems and finite automata, as pointed out by Sontag [139]. Discrete-time linear PWA systems can model *exactly* a large number of physical processes, such as discrete-time linear systems with static piecewise-linearities or switching systems where the dynamic behaviour is described by a finite number of discrete-time linear models, together with a set of logic rules for switching among these models. Moreover, PWA systems can *approximate* nonlinear discrete-time dynamics via multiple linearizations at different operating points and also the more general class of continuous time nonlinear hybrid systems by sampling the continuous dynamics and substituting the nonlinearity with piecewise-linear approximations.

In [85, 20] the authors have proven the equivalence of discrete-time PWA systems and other classes of discrete-time hybrid systems such as linear complementarity (LC) systems [83, 153, 84] and extended linear complementarity (ELC) systems [54], max-min-plus-scaling (MMPS) systems [55], and mixed logic dynamical (MLD) systems [23]. Each modeling framework has its advantages and the equivalence of discrete-time PWA, LC, ELC, MMPS, and MLD hybrid dynamical systems allows one to easily transfer the theoretical properties and tools from one class to another.

Our main motivation for concentrating on discrete-time models stems from the need to analyze these systems and to solve optimization problems, such as optimal control or scheduling problems, for which the continuous time counterpart would not be easily computable. For this reason we need a way to describe discrete-time PWA systems in a form that is suitable for recasting analysis/synthesis problems into a more compact optimization problem. The MLD framework described in the following section has been developed for such a purpose. In particular, it will be proven that MLD models are successful in recasting hybrid dynamical optimization problems into mixed-integer linear and quadratic programs, solvable via branch and bound techniques [120].

7.2 Mixed Logic Dynamical (MLD) Systems

MLD systems [23] allow specifying the evolution of continuous variables through linear dynamic equations, of discrete variables through propositional logic statements and automata, and the mutual interaction between the two. The key idea of the approach consists of embedding the logic part in the state equations by transforming Boolean variables into 0-1 integers, and by expressing the relations as mixed-integer linear inequalities [23, 47, 125, 151, 157, 158, 120]. The MLD modeling framework relies on the idea of translating logic relations into mixed-integer linear inequalities [23, 125, 158]. The following correspondence between a Boolean variable X and its associated binary variable δ will be used:

$$\begin{aligned} X &= \text{true} \Leftrightarrow \delta = 1 \\ X &= \text{false} \Leftrightarrow \delta = 0 \end{aligned} \tag{7.2}$$

Linear dynamics are represented by difference equations $x(t+1) = Ax(t) + Bu(t)$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$. Boolean variables can be defined from linear-threshold conditions over the continuous variables: $[X = \text{true}] \leftrightarrow [a'x \leq b]$, $x,\ a \in \mathbb{R}^n$, $b \in \mathbb{R}$, with $(a'x - b) \in [m, M]$. This condition can be equivalently expressed by the two mixed-integer linear inequalities:

$$\begin{array}{l} a'x - b \leq M(1-\delta) \\ a'x - b > m\delta. \end{array} \tag{7.3}$$

Similarly, a relation defining a continuous variable z depending on the logic value of a Boolean variable X

$$\text{IF } X \text{ THEN } z = a_1'x - b_1 \text{ ELSE } z = a_2'x - b_2,$$

can be expressed as

$$\begin{array}{r} (m_2 - M_1)\delta + z \leq a_2'x - b_2 \\ (m_1 - M_2)\delta - z \leq -a_2'x + b_2 \\ (m_1 - M_2)(1-\delta) + z \leq a_1'x - b_1 \\ (m_2 - M_1)(1-\delta) - z \leq -a_1'x + b_1. \end{array} \tag{7.4}$$

where, assuming that $x \in \mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{X}$ is a given bounded set,

$$M_i \geq \sup_{x \in \mathcal{X}} (a_i'x - b_i),\ m_i \leq \inf_{x \in \mathcal{X}} (a_i'x - b_i),\ i = 1, 2.$$

are upper and lower bounds, respectively, on $(a'x - b)$. Note that (7.4) also represents the hybrid product $z = \delta(a_1'x - b_1) + (1-\delta)(a_2'x - b_2)$ between binary and continuous variables.

A Boolean variable X_n can be defined as a Boolean function of Boolean variables $f : \{true, false\}^{n-1} \to \{true, fase\}$, namely

$$X_n \leftrightarrow f(X_1, X_2, \ldots, X_{n-1}) \tag{7.5}$$

where f is a combination of "not" ($\sim$), "and" ($\wedge$), "or" ($\vee$), "exclusive or" ($\oplus$), "implies" ($\leftarrow$), and "iff" ($\leftrightarrow$) operators. The logic expression (7.5) is equivalent to its conjunctive normal form (CNF) [158]

$$\bigwedge_{j=1}^{k} \left(\left(\bigvee_{i \in P_j} X_i \right) \bigvee \left(\bigvee_{i \in N_j} \sim X_i \right) \right), N_j, P_j \subseteq \{1, \ldots, n\}$$

Subsequently, the CNF can be translated into the set of integer linear inequalities

$$\begin{cases} 1 \leq \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) \\ \vdots \\ 1 \leq \sum_{i \in P_k} \delta_i + \sum_{i \in N_k} (1 - \delta_i) \end{cases} \tag{7.6}$$

Alternative methods for translating any logical relation between Boolean variables into a set of linear integer inequalities can be found in chapter 2 of [115]. In [115] the reader can also find a more comprehensive and complete treatment of the topic.

The state update law of finite state machines can be described by logic propositions involving binary states, their time updates and binary signals, under the assumptions that the transitions are clocked and synchronous with the sampling time of the continuous dynamical equations, and that the automaton is well-posed (i.e., at each time step a transition exists and is unique)

$$x_\ell(t+1) = F(x_\ell(t), u_\ell(t)) \tag{7.7}$$

where u_l is the vector of Boolean signals triggering the transitions of the automaton. Therefore, the automaton is equivalent to a nonlinear discrete time system where F is a purely Boolean function. The translation technique mentioned above can be applied directly to translate the automaton (7.7) into a set of linear integer equalities and inequalities.

By collecting the equalities and inequalities derived from the representation of the hybrid system we obtain the Mixed Logic Dynamical (MLD) system [23]

$$x(t+1) = Ax(t) + B_1 u(t) + B_2 \delta(t) + B_3 z(t) \tag{7.8a}$$

$$y(t) = Cx(t) + D_1 u(t) + D_2 \delta(t) + D_3 z(t) \tag{7.8b}$$

$$E_2 \delta(t) + E_3 z(t) \leq E_1 u(t) + E_4 x(t) + E_5 \tag{7.8c}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is a vector of continuous and binary states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ are the inputs, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$ the outputs, $\delta \in \{0,1\}^{r_\ell}$, $z \in \mathbb{R}^{r_c}$ represent auxiliary binary and continuous variables respectively, which are introduced when transforming logic relations into mixed-integer linear inequalities, and A, B_{1-3}, C, D_{1-3}, E_{1-5} are matrices of suitable dimensions.

We assume that system (7.8) is *completely well-posed* [23], which means that for all x, u within a bounded set the variables δ, z are uniquely determined, i.e., there exist functions F, G such that, at each time t, $\delta(t) = F(x(t), u(t))$, $z(t) = G(x(t), u(t))$. The functions F and G are implicitly defined through the inequalities (7.8c). Then it follows that $x(t+1)$ and $y(t)$ are uniquely defined once $x(t)$, $u(t)$ are given, and therefore that x- and y-trajectories exist and are uniquely determined by the initial state $x(0)$ and input signal $u(t)$. It is clear that the well-posedness assumption stated above is usually guaranteed by the procedure used to generate the linear inequalities (7.8c), and therefore this hypothesis is typically fulfilled by MLD relations derived from modeling real-world plants. Nevertheless, a numerical test for well-posedness is reported in [23, Appendix 1].

Note that the constraints (7.8c) allow us to specify additional linear constraints on continuous variables (e.g., constraints over physical variables of the system), and logic constraints of the type "$f(\delta_1, \ldots, \delta_n) = 1$", where f is a Boolean expression. The ability to include constraints, constraint prioritization, and heuristics adds to the expressiveness and generality of the MLD framework. Note also that the description (7.8) appears to be linear, with the nonlinearity concentrated in the integrality constraints over binary variables.

Remark 7.1. In (7.8c), we only allowed nonstrict inequalities, as we are interested in using numerical solvers with MLD models. Therefore, strict inequalities (as in (7.3)) of the form $a'x > b$ must be approximated by $a'x \geq b + \epsilon$ for some $\epsilon > 0$ (e.g., the machine precision), with the assumption that $0 < a'x - b < \epsilon$ cannot occur due to the finite precision of the computer. After such approximation some regions of the state space will be considered unfeasible for the MLD model (7.8) even if they were feasible for the original PWA model (7.1). However, the measure of this regions tends to zero as ϵ goes to 0. Note that this issue arises only in case of discontinuous PWA systems.

7.3 HYSDEL

The translation procedure of a description of a hybrid dynamical system into mixed integer inequalities is the core of the tool HYSDEL (Hybrid Systems Description Language), which automatically generates an MLD model from a high-level textual description of the system [150]. Given a textual description of the logic and dynamic parts of the hybrid system, HYSDEL returns the matrices $A, B_{1-3}, C, D_{1-3}, E_{1-5}$ of the corresponding MLD form (7.8). A full description of HYSDEL is given in [150].
The compiler is available at `http://control.ethz.ch/ hybrid/hysdel`.

7.4 Theoretical Properties of PWA Systems

Despite the fact that PWA models are just a composition of linear time-invariant dynamical systems, their structural properties such as observability, controllability, and stability are complex and articulated, as typical of nonlinear systems. In this section we want to discuss in a sketchy manner, stability, controllability and observability of discrete time PWA systems.

Stability

Besides simple but very conservative results such as finding one common quadratic Lyapunov function for all the components, researchers have started developing analysis and synthesis tools for PWA systems only very recently [92, 116, 59]. By adopting piecewise quadratic Lyapunov functions, a computational approach based on Linear Matrix Inequalities has been proposed in [81] and [92] for stability analysis and control synthesis. Construction of Lyapunov functions for switched systems has also been tackled in [156]. For the general class of switched systems of the form $\dot{x} = f_i(x)$, $i = 1, \ldots, s$, an extension of the Lyapunov criterion based on multiple Lyapunov functions was introduced in [38] and [40]. In their recent paper, Blondel and Tsitsiklis [34] showed that the stability of autonomous PWA systems is $\mathcal{NP}$-hard to verify (i.e. in general the stability of a PWA systems cannot be assessed by a polynomial-time algorithm, unless $\mathcal{P} = \mathcal{NP}$), even in the simple case of two component subsystems. Several global properties (such as global convergence and asymptotic stability) of PWA systems have been recently shown undecidable in [33].

The research on stability criteria for PWA systems has been motivated by the fact that the stability of each component subsystem is not sufficient to guarantee stability of a PWA system (and vice versa). Branicky [40], gives an example where stable subsystems are suitably combined to generate an unstable PWA system. Stable systems constructed from unstable ones have been reported in [152]. These examples point out that restrictions on the switching have to be imposed in order to guarantee that a PWA composition of stable components remains stable.

Observability and Controllability

Very little research has focused on observability and controllability properties of hybrid systems, apart from contributions limited to the field of timed automata [5, 86, 99] and the pioneering work of Sontag [136] for PWA systems. Needless to say, these concepts are fundamental for understanding *if* a state observer and a controller for a hybrid system can be designed and *how well* it performs. For instance, observability properties were directly exploited for designing convergent state estimation schemes for hybrid systems in [62].

Controllability and observability properties have been investigated in [58, 78] for linear time-varying systems, and in particular for the so-called class of piecewise constant systems (where the matrices in the state-space representation are piecewise constant functions of time). Although in principle applicable, these results do not allow to capture the peculiarities of PWA systems.

General questions of $\mathcal{NP}$-hardness of controllability assessment of nonlinear systems were addressed by Sontag [138]. Following his earlier results [136, 137], Sontag [139] analyzes the computational complexity of observability and controllability of PWA systems through arguments based on the language of piecewise linear algebra. The author proves that observability/controllability is $\mathcal{NP}$-complete over finite time, and is undecidable over infinite time (i.e. in general cannot be solved in finite time by means of any algorithm). Using a different rationale, the same result was derived in [34].

In [20] the authors provide two main contributions to the analysis of controllability and observability of hybrid and PWA systems: (i) they show the reader that observability and controllability properties can be very complex;

they present a number of counterexamples that rule out obvious conjectures about inheriting observability/controllability properties from the composing linear subsystems; (ii) they provide observability and controllability tests based on *Linear* and *Mixed-Integer Linear Programs* (MILPs).

8

Constrained Optimal Control for Hybrid Systems

In this chapter we study the solution to optimal control problems for discrete time linear hybrid systems. First, we show that the closed form of the state feedback solution to finite time optimal control based on quadratic or linear norms performance criteria is a time-varying piecewise affine state feedback control law. Then, we give insight into the structure of the optimal state feedback solution and of the value function. Finally, we describe how the optimal control law can be efficiently computed by means of multiparametric programming, for linear and quadratic performance criteria. For the linear performance criteria the piecewise affine feedback control law can be computed by means of multiparametric mixed integer linear programming. For quadratic performance criteria we present an algorithm to solve multiparametric mixed integer quadratic programs by means of dynamic programming and multiparametric quadratic programs.

8.1 Introduction

Different methods for the analysis and design of controllers for hybrid systems have emerged over the last few years [136, 146, 43, 81, 106, 23]. Among them, the class of optimal controllers is one of the most studied. Most of the literature deals with optimal control of continuous time hybrid systems and is focused on the study of necessary conditions for a trajectory to be optimal [140, 121], and on the computation of optimal or *sub-optimal* solutions by means of Dynamic Programming or the Maximum Principle [76, 82, 42, 129, 44]. Although some techniques for determining feedback control laws seem to be very promising, many of them suffer from the "curse of dimensionality" arising from the *discretization* of the state space necessary in order to solve the corresponding Hamilton-Jacobi-Bellman or Euler-Lagrange differential equations.

Our hybrid modeling framework is completely general, in particular the control switches can be both internal, i.e., caused by the state reaching a particular boundary, and controllable, i.e., one can decide when to switch to some other operating mode. Note however that interesting mathematical phenomena occurring in hybrid systems such as Zeno behaviors [94] do not exist in discrete time. We will show that for the class of linear discrete time hybrid systems we can *characterize* and *compute* the optimal control law *without gridding* the state space.

The solution to optimal control problems for discrete time hybrid systems was first outlined by Sontag in [136]. In his plenary presentation [110] at the 2001 European Control Conference Mayne presented an intuitively appealing characterization of the state feedback solution to optimal control problems for linear hybrid systems with performance criteria based on quadratic and linear norms. The detailed exposition presented in the first part of this chapter follows a similar line of argumentation and shows that the state feedback solution to the finite time optimal control problem is a time-varying piecewise affine feedback control law, possibly defined over non-convex regions. Moreover, we give insight into the structure of the optimal state feedback solution and of the value function.

An infinite horizon controller can be obtained by implementing in a receding horizon fashion a finite-time optimal control law. If one imposes an end-point constraint, then the resulting state feedback controller is stabilizing and respects all input and output constraints. The implementation, as a consequence of the results presented here on finite-time optimal control, requires only the evaluation of a piecewise affine function. This opens up the use of receding horizon techniques to control hybrid systems characterized by fast sampling and relatively small size. In collaboration with different companies we have applied this type of optimal control design to a range of hybrid control problems, for instance traction control described in Chapter 10.

In the next section we use the PWA modeling framework to derive the main properties of the state feedback solution to finite time optimal control problems for hybrid systems. Thanks to the aforementioned equivalence between PWA and MLD systems, the latter will be used in Section 8.5 to compute the optimal control law.

8.2 Problem Formulation

Consider the PWA system (7.1) subject to hard input and state constraints

$$Ex(t) + Lu(t) \leq M \tag{8.1}$$

for $t \geq 0$, and denote by *constrained PWA system* (CPWA) the restriction of the PWA system (7.1) over the set of states and inputs defined by (8.1),

$$x(t+1) = A^i x(t) + B^i u(t) + f^i \quad \text{if} \ \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \tilde{\mathcal{C}}^i \tag{8.2}$$

where $\{\tilde{\mathcal{C}}^i\}_{i=0}^{s-1}$ is the new polyhedral partition of the sets of state+input space $\mathbb{R}^{n+m}$ obtained by intersecting the sets $\mathcal{C}^i$ in (7.1) with the polyhedron described by (8.1).

Define the following cost function

$$J(U_N, x(0)) \triangleq \|Px_N\|_p + \sum_{k=0}^{N-1} \|Qx_k\|_p + \|Ru_k\|_p \tag{8.3}$$

and consider the constrained finite-time optimal control problem (CFTOC)

$$J^*(x(0)) \triangleq \inf_{\{U_N\}} J(U_N, x(0)) \tag{8.4}$$

$$\text{subj. to} \ \begin{cases} x_{k+1} = A^i x_k + B^i u_k + f^i \ \text{ if } \left[\begin{smallmatrix} x_k \\ u_k \end{smallmatrix}\right] \in \tilde{\mathcal{C}}^i \\ x_N \in \mathcal{X}_f \\ x_0 = x(0) \end{cases} \tag{8.5}$$

where the column vector $U_N \triangleq [u'_0, \ldots, u'_{N-1}]' \in \mathbb{R}^{m_c N} \times \{0,1\}^{m_\ell N}$, is the optimization vector, N is the time horizon and $\mathcal{X}_f$ is the terminal region. In (8.3), $\|Qx\|_p$ denotes the p-norm of the vector x weighted with the matrix Q, $p = 1, 2, \infty$. In the following, we will assume that $Q = Q' \succeq 0$, $R = R' \succ 0$, $P \succeq 0$, for $p = 2$, and that Q, R, P are full column rank matrices for $p = 1, \infty$. We will also denote with $\mathcal{X}_i \subseteq \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ the set of initial states $x(i)$ at time i for which the optimal control problem (8.3)-(8.5) is feasible:

$$\begin{array}{l}\mathcal{X}_i = \{x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell} | \begin{array}{l}\exists u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell} \text{ and } i \in \{0, \ldots, s-1\} | \\ \left[\begin{smallmatrix} x_k \\ u_k \end{smallmatrix}\right] \in \tilde{\mathcal{C}}^i \text{ and } A^i x + B^i u + f^i \in \mathcal{X}_{i+1}\end{array}\}, \\ i = 0, \ldots, N-1, \\ \mathcal{X}_N = \mathcal{X}_f. \end{array} \tag{8.6}$$

In general, the optimal control problem (8.3)-(8.5) may not have a minimum for some feasible $x(0)$. This is caused by discontinuity of the PWA system in the input space. We will consider problem (8.3)-(8.5) assuming that a minimizer $U_N{}^*(x(0))$ exists for all feasible $x(0)$. In this case, the optimal control problem (8.3)-(8.5) becomes

$$J^*(x(0)) \triangleq \min_{\{U_N\}} J(U_N, x(0)) \tag{8.7}$$

$$\text{subj. to } \begin{cases} x_{k+1} = A^i x_k + B^i u_k + f^i \quad \text{if } \left[\begin{smallmatrix} x_k \\ u_k \end{smallmatrix}\right] \in \tilde{\mathcal{C}}^i \\ x_N \in \mathcal{X}_f \\ x_0 = x(0) \end{cases} \tag{8.8}$$

The optimizer $U_N{}^*(x(0))$ of problem (8.3)-(8.5) does not exist for some $x(0)$ if the PWA systems (8.2) is discontinuous in the real input space and the infimizer of problem (8.3)-(8.5) lies on a the boundary of a polyhedron belonging to the polyhedral partition of the PWA system (8.2). Such a situation is of doubtful practical interest, in the sense that a small discrepancy between the "real" system and its model could lead to an arbitrarily large deviation between the performance of the closed loop system and its infimum.

In the following we need to distinguish between optimal control based on the 2-norm and optimal control based on the 1-norm or ∞-norm. Note that the results of this chapter also hold when the switching is weighted in the cost function (8.3), if this is meaningful in a particular situation.

8.3 State Feedback Solution of CFTOC, 2-Norm Case

Theorem 8.1. *The solution to the optimal control problem (8.3)-(8.8) is a PWA state feedback control law of the form $u^*(k) = f_k(x(k))$, where*

$$f_k(x(k)) = F_k^i x(k) + g_k^i \quad \textit{if } x(k) \in \mathcal{R}_k^i, \tag{8.9}$$

where $\mathcal{R}_k^i,\ i=1,\dots,N_k^r$ *is a partition of the set* $\mathcal{X}_k$ *of feasible states* $x(k)$ *and the closure* $\bar{\mathcal{R}}_k^i$ *of the sets* $\mathcal{R}_k^i$ *has the following form:*

$$\bar{\mathcal{R}}_k^i \triangleq \left\{x:\ x(k)'L(j)_k^i x(k) + M(j)_k^i x(k) \le N(j)_k^i,\ j=1,\dots,n_k^i\right\},\\ k=0,\dots,N-1.$$

Proof: The piecewise linearity of the solution was first mentioned by Sontag in [136]. In [110] Mayne sketched a proof. In the following we will give the proof for $u^*(x(0))$, the same arguments can be repeated for $u^*(x(1)),\dots,u^*(x(N-1))$. We will denote $u_0^*(x(0))$ by $u^*(x(0))$.

Depending on the initial state $x(0)$ and on the input sequence $U=[\ u_0',\dots,u_{k-1}']$ the state x_k is either infeasible or it belongs to a certain polyhedron $\tilde{\mathcal{C}}^i$. Suppose for the moment that there are no binary inputs, $m_\ell=0$. The number of all possible locations of the state sequence $x_0,\dots,x_{N-1}$ is equal to s^N. Denote by $\{v_i\}_{i=1}^{s^N}$ the set of all possible switching sequences over the horizon N , and by v_i^k the k-th element of the sequence v_i, i.e., $v_i^k=j$ if $x_k\in\tilde{\mathcal{C}}^j$.

Fix a certain v_i and constrain the state to switch according to the sequence v_i. Problem (8.3)-(8.8) becomes

$$J_{v_i}^*(x(0)) \triangleq \min_{\{U_N\}} J(U_N, x(0)) \tag{8.10}$$

$$\text{subj. to}\ \begin{cases} x_{k+1} & = A^i x_k + B^i u_k + f^i \\ & \quad \text{if } \left[\begin{smallmatrix} x_k \\ u_k \end{smallmatrix}\right] \in \tilde{\mathcal{C}}^i \\ \left[\begin{smallmatrix} x_k \\ u_k \end{smallmatrix}\right] \in \tilde{\mathcal{C}}^{v_i^k} & \quad k=0,\dots,N-1 \\ x_N \in \mathcal{X}_f & \\ x_0 = x(0) & \end{cases} \tag{8.11}$$

Assume for the moment that the sets $\tilde{\mathcal{C}}^{v_i^k},\quad k=0,\dots,N-1$ are closed. Problem (8.10)-(8.11) is equivalent to a finite time optimal control problem for a linear time-varying system with time-varying constraints and can be solved by using the approach outlined in Chapter 2. Its solution is the PPWA feedback control law

$$u^i(x(0)) = \tilde{F}^{i,j}x(0) + \tilde{g}^{i,j},\ \ \forall x(0)\in\mathcal{T}^{i,j},\ \ j=1,\dots,N^{ri} \tag{8.12}$$

where $\mathcal{D}^i=\bigcup_{j=1}^{N^{ri}}\mathcal{T}^{i,j}$ is a polyhedral partition of the convex set $\mathcal{D}^i$ of feasible states $x(0)$ for problem (8.10)-(8.11). N^{ri} is the number of regions of the polyhedral partition of the solution and is a function of the number of constraints in problem (8.10)-(8.11). The upperindex i in (8.12) denotes that the input $u^i(x(0))$ is optimal when the switching sequence v_i is fixed.

The optimal solution $u^*(x(0))$ to the original problem (8.3)-(8.8) can be found by solving problem (8.10)-(8.11) for all v_i. The set $\mathcal{X}_0$ of all feasible states at time 0 is $\mathcal{X}_0=\bigcup_{i=1}^{s^N}\mathcal{D}^i$ and, in general, it is not convex.

As some initial states can be feasible for different switching sequences, the sets $\mathcal{D}^i,\ i=1,\dots,s^N$, in general, can overlap. The solution $u^*(x(0))$ can be computed in the following way. For every polyhedron $\mathcal{T}^{i,j}$ in (8.12),

1. If $\mathcal{T}^{i,j} \cap \mathcal{T}^{l,m} = \emptyset$ for all $l \neq i$, $l = 1, \ldots, s^N$, $m = 1, \ldots, N^{rl}$, then the switching sequence v_i is the only feasible one for all the states belonging to $\mathcal{T}^{i,j}$ and therefore the optimal solution is given by (8.12), i.e.

$$u^*(x(0)) = \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j}, \quad \forall x \in \mathcal{T}^{i,j}. \tag{8.13}$$

2. If $\mathcal{T}^{i,j}$ intersects one or more polyhedra $\mathcal{T}^{l_1,m_1}$,$\mathcal{T}^{l_2,m_2}$,..., the states belonging to the intersection are feasible for more than one switching sequence v_i, v_{l_1}, v_{l_2}, ... and therefore the corresponding value functions $J^*_{v_i}(x(0))$, $J^*_{v_{l_1}}(x(0))$,$J^*_{v_{l_2}}(x(0))$, ... in (8.10) have to be compared in order to compute the optimal control law.
   Consider the simple case when only two polyhedra overlap, i.e. $\mathcal{T}^{i,j} \cap \mathcal{T}^{l,m} \triangleq \mathcal{T}^{(i,j),(l,m)} \neq \varnothing$. *We will refer to* $\mathcal{T}^{(i,j),(l,m)}$ *as a polyhedron of multiple feasibility*. For all states belonging to $\mathcal{T}^{(i,j),(l,m)}$ the optimal solution is:

$$u^*(x(0)) = \begin{cases} \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j}, & \forall x(0) \in \mathcal{T}^{(i,j),(l,m)} : J^*_{v_i}(x(0)) < J^*_{v_l}(x(0)) \\ \tilde{F}^{l,m} x(0) + \tilde{g}^{l,m}, & \forall x(0) \in \mathcal{T}^{(i,j),(l,m)} : J^*_{v_i}(x(0)) > J^*_{v_l}(x(0)) \\ \begin{cases} \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j} \text{ or} \\ \tilde{F}^{l,m} x(0) + \tilde{g}^{l,m} \end{cases} & \forall x(0) \in \mathcal{T}^{(i,j),(l,m)} : J^*_{v_i}(x(0)) = J^*_{v_l}(x(0)) \end{cases} \tag{8.14}$$

   Because $J^*_{v_i}(x(0))$ and $J^*_{v_l}(x(0))$ are quadratic functions on $\mathcal{T}^{i,j}$ and $\mathcal{T}^{l,m}$ respectively, the theorem is proved. In general, a polyhedron of multiple feasibility where n value functions intersect is partitioned into n subsets where in each one of them a certain value function is greater than all the others.

Assume now that there exists $\bar{k}$ such that $\tilde{\mathcal{C}}^{v_i^{\bar{k}}}$ is not closed. Then we will proceed in the following way. In problem (8.10)-(8.11) we substitute its closure $\overline{\tilde{\mathcal{C}}^{v_i^{\bar{k}}}}$. After having computed the PPWA solution (8.12), we exclude from the sets $\mathcal{D}^i$ all the initial states $x(0)$ belonging to $\overline{\tilde{\mathcal{C}}^{v_i^{\bar{k}}}}$ but not to the set $\tilde{\mathcal{C}}^{v_i^{\bar{k}}}$. The proof follows as outlined above, except that the sets $\mathcal{T}^{i,j}$ may be neither open nor closed polyhedra.

The proof can be repeated in the presence of binary inputs, $m_\ell \neq 0$. In this case the switching sequences v_i are given by all combinations of region indices and *binary inputs*, i.e., $i = 1, \ldots, (s * m_\ell)^N$. The continuous component of the optimal input is given by (8.13) or (8.14). Such an optimal continuous component of the input has an associated optimal sequence v_i which provides the remaining binary components of the optimal input.

□

Remark 8.1. Let $\mathcal{T}^{(i,j),(l,m)}$ be a polyhedron of multiple feasibility and let $\mathcal{F} = \{x \in \mathcal{T}^{(i,j),(l,m)} : J^*_{v_i}(x) = J^*_{v_l}(x)\}$ be the set where the quadratic functions $J^*_{v_i}(x)$ and $J^*_{v_l}(x)$ intersect (for the sake of simplicity we consider the case where only two polyhedra intersect). We distinguish four cases (sub-cases of case 2 in Theorem 8.1):

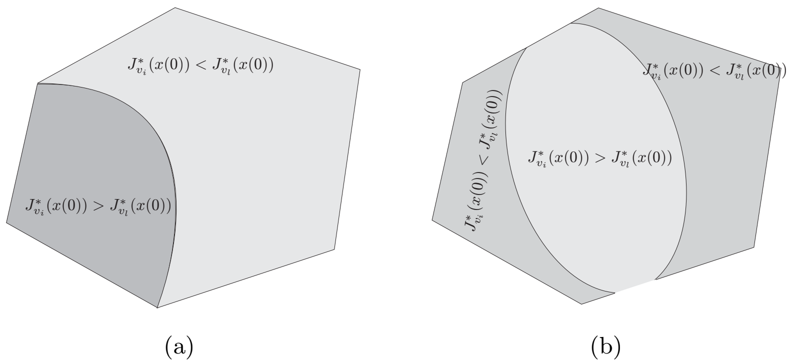


Fig. 8.1. Possible partitions corresponding to the optimal control law in case 2.d of Remark 8.1

2.a $\mathcal{F} = \emptyset$, i.e., $J^*_{v_i}(x)$ and $J^*_{v_l}(x)$ do not intersect over $\tilde{\mathcal{T}}^{(i,j),(l,m)}$.
2.b $\mathcal{F} = \{x : Ux = P\}$ and $J^*_{v_i}(x)$ and $J^*_{v_l}(x)$ are tangent on $\mathcal{F}$.
2.c $\mathcal{F} = \{x : Ux = P\}$ and $J^*_{v_i}(x)$ and $J^*_{v_l}(x)$ are not tangent on $\mathcal{F}$.
2.d $\mathcal{F} = \{x : x'Yx + Ux = P\}$ with $Y \neq 0$.

In the first case $\mathcal{T}^{(i,j),(l,m)}$ is not further partitioned, the optimal solution in $\mathcal{T}^{(i,j),(l,m)}$ is either $\tilde{F}^{i,j}x(0) + \tilde{g}^{i,j}$ or $\tilde{F}^{l,m}x(0) + \tilde{g}^{l,m}$. In case 2.*b*, $\mathcal{T}^{(i,j),(l,m)}$ is not further partitioned but there are multiple optimizers on the set $Ux = P$. In case 2.c, $\mathcal{T}^{(i,j),(l,m)}$ is partitioned into two polyhedra. In case 2.d $\mathcal{T}^{(i,j),(l,m)}$ is partitioned into two sets (not necessarily connected) as shown in Figure 8.1.

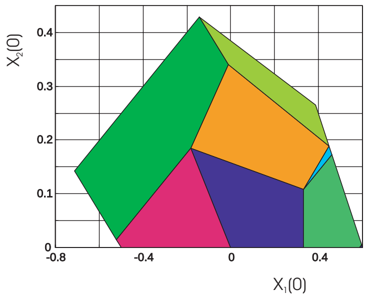
In the special situation where case 2.*c* or 2.*d* occur but the control laws are identical, i.e., $F^{i,j} = F^{l,m}$ and $\tilde{g}^{i,j} = \tilde{g}^{l,m}$, we will assume that the set $\mathcal{T}^{(i,j),(l,m)}$ is not further partitioned.

Example 8.1. Consider the following simple system

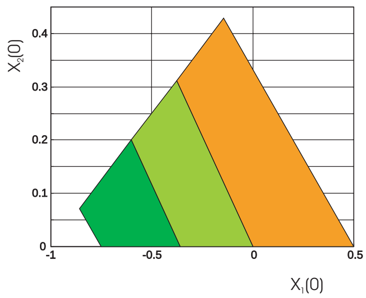
$$\begin{cases} x(t+1) = \begin{cases} \begin{bmatrix} -1 & 2 \\ 2 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \text{ if } x(t) \in \mathcal{C}^1 = \{x : [0\ 1]x \geq 0\} \\ \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \text{ if } x(t) \in \mathcal{C}_2 = \{x : [0\ 1]x < 0\} \end{cases} \\ x(t) \in [-1, -0.5] \times [1, 1] \\ u(t) \in [-1, 1] \end{cases} \tag{8.15}$$

and the optimal control problem (8.3)-(8.8), with $N = 2$, $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 10$, $P = Q$, $\mathcal{X}_f = \{x \in \mathbb{R}^2 | \begin{bmatrix} -1 \\ -0.5 \end{bmatrix} \leq x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}\}$.

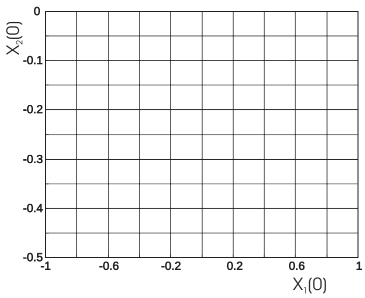
The possible switching sequences are $v_1 = \{1, 1\}$, $v_2 = \{1, 2\}$, $v_3 = \{2, 1\}$, $v_4 = \{2, 2\}$. The solution to problem (8.10)-(8.11) is depicted in Figure (8.2) (note that the switch v_3 is infeasible for all $x(0)$). In Figure 8.3(a) the four solutions are intersected, the white region corresponds to polyhedra of multiple feasibility. The state-space partition of the optimal control law is depicted in Figure 8.3(b) (for lack of space, we do not report here the analytic expressions of the regions and the corresponding affine gains).



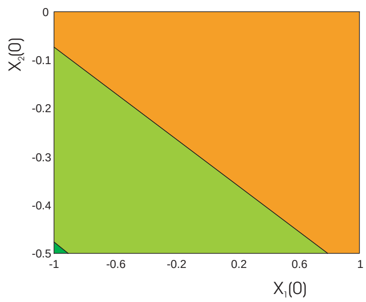
(a) States-space partition corresponding to the solution to problem (8.3)-(8.11) for $v_1 = \{1, 1\}$



(b) States-space partition corresponding to the solution to problem (8.3)-(8.11) for $v_2 = \{1, 2\}$



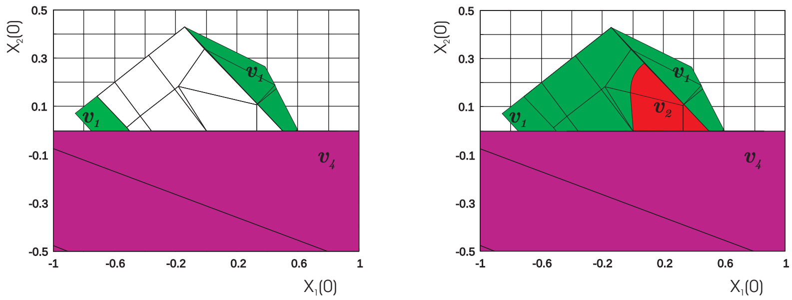
(c) Problem (8.3)-(8.11) for $v_3 = \{2, 1\}$ is infeasible.



(d) States-space partition corresponding to the solution to problem (8.3)-(8.11) for $v_4 = \{2, 2\}$

Fig. 8.2. First step for the solution of Example 8.1. Problem (8.3)-(8.11) is solved for different v_i, $i = 1, \ldots, 4$

Theorem 8.2. *Assume that the PWA system (8.2) is continuous and unconstrained. Then, the value function $J^*(x(0))$ in (8.8) is continuous.*



(a) Feasibility domain corresponding to the solution of Example 8.1 obtained by joining the solutions plotted in Figure 8.2. The white region corresponds to polyhedra of multiple feasibility.

(b) State-space partition corresponding to the optimal control law of Example 8.1

Fig. 8.3. State-space partition corresponding to the optimal control law of Example 8.1

Proof: The proof follows from the continuity of $J(U_N, x(0))$ and Theorem 1.1.

□

In general, for constrained PWA system it is difficult to find weak assumptions ensuring the continuity of the value function $J^*(x(0))$. Ensuring the continuity of the optimal control law $u^*(x(k))$ is even more difficult. Theorem 1.1 provides a sufficient condition for $U_N^*(x(0))$ to be continuous: it requires the cost $J(U_N, x(0))$ to be continuous and convex in U_N for each $x(0)$ and the constraints in (8.8) to be continuous and convex in U_N for each $x(0)$. Such conditions are clearly too restrictive since the cost and the constraints in problem (8.8) are a composition of quadratic and linear functions, respectively, with the piecewise linear dynamics of the system.

The next theorem provides a condition under which the solution $u^*(x(k))$ of the optimal control problem (8.3)-(8.8) is a PPWA state feedback control law.

Theorem 8.3. *Assume that the optimizer ${U_N}^*(x(0))$ of (8.3)-(8.8) is unique for all $x(0)$. Then the solution to the optimal control problem (8.3)-(8.8) is a PPWA state feedback control law of the form*

$$u^*(x(k)) = F_k^i x(k) + g_k^i \quad \text{if } x(k) \in \mathcal{P}_k^i \quad k = 0, \dots, N-1 \tag{8.16}$$

where $\mathcal{P}_k^i$, $i = 1, \dots, N_k^r$ is a polyhedral partition of the set $\mathcal{X}_k$ of feasible states $x(k)$.

Proof: : We will show that case 2.d in Remark 8.1 cannot occur by contradiction. Suppose case 2.d occurs. From the hypothesis the optimizer $u^*(x(0))$ is unique and from Theorem 8.1 the value function $J^*(x(0))$ is continuous on $\mathcal{F}$, this implies that $\tilde{F}^{i,j}x(0)+\tilde{g}^{i,j} = \tilde{F}^{l,m}x(0)+\tilde{g}^{l,m}$, $\forall x(0) \in \mathcal{F}$. That contradicts the hypothesis since the set $\mathcal{F}$ is not a hyperplane. The same arguments can be repeated for $u^*(x(k)),\ k = 1, \dots, N-1$. □

Remark 8.2. Theorem 8.3 relies on a rather weak uniqueness assumption. As the proof indicates, the key point is to exclude case 2d in Remark 8.1. Therefore, it is reasonable to believe that there are other conditions or problem classes which satisfy this structural property without claiming uniqueness. We are also currently trying to identify and classify situations where the state transition structure guarantees the absence of disconnected sets as shown in Figure 8.1(b).

The following Corollary summarizes the properties enjoyed by the solution to problem (8.3)-(8.8) as a direct consequence of Theorems 8.1-8.3

Corollary 8.1.
1. *$u^*(x(k))$ and $J^*(x(k))$ are, in general, discontinuous and $\mathcal{X}_k$ may be non-convex.*
2. *$J^*(x(k))$ can be discontinuous only on a facet of a polyhedron of multiple feasibility.*
3. *If there exists a polyhedron of multiple feasibility with $\mathcal{F} = \{x : x'Yx + Ux = P\}$, $Y \neq 0$, then on $\mathcal{F}$ $u^*(x(k))$ is not unique, except possibly at isolated points.*

8.4 State Feedback Solution of CFTOC, $1, \infty$-Norm Case

The results of the previous section can be extended to piecewise linear cost functions, i.e., cost functions based on the 1-norm or the ∞-norm.

Theorem 8.4. *The solution to the optimal control problem (8.3)-(8.8) with $p = 1,\ \infty$ is a PPWA state feedback control law of the form $u^*(k) = f_k(x(k))$*

$$f_k(x(k)) = F_k^i x(k) + g_k^i \quad \text{if } x(k) \in \mathcal{P}_k^i \tag{8.17}$$

where $\mathcal{P}_k^i,\ i = 1, \dots, N_k^r$ is a polyhedral partition of the set $\mathcal{X}_k$ of feasible states $x(k)$.

Proof: The proof is similar to the proof of Theorem 8.1. Fix a certain switching sequence v_i, consider the problem (8.3)-(8.8) and constrain the state to switch according to the sequence v_i to obtain problem (8.10)-(8.11). Problem (8.10)-(8.11) can be viewed as a finite time optimal control problem with performance index based on 1-norm or ∞-norm for a linear time varying system with time varying constraints and can be solved by using the multiparametric linear program as described in [18]. Its solution is a PPWA feedback control law

$$u^i(x(0)) = \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j}, \quad \forall x \in \mathcal{T}^{i,j}, \quad j = 1, \dots, N^{ri} \tag{8.18}$$

and the value function $J^*_{v_i}$ is piecewise affine on polyhedra and convex. The rest of the proof follows the proof of Theorem 8.1. Note that in this case the value functions to be compared are piecewise affine and not piecewise quadratic. □

8.5 Efficient Computation of the Optimal Control Input

In the previous section the properties enjoyed by the solution to hybrid optimal control problems were investigated. Despite the fact that the proof is constructive (as shown in the figures), it is based on the enumeration of all the possible switching sequences of the hybrid system, the number of which grows exponentially with the time horizon. Although the computation is performed off-line (the on-line complexity is the one associated with the evaluation of the PWA control law (8.16)), more efficient methods than enumeration are desirable. Here we show that the MLD framework can be used in order to avoid enumeration. In fact, when the model of the system is an MLD model and the performance index is quadratic, the optimization problem can be cast as a Mixed-Integer Quadratic Program (MIQP). Similarly, 1-norm and ∞-norm performance indices lead to Mixed-Integer Linear Programs (MILPs) problems. In the following we detail the translation of problem (8.3)-(8.8) into a mixed integer linear or quadratic program for which efficient branch and bound algorithms exist.

Consider the equivalent MLD representation (7.8) of the PWA system (8.2). Problem (8.3)-(8.8) is rewritten as:

$$\begin{aligned} J^*(x(0)) = \ \min_{U_N} \ & \|Px_N\|_p + \sum_{k=0}^{N-1} \|Q_1 u_k\|_p + \|Q_2 \delta_k\|_p \\ & + \|Q_3 z_k\|_p + \|Q_4 x_k\|_p \end{aligned} \tag{8.19}$$

$$\text{subj. to} \begin{cases} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k \\ E_2 \delta_k + E_3 z_k \leq E_1 u_k + E_4 x_k + E_5 \\ x_N \in \mathcal{X}_f \\ x_0 = x(0) \end{cases} \tag{8.20}$$

Note that the cost function (8.19) is more general than (8.3), and includes also the weight on the switching sequence.

The optimal control problem in (8.19)-(8.20) can be formulated as a *Mixed Integer Quadratic Program* (MIQP) when the squared Euclidean norm is used ($p = 2$) [23], or as a *Mixed Integer Linear Program* (MILP), when $p = \infty$ or $p = 1$ [16],

$$\begin{aligned} \min_{\varepsilon} \quad & \varepsilon' H_1 \varepsilon + \varepsilon' H_2 x(0) + x(0)' H_3 x(0) + c_1' \varepsilon + c_2' x(0) + c \\ \text{subj. to } & G\varepsilon \leq W + Sx(0) \end{aligned} \tag{8.21}$$

where H_1, H_2, H_3, c_1, c_2, G, W, S are matrices of suitable dimensions, $\varepsilon = [\varepsilon_c', \varepsilon_d']$ where ε_c, ε_d represent continuous and discrete variables, respectively and H_1, H_2, H_3, are null matrices if problem (8.21) is an MILP.

The translation of (8.19)-(8.20) into (8.21) for $p = 2$ is simply obtained by substituting the state update equation

$$x_k = A^k x_0 + \sum_{j=0}^{k-1} A^j (B_1 u_{k-1-j} + B_2 \delta_{k-1-j} + B_3 z_{k-1-j}) \tag{8.22}$$

and the optimization vector $\varepsilon = \{u_0, \dots, u_{N-1}, \delta_0, \dots, \delta_{N-1}, z_0, \dots, z_{N-1}\}$ in (8.21).

For $p = 1, \infty$, the translation requires the introductions of slack variables. In particular, for $p = \infty$ the sum of the components of any vector $\{\varepsilon_0^u, \dots, \varepsilon_{N-1}^u, \varepsilon_0^\delta, \dots, \varepsilon_{N-1}^\delta, \varepsilon_0^z, \dots, \varepsilon_{N-1}^z, \varepsilon_0^x, \dots, \varepsilon_N^x\}$ that satisfies

$$\begin{array}{l} -\mathbf{1}_m \varepsilon_k^u \leq Q_1 u_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_m \varepsilon_k^u \leq -Q_1 u_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_{r_\ell} \varepsilon_k^\delta \leq Q_2 \delta_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_{r_\ell} \varepsilon_k^\delta \leq -Q_2 \delta_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_{r_c} \varepsilon_k^z \leq Q_3 z_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_{r_c} \varepsilon_k^z \leq -Q_3 z_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_n \varepsilon_k^x \leq Q_4 x_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_n \varepsilon_k^x \leq -Q_4 x_k, \ k = 0, 1, \dots, N-1 \\ -\mathbf{1}_n \varepsilon_N^x \leq P x_N, \\ -\mathbf{1}_n \varepsilon_N^x \leq P x_N, \end{array} \tag{8.23}$$

represents an upper bound on $J^*(x(0))$, where $\mathbf{1}_k$ is a column vector of ones of length k, and where $x(k)$ is expressed as in (8.22). Similarly to what was shown in [45], it is easy to prove that the vector $\varepsilon \triangleq \{\varepsilon_0^u, \dots, \varepsilon_{N-1}^u, \varepsilon_0^\delta, \dots, \varepsilon_{N-1}^\delta, \varepsilon_0^z, \dots, \varepsilon_{N-1}^z, \varepsilon_0^x, \dots, \varepsilon_N^x, u(0), \dots, u(N-1)\}$ that satisfies equations (8.23) and simultaneously minimizes

$$J(\varepsilon) = \varepsilon_0^u + \dots + \varepsilon_{N-1}^u + \varepsilon_0^\delta + \dots + \varepsilon_{N-1}^\delta + \varepsilon_0^z + \dots + \varepsilon_{N-1}^z + \varepsilon_0^x + \dots + \varepsilon_N^x \tag{8.24}$$

also solves the original problem, i.e., the same optimum $J^*(x(0))$ is achieved. Therefore, problem (8.19)-(8.20) can be reformulated as the following MILP problem

$$
\begin{array}{rl}
\min\limits_{\varepsilon} & J(\varepsilon) \\
\text{subj. to} & -\mathbf{1}_m\varepsilon_k^u \leq \pm Q_1 u_k,\ k=0,1,\dots,N-1 \\
& -\mathbf{1}_m\varepsilon_k^\delta \leq \pm Q_2 \delta_k,\ k=0,1,\dots,N-1 \\
& -\mathbf{1}_m\varepsilon_k^z \leq \pm Q_3 z_k,\ k=0,1,\dots,N-1 \\
& -\mathbf{1}_n\varepsilon_k^x \leq \pm Q_4 (A^k x_0 + \sum_{j=0}^{k-1} A^j (B_1 u_{k-1-j} + \\
& \qquad B_2 \delta_{k-1-j} + B_3 z_{k-1-j}))\ k=0,\dots,N-1 \\
& -\mathbf{1}_n\varepsilon_N^x \leq \pm P (A^N x_0 + \sum_{j=0}^{N-1} A^j (B_1 u_{k-1-j} + \\
& \qquad B_2 \delta_{k-1-j} + B_3 z_{k-1-j})) \\
& x_{k+1} = A x_k + B_1 u_k + B_2 \delta_k + B_3 z_k,\ k \geq 0 \\
& E_2 \delta_k + E_3 z_k \leq E_1 u_k + E_4 x_k + E_5,\ k \geq 0 \\
& x_N \in \mathcal{X}_f \\
& x_0 = x(0)
\end{array}
\tag{8.25}
$$

where the variable $x(0)$ appears only in the constraints in (8.25) as a parameter vector.

Given a value of the initial state $x(0)$, the MIQP (or MILP) (8.21) can be solved to get the optimizer $\varepsilon^*(x(0))$ and therefore the optimal input $U_N^*(0)$. In Section 8.6 we will show how multiparametric programming can be used to efficiently compute the piecewise affine state feedback optimal control law (8.9) or (8.17).

8.6 Efficient Computation of the State Feedback Solution

Multiparametric programming [68, 57, 25, 37] can be used to efficiently compute the PWA form of the optimal state feedback control law $u^*(x(k))$. By generalizing the results for linear systems of previous chapters to hybrid systems, the state vector $x(0)$, which appears in the objective function and in the linear part of the right-hand-side of the constraints (8.21), can be handled as a vector of parameters. Then, for performance indices based on the ∞-norm or 1-norm, the optimization problem can be treated as a *multiparametric MILP* (mp-MILP), while for performance indices based on the 2-norm, the optimization problem can be treated as a *multiparametric MIQP* (mp-MIQP). Solving an mp-MILP (mp-MIQP) amounts to expressing the solution of the MILP (MIQP) (8.21) as a function of the parameters $x(0)$.

In Section 1.5 we have presented an algorithm for solving mp-MILP problems, while, to the authors' knowledge, there does not exist an efficient method for solving mp-MIQPs. In Section 8.7 we will present an algorithm that efficiently solves mp-MIQPs derived from optimal control problems for discrete time hybrid systems.

8.7 Computation of the State Feedback Solution, $1,\infty$-Norm Case

As mentioned before the solution of the mp-MILP (8.21) provides the state feedback solution $u^*(k) = f_k(x(k))$ (8.17) of CFTOC (8.3)-(8.8) for $k = 0$. A similar approach used for linear systems in Section 2.2, can be use here to compute the state feedback piecewise-affine optimal control law $f_k : x(k) \mapsto u^*(k)$ for $k = 1, \dots, N$, by solving N mp-MILP over a shorter horizon.

8.7.1 Example

Consider the problem of steering to a small region around the origin in three steps the simple piecewise affine system [23]

$$\begin{cases} x(t+1) = 0.8 \begin{bmatrix} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ \alpha(t) = \begin{cases} \frac{\pi}{3} \text{ if } [1\ 0]x(t) \geq 0 \\ -\frac{\pi}{3} \text{ if } [1\ 0]x(t) < 0 \end{cases} \\ x(t) \in [-10, 10] \times [-10, 10] \\ u(t) \in [-1, 1] \end{cases} \tag{8.26}$$

while minimizing the cost function (8.3) with $p = \infty$. By using auxiliary variables $z(t) \in \mathbb{R}^4$ and $\delta(t) \in \{0, 1\}$ such that $[\delta(t) = 1] \leftrightarrow [[1\ 0]x(t) \geq 0]$, Eq. (8.26) can be rewritten in the form (7.8) as in [23].

$$\begin{cases} x(t+1) = [\, I \ \ I \,]\, z(t) \\ \begin{bmatrix} 5 \\ -5-\epsilon \\ -M \\ -M \\ M \\ M \\ M \\ M \\ -M \\ -M \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \\ I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} z(t) \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ B \\ -B \\ B \\ -B \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} u(t) + \begin{bmatrix} 1 \quad 0 \\ -1 \quad 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ A_1 \\ -A_1 \\ A_2 \\ -A_2 \\ I \\ -I \\ 0 \\ 0 \end{bmatrix} x(t) + \begin{bmatrix} 5 \\ -\epsilon \\ 0 \\ 0 \\ M \\ M \\ M \\ M \\ 0 \\ 0 \\ N \\ N \\ 1 \\ 1 \end{bmatrix} \end{cases}$$

where $B = [0\ 1]'$, A_1, A_2 are obtained by (8.26) by setting respectively $\alpha = \frac{\pi}{3}, -\frac{\pi}{3}$, $M = 4(1+\sqrt{3})[1\ 1]' + B$, $N \triangleq 5[1\ 1]'$, and ϵ is a properly chosen small positive scalar.

The finite-time constrained optimal control problem (8.3)-(8.8) is solved with $N = 3$, $P = Q = \left[\begin{smallmatrix} 700 & 0 \\ 0 & 700 \end{smallmatrix}\right]$, $R = 1$, and $\mathcal{X}_f = [-0.01\ 0.01] \times [-0.01\ 0.01]$, by solving the associate mp-MILP problem (8.21). The polyhedral regions corresponding to the state feedback solution $u^*(x(k)),\ k = 0, \dots, 2$ in (8.17) are depicted in Fig. 8.4. The resulting optimal trajectories for the initial state $x(0) = [-1\ 1]'$ are shown in Fig. 8.5.

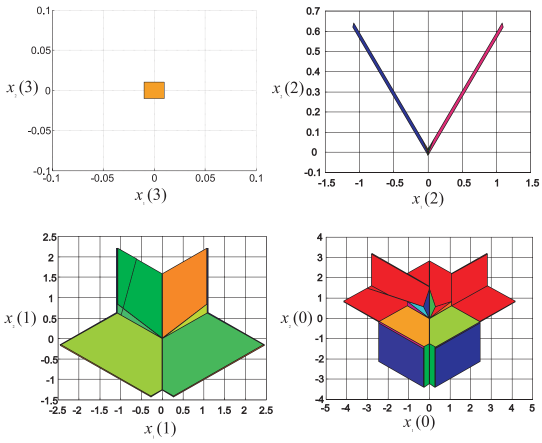


Fig. 8.4. State space partition corresponding to the state feedback finite time optimal control law $u^*(x(k))$ of system (8.26).

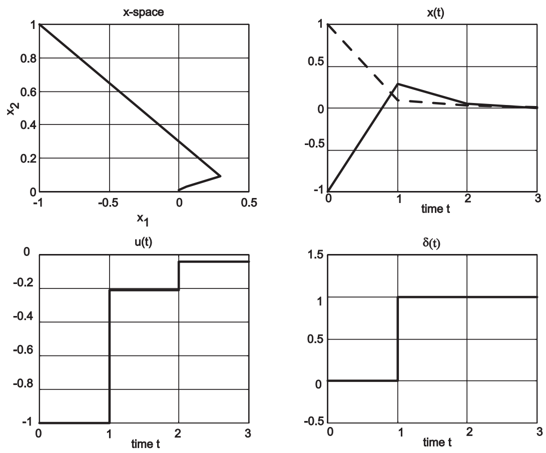


Fig. 8.5. Finite time optimal control of system (8.26).

8.8 Computation of the State Feedback Solution, 2-Norm Case

Often the use of linear norms has practical disadvantages. A satisfactory performance may be only achieved with long time-horizons with a consequent increase of complexity. Also closed-loop performance may not depend smoothly on the weights used in the performance index, i.e., slight changes of the weights could lead to very different closed-loop trajectories, so that the tuning of the controller becomes difficult. Here we propose an efficient algorithm for computing the solution to the finite time optimal control problem for discrete time linear hybrid systems with a quadratic performance criterion. The algorithm is based on a dynamic programming recursion and a multiparametric quadratic programming solver.

8.8.1 Preliminaries and Basic Steps

Denote with $f_{PPWA}(x)$ and $f_{PPWQ}(x)$ a generic PPWA and PPWQ function of x, respectively.

Definition 8.1. *A function $q : \Theta \to \mathbb{R}$, where $\Theta \subseteq \mathbb{R}^s$ is a* multiple quadratic function *of multiplicity $d \in \mathbb{N}^+$ if $q(\theta) = \min\{q^1(\theta) \triangleq \theta' Q^1\theta + l^1\theta + c^1, \ldots, q^d(\theta) \triangleq \theta' Q^d\theta + l^d\theta + c^d\}$ and Θ is a convex polyhedron.*

Definition 8.2. *A function $q : \Theta \to \mathbb{R}$, where $\Theta \subseteq \mathbb{R}^s$, is a* multiple PWQ on polyhedra *(multiple PPWQ) if there exists a polyhedral partition $\mathcal{R}_1,\ldots,\mathcal{R}_N$ of Θ and $q(\theta) = min\{q_i^1\theta \triangleq \theta' Q_i^1\theta + l_i^1\theta + c_i^1, \ldots, q_i^{d_i} \triangleq \theta' Q_i^{d_i}\theta + l_i^{d_i}\theta + c_i^{d_i}\}$, $\forall \theta \in \mathcal{R}_i$, $i = 1, \ldots, N$. We define d_i to be the multiplicity of the function q in the polyhedron $\mathcal{R}_i$ and $d = \sum_{i=1}^N d_i$ to be the multiplicity of the function q. (Note that Θ is not necessarily convex.)*

Multiparametric Quadratic Programming

The following quadratic program

$$\begin{aligned} V(x) = \min_u \quad & \tfrac{1}{2}u' H u + x' F u \\ \text{subj. to } & Gu \le W + Ex \end{aligned} \tag{8.27}$$

can be solved for all x by using an Multiparametric Quadratic Program solver (mp-QP) described in Section 1.4. The solution to the parametric program (8.27) is a PPWA law $u^*(x) = f_{PPWA}(x)$ and the value function is PPWQ, $V(x) = f_{PPWQ}(x)$.

Procedure Intersect and Compare

Consider the PWA map $\zeta(x)$

$$\zeta : \ x \mapsto F^i x + g^i \text{ if } x \in \mathcal{R}^i \ i = 1, \ldots, N^{\mathcal{R}} \tag{8.28}$$

where $\mathcal{R}^i$, $i = 1, \ldots, N^{\mathcal{R}}$ are sets of the $x-$space. If there exist $l, m \in \{1, \ldots, N^{\mathcal{R}}\}$ such that for $x \in \mathcal{R}^l \cap \mathcal{R}^m$, $F^l x + g^l \neq F^m x + g^m$ the map $\zeta(x)$ (8.28) is not single valued.

Definition 8.3. *Given a PWA map (8.28) we define the function $f_{PWA}(x) = \zeta_o(x)$ as the* **ordered region** *single-valued function associated with (8.28) when $\zeta_o(x) = F^j x + g^j$, $j \in \{1, \ldots, N^{\mathcal{R}}\} | x \in \mathcal{R}^j$ and $\forall i < j \ x \notin \mathcal{R}^i$.*

Note that given a PWA map (8.28) the corresponding *ordered region* single-valued function changes if the order used to store the regions $\mathcal{R}^i$ and the corresponding affine gains changes.

In the following we assume that the sets $\mathcal{R}^i_k$ in the optimal solution (8.9) can overlap. When we refer to the PWA function $u^*(x(k))$ in (8.9) we will implicitly mean the *ordered region single-valued function* associated with the mapping (8.9).

Theorem 8.5. *Let $J_1^* : \mathcal{P}^1 \to \mathbb{R}$ and $J_2^* : \mathcal{P}^2 \to \mathbb{R}$ be two quadratic functions, $J_1^*(x) \triangleq x'L^1x + M^1x + N^1$ and $J_2^*(x) \triangleq x'L^2x + M^2x + N^2$, where $\mathcal{P}^1$ and $\mathcal{P}^2$ are convex polyhedra and $J_i^*(x) = +\infty$ if $x \notin \mathcal{P}^i$. Consider the nontrivial case $\mathcal{P}^1 \cap \mathcal{P}^2 \triangleq \mathcal{P}^3 \neq \emptyset$ and the expressions*

$$J^*(x) = \min\{J_1^*(x), J_2^*(x)\} \tag{8.29}$$

$$u^*(x) = \begin{cases} {u^1}^*(x) & \text{if } J_1^*(x) \le J_2^*(x) \\ {u^2}^*(x) & \text{if } J_1^*(x) > J_2^*(x) \end{cases} \tag{8.30}$$

Define $L^3 = L^2 - L^1$, $M^3 = M^2 - M^1$, $N^3 = N^2 - N^1$. Then, corresponding to the three following cases

1. $J_1^*(x) \le J_2^*(x)\ \forall x \in \mathcal{P}^3$
2. $J_1^*(x) \ge J_2^*(x)\ \forall x \in \mathcal{P}^3$
3. $\exists\ x_1, x_2 \in \mathcal{P}^3 | J_1^*(x_1) < J_2^*(x_1)$ *and* $J_1^*(x_2) > J_2^*(x_2)$

the expressions (8.29) and (8.30) can be written equivalently as:

1.

$$J^*(x) = \begin{cases} J_1^*(x) & \text{if } x \in \mathcal{P}^1 \\ J_2^*(x) & \text{if } x \in \mathcal{P}^2 \end{cases} \tag{8.31}$$

$$u^*(x) = \begin{cases} {u^1}^*(x) & \text{if } x \in \mathcal{P}^1 \\ {u^2}^*(x) & \text{if } x \in \mathcal{P}^2 \end{cases} \tag{8.32}$$

2. *as in (8.31) and (8.32) but switching the index* 1 *with* 2
3.

$$J^*(x) = \begin{cases} min\{J_1^*(x), J_2^*(x)\} & \text{if } x \in \mathcal{P}^3 \\ J_1^*(x) & \text{if } x \in \mathcal{P}^1 \\ J_2^*(x) & \text{if } x \in \mathcal{P}^2 \end{cases} \tag{8.33}$$

$$u^*(x) = \begin{cases} {u^1}^*(x) & \text{if } x \in \mathcal{P}^3 \text{ and } x'L^3x + M^3x + N^3 \ge 0 \\ {u^2}^*(x) & \text{if } x \in \mathcal{P}^3 \text{ and } x'L^3x + M^3x + N^3 \le 0 \\ {u^1}^*(x) & \text{if } x \in \mathcal{P}^1 \\ {u^2}^*(x) & \text{if } x \in \mathcal{P}^2 \end{cases} \tag{8.34}$$

where (8.31)–(8.34) have to be considered as PWA and PPWQ functions in the ordered region *sense.*

Proof: The proof is very simple and is omitted here. □

The results of Theorem 8.5 allow one

- to avoid the storage of the intersections of two polyhedra in case 1. and 2. of Theorem 8.5
- to avoid the storage of possibly non-convex regions $\mathcal{P}^3 \setminus \mathcal{P}^1$ and $\mathcal{P}^3 \setminus \mathcal{P}^2$
- to work with multiple quadratic functions instead of quadratic functions defined over non-convex and non-polyhedral regions.

The three point listed above will be the three basic ingredients for storing and simplifying the optimal control law (8.9). Next we will show how to compute it.

Remark 8.3. To distinguish between cases 1, 2 and 3 of Theorem 8.5 one needs to solve an indefinite quadratic program. In our approach if one fails to distinguish between the three cases (e.g. if one solves a relaxed problem instead of the indefinite quadratic program) then the form (8.34) corresponding to the third case, will be used. The only drawback is that the form (8.34) could be a non-minimal representation of the value function and could therefore increase the computational complexity of the on-line algorithm for computing the optimal control action (8.9).

Parametric Programming

Consider the multiparametric program

$$J^*(x) \triangleq \min_{\{u\}} \; l(x,u) + q(f(x,u)) \tag{8.35}$$

$$\text{subj. to } \; f(x,u) \in \mathcal{P} \tag{8.36}$$

where $\mathcal{P} \subseteq \mathbb{R}^n$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $q : \mathcal{P} \to \mathbb{R}$, and $l : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a quadratic function of x and u. We aim at determining the region $\mathcal{X}$ of variables x such that the program (8.35)–(8.36) is feasible and the optimum $J^*(x)$ is finite, and at finding the expression of the optimizer $u^*(x)$.

In the following we will assume that $f(x,u)$ linear in x and u and we will show how to solve the following problems

1. *one to one problem*: $q(x)$ is quadratic in x, and $\mathcal{P}$ is a convex polyhedron.
2. *one to one problem of multiplicity d*: $q(x)$ is a multiple quadratic function of x of multiplicity d and $\mathcal{P}$ is a convex polyhedron.
3. *one to r problem*: $q(x)$ is a PPWQ function of x defined over r polyhedral regions.
4. *one to r problem of multiplicity d*: $q(x)$ is a multiple PPWQ function of x of multiplicity d, defined over r polyhedral regions.

If the function f is PPWA defined over s regions then we have an *s to X problem* where X can belong to any combination listed above. For example we have an *s to r problem of multiplicity d* if $f(x,u)$ is PPWA in x and u defined over s regions and $q(x)$ is a multiple PPWQ function of x of multiplicity d, defined over r polyhedral regions.

Theorem 8.6. *A* one to one problem *is solved with one mp-QP*

Proof: See Section 1.4

Theorem 8.7. *A* one to one problem of multiplicity *d is solved by solving d mp-QPs.*

Proof: The multiparametric program to be solved is

$$J^*(x) = \min_{\{u\}} \quad l(x,u) + \min\{q_1(f(x,u)), \ldots, q_d(f(x,u))\} \tag{8.37}$$

$$\text{subj. to } f(x,u) \in \mathcal{P} \tag{8.38}$$

and it is equivalent to

$$J^*(x) = \min \left\{ \begin{array}{ll} \min_{\{u\}} l(x,u) + q_1(f(x,u)), \ldots, & \min_{\{u\}} l(x,u) + q_d(f(x,u))\} \\ \text{subj. to } f(x,u) \in \mathcal{P} & \text{subj. to } f(x,u) \in \mathcal{P} \end{array} \right\} \tag{8.39}$$

The i-th sub-problems in (8.39)

$$J_i^*(x) \triangleq \min_{\{u\}} l(x,u) + q_i(f(x,u)) \tag{8.40}$$

$$\text{subj. to } f(x,u) \in \mathcal{P} \tag{8.41}$$

is a *one to one problem* and therefore it is solvable with an mp-QP. Let the solution of the i-th mp-QPs be

$$u^i(x) = \tilde{F}^{i,j} x + \tilde{g}^{i,j}, \quad \forall x \in \mathcal{T}^{i,j}, \quad j = 1, \ldots, N^{ri} \tag{8.42}$$

where $\mathcal{T}^i = \bigcup_{j=1}^{N^{ri}} \mathcal{T}^{i,j}$ is a polyhedral partition of the convex set $\mathcal{T}^i$ of feasible states x for the i-th sub-problem and N^{ri} is the number of polyhedral regions. The feasible set $\mathcal{X}$ satisfies $\mathcal{X} = \mathcal{T}^1 = \ldots = \mathcal{T}^d$ since the constraints of the d sub-problems are identical.

The solution $u^*(x)$ to the original problem (8.38) is obtained by comparing and storing the solution of d mp-QPs subproblems (8.41) as explained in Theorem 8.5. Consider the case $d = 2$, and consider the intersection of the polyhedra $\mathcal{T}^{1,i}$ and $\mathcal{T}^{2,l}$ for $i = 1, \ldots, N^{r1}$, $l = 1, \ldots, N^{r2}$. For all $\mathcal{T}^{1,i} \cap \mathcal{T}^{2,l} \triangleq \mathcal{T}^{(1,i),(2,l)} \neq \varnothing$ the optimal solution is stored in an ordered way as described in Theorem 8.5, while paying attention that a region could be already stored. Moreover while storing a new polyhedron with the corresponding value function and optimal controller the relative order of the regions already stored must not be changed. The result of this *Intersect and Compare* procedure is

$$u^*(x) = F^i x + g^i \text{ if } x \in \mathcal{R}^i \triangleq \{x: \ x' L^i(j) x + M^i(j) x \leq N^i(j), \ j = 1, \ldots, n^i\} \tag{8.43}$$

where $\mathcal{R} = \bigcup_{j=1}^{N_{\mathcal{R}}} \mathcal{R}^j$ is a convex polyhedron and the value function

$$J^*(x) = \tilde{J}_j^*(x) \text{ if } x \in \mathcal{D}^j, j = 1, \ldots, N^{\mathcal{D}} \tag{8.44}$$

where $\tilde{J}_j^*(x)$ are multiple quadratic functions defined over the convex polyhedra $\mathcal{D}^j$. The polyhedra $\mathcal{D}^j$ can contain more regions $\mathcal{R}^i$'s or can coincide with one of them. Note that (8.43) and (8.44) have to be considered as PWA and PPWQ function in the *ordered region* sense.

If $d > 2$ then the value function in (8.44) is intersected with the solution of the third mp-QP sub-problem and the procedure is iterated making sure not to change the relative order of polyhedra and corresponding gain of the solution constructed at the previous steps. The solution will still have the same form of (8.43)– (8.44). □

Theorem 8.8. *A* one to r problem *is solved with* r *mp-QPs*

Proof: Let $q(x) \triangleq q_i$ if $x \in \mathcal{P}^i$ be a PWQ function where $\mathcal{P}^i$ are convex polyhedra and q_i quadratic functions. The multiparametric program to solve is

$$J^*(x) = \min \left\{ \begin{array}{ll} \min_{\{u\}} l(x,u) + q_1(f(x,u)), \ldots, & \min_{\{u\}} l(x,u) + q_r(f(x,u))\} \\ \text{subj. to } f(x,u) \in \mathcal{P}_1 & \text{subj. to } f(x,u) \in \mathcal{P}_r \end{array} \right\} \tag{8.45}$$

The proof is similar to the proof of the previous theorem with the only difference that the constraints of the i-th mp-QP subproblem differ from the one of the j-th mp-QP subproblem, $i \neq j$. Therefore the procedure based on solving mp-QPs and storing as in Theorem 8.5 will be the same but the domain $\mathcal{R} = \bigcup_{j=1}^{N^{\mathcal{R}}} \mathcal{R}^j$ of the solution

$$u^*(x) = F^i x + g^i \text{ if } x \in \mathcal{R}^i \triangleq \{x : \ x' L^i x + M^i x \leq N^i\} \tag{8.46}$$

$$J^*(x) = \tilde{J}_i^*(x) \text{ if } x \in \mathcal{P}^i \tag{8.47}$$

can be a non-convex polyhedron and even disconnected if the domain of the PWA function $f(x,u)$ is not connected. □

The following Lemmas can be proven along the same lines of the proofs given before.

Lemma 8.1. *An* one to r problem of multiplicity d *is solved with* rd *mp-QPs*

Lemma 8.2. *An* s to one problem *can be decomposed into* s one to one problem.

Lemma 8.3. *An* s to r problem *can be decomposed into* s one to r problems.

Lemma 8.4. *An* s to r problem of multiplicity d *can be decomposed into* s one to r problem of multiplicity d.

8.8.2 Efficient Dynamic Program for the Computation of the Solution

In the following we will substitute the CPWA system equations (8.2) with the shorter form

$$x(k+1) = \tilde{f}_{PWA}(x(k), u(k)) \tag{8.48}$$

where $\tilde{f}_{PWA} : \tilde{\mathcal{C}} \to \mathbb{R}^n$ and $\tilde{f}_{PWA}(x,u) = A^i x + B^i u + f^i$ if $\left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \in \tilde{\mathcal{C}}^i,\ i = 1, \ldots, s$.

The PWA solution (8.9) to the CFTOC (8.3)-(8.8) can be computed efficiently in the following way.

Consider the dynamic programming solution to the CFTOC (8.3)-(8.8)

$$\begin{aligned} J_j^*(x(j)) \triangleq \ & \min_{\{u_j\}} \ \|Qx_j\|_2 + \|Ru_j\|_2 + J_{j+1}^*(\tilde{f}_{PWA}(x(j), u_j) && (8.49) \\ & \text{subj. to } \ \tilde{f}_{PWA}(x(j), u_j) \in \mathcal{X}_{j+1} && (8.50) \end{aligned}$$

for $j = N-1, \ldots, 0$, with terminal conditions

$$\begin{aligned} \mathcal{X}_N &= \mathcal{X}_f && (8.51) \\ J_N^*(x) &= \|Px\|_2 && (8.52) \end{aligned}$$

where $\mathcal{X}_j$ is the set of all initial states for which problem (8.49)–(8.50) is feasible:

$$\mathcal{X}_j = \{x \in \mathbb{R}^n |\ \exists u,\ \tilde{f}_{PWA}(x,u) \in \mathcal{X}_{j+1}\} \tag{8.53}$$

The dynamic program (8.49)–(8.52) can be solved backwards in time by using a multiparametric quadratic programming solver and the results of the previous section.

Assume for the moment that there are no binary inputs and logic states, $m_\ell = n_\ell = 0$. Consider the first step of the dynamic program (8.49)–(8.52)

$$\begin{aligned} J_{N-1}^*(x(N-1)) \triangleq \ & \min_{\{u_{N-1}\}} \ \|Qx(N-1)\|_2 + \|Ru(N-1)\|_2 + \\ & \qquad + J_N^*(\tilde{f}_{PWA}(x(N-1), u_{N-1})) && (8.54) \\ & \text{subj. to } \ \tilde{f}_{PWA}(x(N-1), u_{N-1}) \in \mathcal{X}_f && (8.55) \end{aligned}$$

The cost to go function $J_N^*(x)$ in (8.54) is quadratic, the terminal region $\mathcal{X}_f$ is a polyhedron and the constraints are piecewise affine. Problem (8.54)–(8.55) is an *s to one problem* that can be solved by solving s mp-QPs (Lemma 8.2).

From the second step $j = N-2$ to the last one $j = 0$ the cost to go function $J_{j+1}^*(x)$ is a PPWQ with a certain multiplicity d_{j+1}, the terminal region $\mathcal{X}_{j+1}$ is a polyhedron (not necessary convex) and the constraints are piecewise affine. Therefore, problem (8.49)–(8.52) is a *s to N_{j+1}^r problem with multiplicity* d_{j+1} (where N_{j+1}^r is the number of polyhedra of the cost to go function J_{j+1}^*), that can be solved by solving $sN_{j+1}^r d_{j+1}$ mp-QPs (Lemma 8.4). The resulting optimal solution will have the form (8.9) considered in the ordered region sense.

In the presence of binary inputs the procedure can be repeated, with the difference that all the possible combinations of binary inputs have to be enumerated. Therefore a *one to one problem* becomes a 2^{m_ℓ} *to one problem* and so on. Clearly the procedure become prohibitive for a system with a high number of binary inputs. In this case a multiparametric mixed integer programming (or a combination of the two techniques) is preferable. In the presence of logic states the procedure can be repeated either by relaxing the n_ℓ logic states to assume continuous values between 0 and 1 or by enumerating them all.

8.8.3 Example

Consider the problem of steering the piecewise affine system (8.26) to a small region around the origin in three steps while minimizing the cost function (8.3). The finite-time constrained optimal control problem (8.3)-(8.8) is solved with $N = 3, Q = I, R = 1, P = I$, and $\mathcal{X}_f = [-0.01\ 0.01]\times[-0.01\ 0.01]$. The solution was computed in less than 1 minute by using Matlab 5.3 on a Pentium II-500 MH. The polyhedral regions corresponding to the state feedback solution $u^*(x(k)),\ k = 0, 1, 2$ in (8.9) are depicted in Fig. 8.6. The resulting optimal trajectories for the initial state $x(0) = [-1\ 1]'$ are shown in Fig. 8.7.

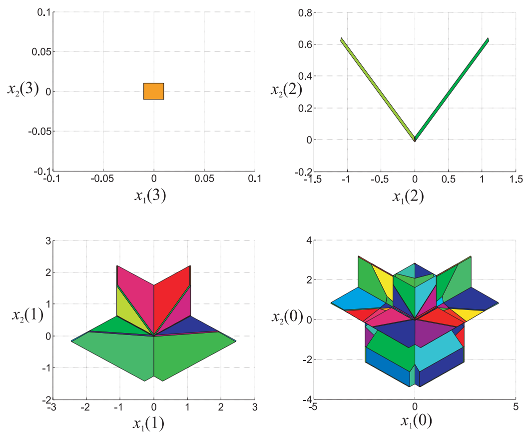


Fig. 8.6. State space partition corresponding to the state feedback finite time optimal control law $u^*(x(k))$ of system (8.26).

As explained in Section 8.8.1 the optimal control law is stored in a special data structure where:

1. The ordering of the regions is important.
2. The polyhedra can overlap.
3. The polyhedra can have an associated value function of multiplicity $s > 1$. Thus, s quadratic functions have to be compared on-line in order to compute the optimal control action.

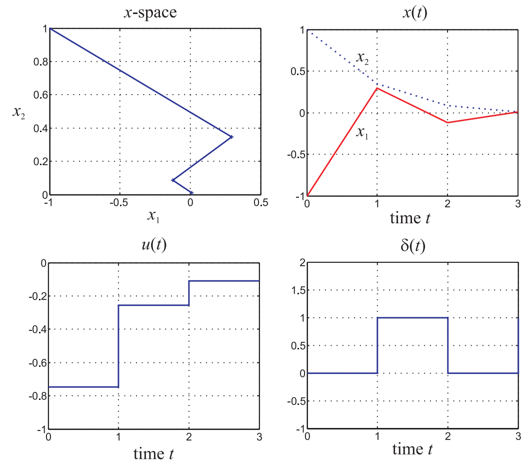


Fig. 8.7. Finite time optimal control of system (8.26).

8.9 Receding Horizon Control

Consider the problem of regulating to the origin the PWA system system (8.2). Receding Horizon Control (RHC) can be used to solve such a constrained regulation problem. The control algorithm is identical to the one outlined in Chapter 4 for linear systems. Assume that a full measurement of the state $x(t)$ is available at the current time t. Then, the finite time optimal control problem (8.3)-(8.8) (restated here)

$$\begin{array}{ll} J^*(x(t)) \triangleq \min_{\{U_t\}} & \|Px_N\|_p + \sum_{k=0}^{N-1} \|Q_1 u_{k,t}\|_p + \|Q_2\delta_k\|_p \\ & +\|Q_3 z_k\|_p + \|Q_4 x_k\|_p \\ \text{subj. to} & \left\{ \begin{array}{l} x_{k+1} = Ax_k + B_1 u_k + B_2\delta_k + B_3 z_k \\ E_2\delta_k + E_3 z_k \leq E_1 u_k + E_4 x_k + E_5 \\ x_N \in \mathcal{X}_f \\ x_0 = x(t) \end{array} \right. \end{array} \tag{8.56}$$

is solved at each time t, where $U_t = \{u_{0,t}, \ldots, u_{N-1,t}\}$ and where the time invariance of the plants permits the use of initial time 0 rather than t in the finite time optimal control problem. The sub-index t is used only to stress the fact that that finite time optimal control problem (8.56) is solved for $x_0 = x(t)$. Let $U_t^* = \{u_{0,t}^*, \ldots, u_{N-1,t}^*\}$ be the optimal solution of (8.56) at time t. Then, the first sample of U_t^* is applied to system (8.2):

$$u(t) = u_{0,t}^*. \tag{8.57}$$

The optimization (8.56) is repeated at time $t+1$, based on the new state $x_0 = x(t+1)$, yielding a *moving* or *receding horizon* control strategy.

Based on the results of previous sections the state feedback receding horizon controller (8.56)–(8.57) can be immediately obtained in two ways: (i) solve the MIQP (MILP) (8.21) for $x_0 = x(t)$ or (ii) by setting

$$u(t) = f_0^*(x(t)), \tag{8.58}$$

where $f_0^*(x(0)) : \mathbb{R}^n \to \mathbb{R}^{n_u}$ is the piecewise affine solution to the CFTOC (8.56) computed as explained in Section 8.8 (Section 8.7). Clearly, the explicit form (8.58) has the advantage of being easier to implement, and provides insight on the type of action of the controller in different regions of the state space.

8.9.1 Convergence

We remark that an infinite horizon formulation [96, 127, 19] would be inappropriate in the present hybrid context for both practical and theoretical reasons. In fact, approximating the infinite horizon with a large N is computationally prohibitive, as the number of possible switches (i.e., the combinations of 0-1 variables involved in the MIQP (MILP) (8.21)) depends exponentially on N. Moreover, the quadratic term in δ might oscillate, and hence "good" (i.e. asymptotically stabilizing) input sequences might be ruled out by a corresponding infinite value of the performance index. From a theoretical point of view, without any further restriction, an infinite dimensional problem is not meaningful. An oscillation of the quadratic term in δ could easily lead to an infinite cost associated to all possible input sequences. Even for meaningful problems it is not clear how to reformulate an infinite dimensional problem as a finite dimensional one for a PWA system, as can be done for linear systems through Lyapunov or Riccati algebraic equations.

In order to synthesize RHC controllers which inherently possess convergence guarantees, in [23] the authors adopted the standard stability constraint on the final state $\mathcal{X}_f = 0$.

Theorem 8.9. *Let $x_e = 0$, $u_e = 0$ be an equilibrium pair and assume that $\delta_e = 0$, $z_e = 0$ are the corresponding equilibrium points of the auxiliary variables. Assume that the initial state $x(0)$ is such that a feasible solution of problem (8.56) with $\mathcal{X}_f = 0$ exists at time $t = 0$. Then for all $Q_1 = Q_1' > 0$, $Q_2 = Q_2' \geq 0$, $Q_3 = Q_3' \geq 0$ and $Q_4 = Q_4' > 0$ if $p = 2$ (for all $Q_1 and Q_4$ full column rank if $p = 1, \infty$), the PWA system (8.2) controlled by the RHC (8.56)–(8.57) converges to the equilibrium, i.e.*

$$\lim_{t\to\infty} x(t) = 0$$
$$\lim_{t\to\infty} u(t) = 0$$
$$\lim_{t\to\infty} \|Q_2\delta(t)\|_p = 0$$
$$\lim_{t\to\infty} \|Q_3 z(t)\|_p = 0$$

Note that if $Q_2 > 0$ and $Q_3 > 0$ in the case $p = 2$ or if Q_2 and Q_3 are full column rank matrices in the case $p = 1, \infty$, convergence of $\delta(t)$ and $z(t)$ follows as well.

Proof: The proof is omitted here. It easily follows from standard Lyapunov arguments along the same lines of Theorem 4.3

□

The end-point constraint of Theorem 8.9 typically deteriorates the overall performance, especially for short prediction horizons. In order to avoid such a constraint, we can compute a controlled invariant set for the hybrid system (8.2) and force the final state $x(N)$ to belong to such a set while using as terminal weight a piecewise-linear or a piecewise-quadratic Lyapunov function for the system. We have discussed piecewise-quadratic and piecewise-linear Lyapunov functions for hybrid systems in Section 7.4. The computation of invariant sets for hybrid systems has been investigated in [97], and is a subject of current research in the hybrid systems community.

8.9.2 Extensions

All the formulations of predictive controllers for linear systems presented in Section 4.6 can be stated for hybrid systems. For example, in Chapter 10.3 a reference tracking predictive controller for a hybrid system is formulated. Also, the number of control degrees of freedom can be reduced to $N_u < N$, by setting $u_k \equiv u_{N_u-1}$, $\forall k = N_u, \dots, N$. However, while in Section 4.6 this can reduce the size of the optimization problem dramatically at the price of a reduced performance, here the computational gain is only partial, since all the N $\delta(k)$ and $z(k)$ variables remain in the optimization.

8.9.3 Examples

Example 8.2. Consider the problem of regulating the piecewise affine system (8.26) to the origin. The finite-time constrained optimal control (8.3)-(8.8) is solved for $p = 2$ with $N = 3$, $Q = I$, $R = 1$, $P = I$, $\mathcal{X}_f = \mathbb{R}^2$. Its state feedback solution (8.9) at time 0 $u^*(x(0)) = f_0^*(x(0))$ is implemented in a receding horizon fashion, i.e. $u(x(k)) = f_0^*(x(k))$. The state feedback control law consists of 48 polyhedral regions, and none of them has multiplicity higher than 1 (note that the enumeration of all possible switching sequences could lead to a multiplicity of 2^3 in all regions). The polyhedral regions are depicted in Figure 8.8 and in Figure 8.9 we show the corresponding closed-loop trajectories starting from the initial state $x(0) = [-1\ 1]'$.

Example 8.3. Consider the problem of regulating to the origin the piecewise affine system (8.26). The finite-time constrained optimal control (8.3)-(8.8) is solved for $p = \infty$ with $N = 3$, $P = Q = \left[\begin{smallmatrix} 50 & 0 \\ 0 & 50 \end{smallmatrix}\right]$, $R = 0.001$, $\mathcal{X}_f = \mathbb{R}^2$. Its state feedback solution (8.17) at time 0 $u^*(x(0)) = f_0^*(x(0))$ is implemented in a receding horizon fashion, i.e. $u(x(k)) = f_0^*(x(k))$. The state feedback control

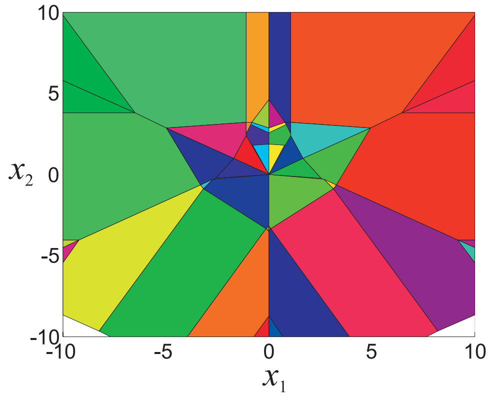


Fig. 8.8. State space partition corresponding to the state feedback receding horizon control law $u(x)$ of system (8.26).

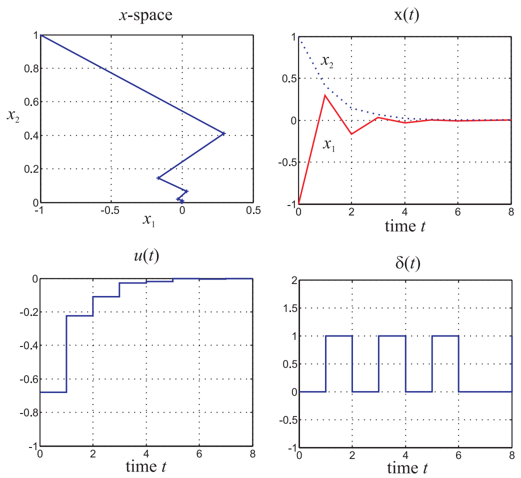


Fig. 8.9. Receding horizon control of system (8.26)

law consists of 35 polyhedral regions depicted in Figure 8.10. In Figure 8.11 we show the corresponding closed-loop trajectories starting from the initial state $x(0) = [-1\ 1]'$.

Example 8.4. Consider the following hybrid control problem for the heat exchange example proposed by Hedlund and Rantzer [82]. The temperature of two furnaces should be controlled to a given set-point by alternate heating. Only three operational modes are allowed: heat only the first furnace, heat only the second one, do not heat. The amount of power u_0 to be fed to the furnaces at each time instant is fixed. The system is described by the following equations:

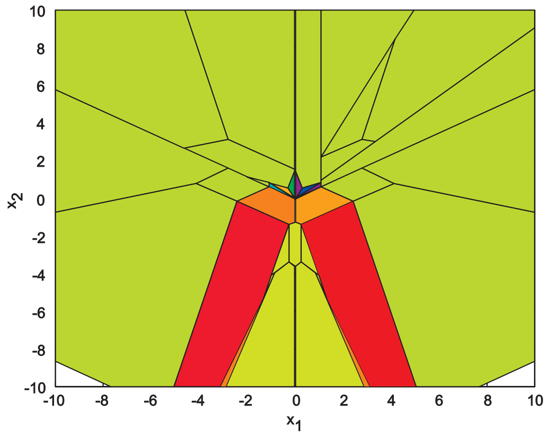


Fig. 8.10. State space partition corresponding to the state feedback receding horizon control law $u(x)$ of system (8.26).

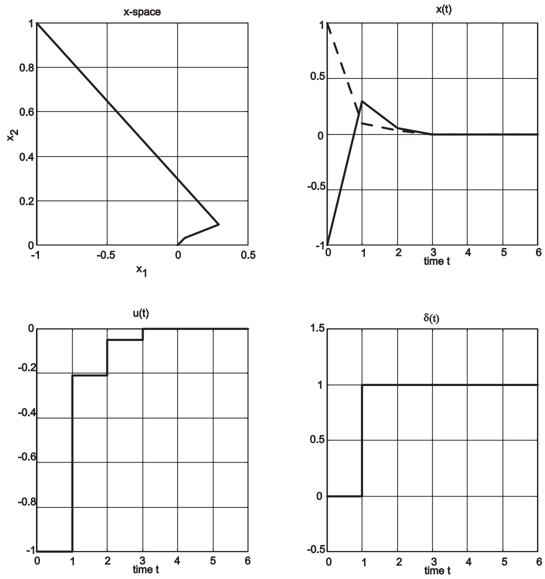
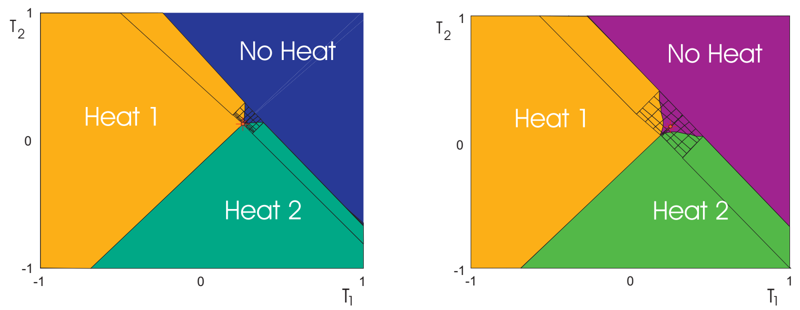


Fig. 8.11. Receding horizon control of system (8.26)



(a) State space partition for $u_0 = 0.4$

(b) State space partition for $u_0 = 0.8$

Fig. 8.12. State space partition corresponding to the control law $u^*(x)$, solution of the optimal control problem (8.59)–(8.60).

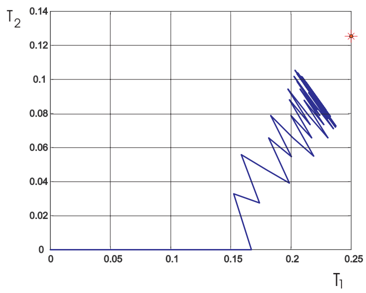
$$\begin{cases} \dot{T} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} T + u_0 u \\ u = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ if heating the first furnace} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ if heating the second furnace} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ if no heating} \end{cases} \end{cases} \tag{8.59}$$

System (8.59) is discretized with a sampling time $T_s = 0.08s$, and its equivalent MLD form (7.8) is computed as described in [23] by introducing an auxiliary vector $z(t) \in \mathbb{R}^9$.

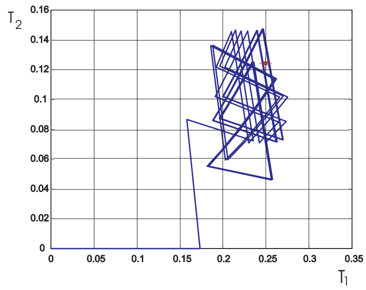
In order to optimally control the two temperatures to the desired values $T_e^1 = 1/4$ and $T_e^2 = 1/8$, the following performance index is minimized at each time step t:

$$\min_{U_t} J(U_t, x(t)) \triangleq \sum_{k=0}^{2} \|R(u_{k+1,t} - u_{k,t})\|_\infty + \|Q(T_{k+1} - T_e)\|_\infty \tag{8.60}$$

subject to the MLD system dynamics, along with the weights $Q = 1$, $R = 700$. The cost function penalizes the tracking error and trades it off with the number of input switches occurring along the prediction horizon. By solving the mp-MILP associated with the optimal control problem we obtain the state feedback controller for the range $T \in [-1, 1] \times [-1, 1]$, $u_0 \in [0, 1]$. In Fig. 8.12 two slices of the three-dimensional state-space partition for different constant power levels u_0 are depicted. Note that around the equilibrium the solution is more finely partitioned. The resulting optimal trajectories are shown in Fig. 8.13. For a low power $u_0 = 0.4$ the set point is never reached.



(a) Receding horizon control of system (8.59) for $u_0 = 0.4$



(b) Receding horizon control of system (8.59) for $u_0 = 0.8$

Fig. 8.13. Receding horizon control of system (8.59)

Part IV

Applications

Introduction

In collaboration with different companies and institutes, the results described in the previous chapters have been applied to a wide range of problems.

Constrained optimal control for linear system was implemented for the control of blood pressure in anesthesia [74], of a mechanical system available at our laboratory named "Ball and Plate" [118, 114, 4, 53], of the gear shift/clutch operation on automotive vehicles [15] and of active automotive suspension [155].

Constrained optimal control for hybrid systems was used for the control of Co-generation Power Plants [61, 60], of a benchmark multi-tank system [115], of a gas supply system [23], of a gear shift operation on automotive vehicles [150], of the direct injection stratified charge engine [22], of the integrated management of the power-train [108] and of a traction control problem [36].

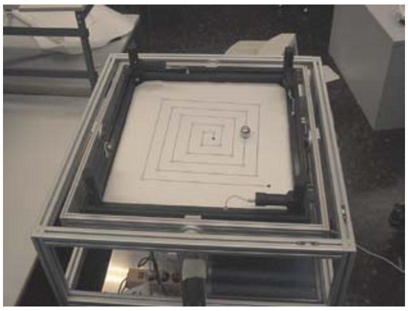
All the investigated theoretical problems are "hard" in the mathematical sense, which implies - loosely speaking - that in the worst case the computational effort grows exponentially with the problem size. Thus the future challenge will be to develop approximate methods which provide good, if not optimal answers for problems with specific structures and where the computational effort grows only in a polynomial fashion. Otherwise the applicability of the developed tools will be remain limited to small problems.

In the next two chapters we will describe into details two practical experiment: a"Ball and Plate" system available at our Laboratory [118, 114, 4, 53] and a traction control problem currently under investigation at Ford Research Laboratories.

9

Ball and Plate

(a) Photo 1



(b) Photo2

Fig. 9.1. "Ball and Plate" Photos

The "Ball & Plate" is a mechanical system available at the Automatic Control Laboratory of ETH-Zurich. Two pictures of the system can be seen in Figure 9.1(a) and Figure 9.1(b).

The experiment consists of a ball rolling over a gimbal-suspended plate actuated by two independent motors. The plate is equipped with four laser sensors that measure the position of the ball. The control objective is to control the ball position and to let it follow given trajectories on the plate.

The presence of several constraints and the fast sampling time required for the control task make the experiment a good benchmark test for the constrained optimal control theory presented in the previous chapters.

The design, modeling, identification and control of the system was carried out throughout several students projects [118, 114, 4, 53] at the Automatic Control Laboratory. In the following we will presents only a brief summary of the results.

9.1 Ball and Plate Dynamic Model

The nonlinear model of the "Ball and Plate" system is rather complex and was first studied in [87]. The system in sketched in Figure 9.2. X and Y represent the plate fixed Cartesian axes. The plate is actuated by two independent

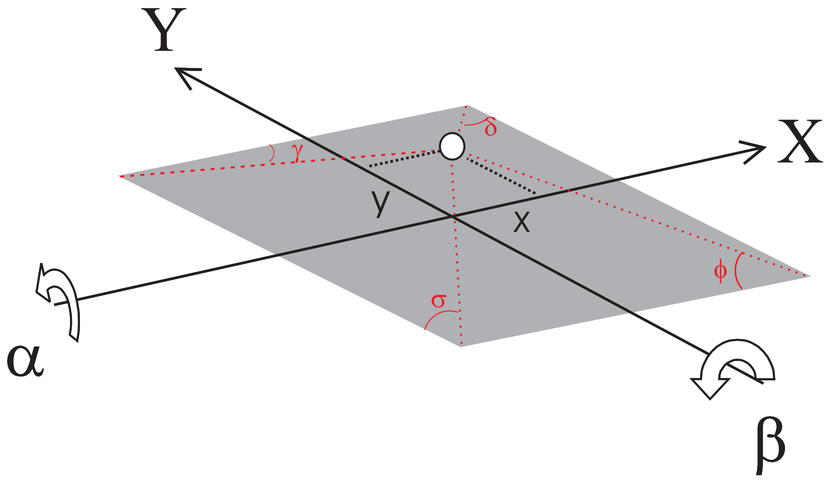


Fig. 9.2. Ball and Plate System

motors that rotate it of an angle α around the X axis and β around the Y axis. The position x, y of the ball is indirectly measured through four rotating laser beams positioned in the four corners of the plate and four corresponding sensors. The angles δ, σ, ϕ and γ are measured by measuring the time interval between the initial time of a laser rotation and the instant when the laser light is reflected by the ball. The relation between δ, σ, ϕ and γ and x, y is algebraic. The system is also equipped with two potentiometers that measure the angles α and β of the plate.

A good approximate linear model of the system is given by the following linear time invariant systems for the x-axis and y-axis

$$\begin{cases} \dot{x}_\beta = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -700 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 33.18 \end{bmatrix} x_\beta + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3.7921 \end{bmatrix} u_\beta \\ y_\beta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_\beta \end{cases} \tag{9.1}$$

$$\begin{cases} \dot{x}_\alpha = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 700 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -34.69 \end{bmatrix} x_\alpha + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3.1119 \end{bmatrix} u_\alpha \\ y_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_\alpha \end{cases} \tag{9.2}$$

where $x_\beta = [x, \dot{x}, \beta, \dot{\beta}]$ and $x_\alpha = [y, \dot{y}, \alpha, \dot{\alpha}]$, where x [cm] and $\dot{x}$ [cm/s] are the position and the velocity of the ball with respect to the x-coordinate, β

[rad] and $\dot{\beta}$ [rad/s] are the angular position and the angular velocity of the plate around the y-axis, y [cm] and $\dot{y}$ [cm/s] are the position and the velocity of the ball with respect to the y-coordinate, α [rad] and $\dot{\alpha}$ [rad/s] are the angular position and the angular velocity of the plate around the x-axis.

The input u_α and u_β are the voltage to the actuators that make the plate rotate around the X and Y axis. The zero state of models (9.2)-(9.2) represents the system standing still with a ball stable in the center position of the plate. Note that in the linearized models (9.2)-(9.2) of the "Ball and Plate", the dynamics of the ball along the X-axis are decoupled from the dynamics along the Y-axis.

The system is subject to the following design constraints:

$$\begin{array}{l} u_{\max} = -u_{\min} = 10 \text{ V} \\ x_{\max} = -x_{\min} = 30 \text{ cm} \\ y_{\max} = -y_{\min} = 30 \text{ cm} \\ \alpha_{max} = -\alpha_{min} = 0.26 \text{ rad} \\ \beta_{max} = -\beta_{min} = 0.26 \text{ rad} \end{array} \tag{9.3}$$

9.2 Constrained Optimal Control

Models (9.1)-(9.2) are discretized by exact sampling (sampling period $T_s = 30$ ms), to obtain

$$x_\alpha(t+1) = A_\alpha x_\alpha(t) + B_\alpha u_\alpha(t) \tag{9.4}$$

$$y_\alpha(t) = C_\alpha x_\alpha(t) \tag{9.5}$$

and

$$x_\beta(t+1) = A_\beta x_\beta(t) + B_\beta u_\beta(t) \tag{9.6}$$

$$y_\beta(t) = C_\beta x_\beta(t) \tag{9.7}$$

To control the position of the ball on the plate, a reference tracking receding horizon optimal control controller is designed for the X-axis

$$\begin{array}{ll} \min\limits_{\Delta U \triangleq \{\delta u_{\beta,0},\ldots,\delta u_{\beta,N_u-1}\}} & \left\{\sum_{k=0}^{N_y-1} \|Q(y_{\beta,k} - r_\beta)\|_2 + \|R\delta u_{\beta,k}\|_2\right\} \\ \text{subj. to} & y_{\min\beta} \le y_{\beta,k} \le y_{\max\beta},\ k = 1,\ldots,N_c \\ & u_{\min} \le u_{\beta,k} \le u_{\max},\ k = 0,1,\ldots,N_c \\ & x_{\beta,k+1} = A_\beta x_{\beta,k} + B_\beta u_{\beta,k},\ k \ge 0 \\ & y_{\beta,k} = C_\beta x_{\beta,k},\ k \ge 0 \\ & u_{\beta,k} = u_{\beta,k-1} + \delta u_{\beta_k},\ k \ge 0 \\ & \delta u_{\beta,k} = 0,\ k \ge N_u \end{array} \tag{9.8}$$

and an identically tuned optimal controller controller for the Y-axis

$$
\begin{array}{ll}
\min\limits_{\Delta U \triangleq \{\delta u_{\alpha,0},\ldots,\delta u_{\alpha,N_u-1}\}} & \left\{\sum_{k=0}^{N_y-1} \|Q(y_{\alpha,k}-r_\alpha)\|_2 + \|R\delta u_{\alpha,k}\|_2\right\} \\
\text{subj. to} & y_{\min\alpha} \leq y_{\alpha,k} \leq y_{\max\alpha},\ k=1,\ldots,N_c \\
& u_{\min} \leq u_{\alpha,k} \leq u_{\max},\ k=0,1,\ldots,N_c \\
& x_{\alpha,k+1} = A_\alpha x_{\alpha,k} + B_\alpha u_{\alpha,k},\ k \geq 0 \\
& y_{\alpha,k} = C_\alpha x_{\alpha,k},\ k \geq 0 \\
& u_{\alpha,k} = u_{\alpha,k-1} + \delta u_{\alpha,k},\ k \geq 0 \\
& \delta u_{\alpha,k} = 0,\ k \geq N_u
\end{array}
\tag{9.9}
$$

A Kalman filter is designed in order to estimate $\dot{x}, \dot{y}, \dot{\alpha}, \dot{\beta}$ and filter the noise from the measurements.

9.2.1 Tuning

In the first phase the parameters are tuned in simulation until the desired performance is achieved by using Simulink and the MPC Toolbox [26].

The parameters to be tuned are the prediction horizons N_y, the number of free moves N_u, the constraint horizon N_c and the weights Q and R. The trade-off between fast tracking of the ball and smooth plate dynamics is easily adjustable by appropriately choosing the weights Q and R. N_y, N_u, N_c are the result of a trade off between controller performance and its complexity in terms of number of regions resulting from the PPWA solution.

The state feedback PPWA solutions of optimal control problems (9.8)-(9.9) are computed for the following sets of parameters

$$
N_y = 10,\ N_u = 1,\ N_c = 2,\ R = 1,\ Q = 2, \tag{9.10}
$$

and consist of 22/23 regions (x-axis/y-axis).

Note that the number of states of the state feedback PPWA control law is seven for each axis. In fact, we have to add the two reference signals r_α (r_β) and the previous input $u_\alpha(t-1)$ ($u_\beta(t-1)$) to the four states of model (9.1) (model 9.2).

9.3 Experimental Setup

The control software for the "Ball and Plate" system was implemented in the Matlab/Simulink environment. The algorithm to compute the polyhedral regions and the corresponding affine gains of the state feedback optimal control law was coded in Matlab, while the algorithm to implement the optimal control law (a simple look-up table) was coded in C as Simulink S-Functions. The Real-Time-Workshop and xPC packages were then used to translate and compile the Simulink model with S-Functions into a real-time executable program. This program was downloaded via a TCP/IP link from the host PC (used for software development) to the target PC, which executed the real-time program. The target PC also contained the A/D cards necessary to read

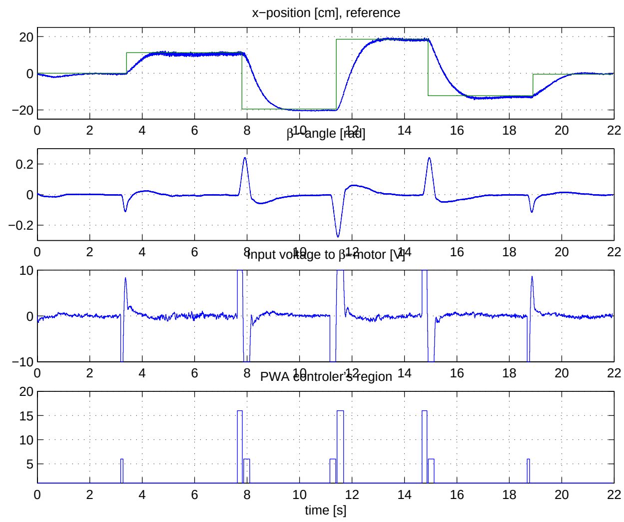


Fig. 9.3. Closed-loop simulation of the "Ball and Plate" system controlled with the PPWA solution of the MPC law (9.8)–(9.10)

the position of the ball and angles of the plate, and the D/A cards that controlled the DC motor. The real-time program executed at the sampling time of 30 ms.

We tested the system with a variety of ramp, step and sinusoidal reference signals. Experiments were carried on with different PPWA optimal controllers ranging from small number of polyhedral regions (22 regions for each axis) to high number of polyhedral regions (up to 500 regions for each axis). Figures 9.3 and 9.4 illustrate the results of an experiment were the "Ball and Plate" was controlled with optimal control laws (9.8)–(9.9) with the tuning (9.10). The upper plots represent the x and y positions of the ball and the corresponding references. The second plots from the top represent the angular positions α and β of the plate. The third plots from the top represent the commands to the motors and the plots at the bottom represent the polyhedral regions active at different time instants. One can observe that when the region zero is active when the system is at steady state. Region zero corresponds to the unconstrained LQR.

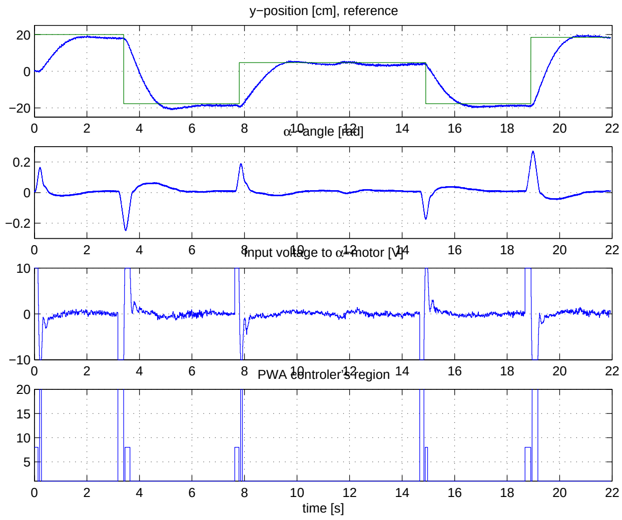


Fig. 9.4. Closed-loop simulation of the "Ball and Plate" system controlled with the PPWA solution of the MPC law (9.9)–(9.10

10

Traction Control

In this chapter we describe a hybrid model and an optimization-based control strategy for solving a traction control problem. The problem is tackled in a systematic way from modeling to control synthesis and implementation by using the results and tools developed at the Automatic Control Laboratory of Zurich over the last two years and largely described in the previous chapters. The model is described first in the language HYSDEL (HYbrid Systems DEscription Language) to obtain a mixed-logical dynamical (MLD) hybrid model of the open-loop system. For the resulting MLD model we design a receding horizon finite-time optimal controller. The resulting optimal controller is converted to its equivalent piecewise affine form by employing multiparametric mixed-integer linear programming techniques, and finally experimentally tested on a car prototype. Experiments show that good and robust performance is achieved in a limited development time, by avoiding the design of ad-hoc supervisory and logical constructs usually required by controllers developed according to standard techniques.

10.1 Introduction

For more than a decade advanced mechatronic systems controlling some aspects of vehicle dynamics have been investigated and implemented in production [90, 66]. Among them, the class of traction control problems is one of the most studied. Traction controllers are used to improve a driver's ability to control a vehicle under adverse external conditions such as wet or icy roads. By maximizing the tractive force between the vehicle's tire and the road, a traction controller prevents the wheel from slipping and at the same time improves vehicle stability and steerability. In most control schemes the wheel slip, i.e., the difference between the vehicle speed and the speed of the wheel is chosen as the controlled variable. The objective of the controller is to maximize the tractive torque while preserving the stability of the system. The relation between the tractive force and the wheel slip is nonlinear and is a function of the road condition [8]. Therefore, the overall control scheme is composed of two parts: a device that estimates the road surface condition, and a traction controller that regulates the wheel slip at the desired value. Regarding the second part, several control strategies have been proposed in the literature mainly based on sliding-mode controllers, fuzzy logic and adaptive schemes [13, 95, 11, 143, 144, 109, 8, 142]. Such control schemes are motivated by the fact that the system is nonlinear and uncertain.

The presence of nonlinearities and constraints on one hand, and the simplicity needed for real-time implementation on the other, have discouraged the design of optimal control strategies for this kind of problem. In Chapter 7 we have described a new framework for modeling hybrid systems proposed in [23] and in Chapter 8 an algorithm to synthesize piecewise linear optimal controllers for such systems. In this chapter we describe how the hybrid framework and the optimization-based control strategy can be successfully applied

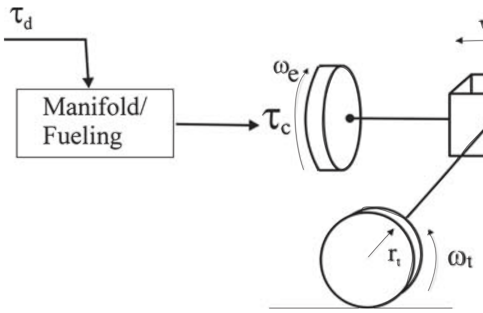


Fig. 10.1. Simple vehicle model

for solving the traction control problem in a systematic way. The language HYSDEL is first used to describe a linear hybrid model of the open-loop system suitable for control design. Such a model is based on a simplified model and a set of parameters provided by Ford Research Laboratories, and involves piecewise linearization techniques of the nonlinear torque function that are based on hybrid system identification tools [63]. Then, an optimal control law is designed and transformed to an equivalent piecewise affine function of the measurements, that is easily implemented on a car prototype. Experimental results show that good and robust performance is achieved.

A mathematical model of the vehicle/tire system is introduced in Section 10.2. The hybrid modeling and the optimal control strategy are discussed in Section 10.3. In Section 10.4 we derive the piecewise affine optimal control law for traction control and in Section 10.6 we present the experimental setup and the results obtained.

10.2 Vehicle Model

The model of the vehicle used for the design of the traction controller is depicted in Figure 10.1, and consists of the equations

$$\begin{pmatrix} \dot{\omega}_e \\ \dot{v}_v \end{pmatrix} = \begin{pmatrix} -\frac{b_e}{J'_e} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_e \\ v_v \end{pmatrix} + \begin{pmatrix} \frac{1}{J'_e} \\ 0 \end{pmatrix} \tau_c + \begin{pmatrix} -\frac{1}{J'_e g_r} \\ -\frac{1}{m_v r_t} \end{pmatrix} \tau_t \tag{10.1}$$

with

$$\tau_c(t) = \tau_d(t - \tau_f) \tag{10.2}$$

where the involved physical quantities and parameters are described in Table 10.1.

Model (10.1) contains two states for the mechanical system downstream of the manifold/fueling dynamics. The first equation represents the wheel dynamics under the effect of the combustion torque and of the traction torque,

ω_e	Engine speed	r_t	Tire radius
v_v	Vehicle speed	τ_c	Actual combustion torque
J_e'	Combined engine/wheel inertia	τ_d	Desired combustion torque
b_e	Engine damping	τ_t	Frictional torque on the tire
g_r	Total driveline gear ratio between ω_e and v_v	μ	Road coefficient of friction
m_v	Vehicle mass	τ_f	Fueling to combustion pure delay period
$\Delta\omega$	Wheel slip		

Table 10.1. Physical quantities and parameters of the vehicle model

while the second one describes the longitudinal motion dynamics of the vehicle. In addition to the mechanical equations (10.1) the air intake and fueling model (10.2) also contributes to the dynamic behaviour of the overall system. For simplicity, since the actuator will use just the spark advance, the intake manifold dynamics is neglected and the fueling combustion delay is modeled as a pure delay. Both states are indirectly measured through measurements of front and rear wheel speeds: assuming we are modeling a front wheel driven vehicle, ω_e is estimated from the speed of the front wheel, while v_v is estimated from the speed of the rear wheel. The slip $\Delta\omega$ of the car is defined as the difference between the normalized vehicle and engine speeds:

$$\Delta\omega = \frac{v_v}{r_t} - \frac{\omega_e}{g_r}. \tag{10.3}$$

The frictional torque τ_t is a nonlinear function of the slip $\Delta\omega$ and of the road coefficient of friction μ

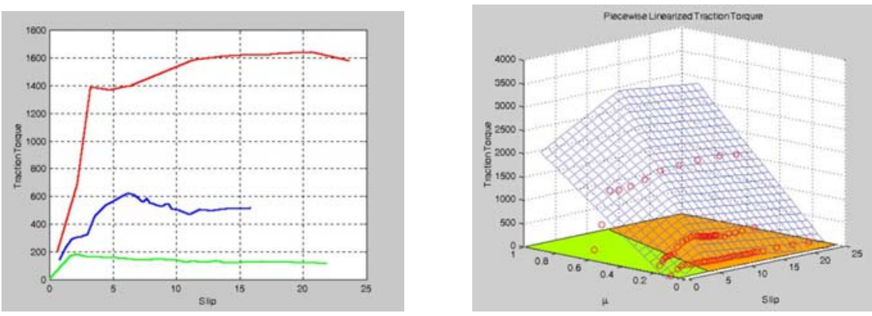
$$\tau_t = f_\tau(\Delta\omega, \mu). \tag{10.4}$$

The road coefficient of friction μ depends on the road-tire conditions, while the function f_τ depends on vehicle parameters such as the mass of the vehicle, the location of the center of gravity and the steering and suspension dynamics [142]. Figure 10.2(a) shows a typical curves $\tau_t = f(\Delta\omega)$ for three different road conditions (ice, snow and pavement).

10.2.1 Discrete-Time Hybrid Model of the Vehicle

The model obtained in Section 10.2 is transformed into an equivalent discrete-time MLD model through the following steps:

1. The frictional torque τ_t is approximated as a piecewise linear function of the slip $\Delta\omega$ and of the road coefficient of friction μ by using the approach described in [63]. The algorithm proposed in [63] generates a polyhedral partition of the $(\Delta\omega, \mu)$-space and the corresponding affine approximation of the torque τ_t in each region.
   If the number of regions is limited to two we get:



(a) Measured tire torque τ_t for three different road conditions

(b) Piecewise linear model of the tire torque τ_t

Fig. 10.2. Nonlinear behaviour and its piecewise-linear approximation of the traction torque τ_t as a function of the slip $\Delta\omega$ and road adhesion coefficient μ

$$\tau_t(\Delta\omega,\mu) = \begin{cases} 67.53\Delta\omega + 102.26\mu - 31.59 & \text{if } 0.21\Delta\omega - 5.37\mu \leq -0.61 \\ -1.85\Delta\omega + 1858.3\mu - 232.51 & \text{if } 0.21\Delta\omega - 5.37\mu > -0.61 \end{cases} \tag{10.5}$$

as depicted in Figure 10.2(b).

2. Model (10.1) is discretized with sampling time $T_s = 20$ ms and the PWA model (10.5) of the frictional torque is used to obtain the following discrete time PWA model of the vehicle:

$$\tilde{x}(t+1) = \begin{cases} \left[\begin{smallmatrix} 0.98 & 0.78 \\ 2e-4 & 0.98 \end{smallmatrix}\right] \tilde{x}(t) + \left[\begin{smallmatrix} 0.04 \\ 5e\text{-}6 \end{smallmatrix}\right] \tau_c(t) + \left[\begin{smallmatrix} -0.35 \\ 4e-3 \end{smallmatrix}\right] \mu(t) + \left[\begin{smallmatrix} 0.10 \\ -1e-3 \end{smallmatrix}\right] \\ \text{if } \left[\, 0.21\Delta\omega - 5.37\mu \leq -0.61 \,\right] \\ \\ \left[\begin{smallmatrix} 1.00 & -0.02 \\ -6e\text{-}6 & 1.00 \end{smallmatrix}\right] \tilde{x}(t) + \left[\begin{smallmatrix} 0.04 \\ -1e\text{-}7 \end{smallmatrix}\right] \tau_c(t) + \left[\begin{smallmatrix} -6.52 \\ 0.08 \end{smallmatrix}\right] \mu(t) + \left[\begin{smallmatrix} 0.81 \\ -0.01 \end{smallmatrix}\right] \\ \text{if } \left[\, 0.21\Delta\omega - 5.37\mu > -0.61 \,\right] \end{cases} \tag{10.6}$$

where $\tilde{x} = \left[\begin{smallmatrix} \omega_e \\ v_v \end{smallmatrix}\right]$. At this stage τ_c is considered as the control input to the system. The time delay between τ_c and τ_d will be taken into account in the controller design phase detailed in Subsection 10.4.2.

3. The following constraints on the torque, on its variation, and on the slip need to be satisfied:

$$\tau_c \leq 176 \text{ Nm} \tag{10.7}$$

$$\tau_c \geq -20 \text{ Nm} \tag{10.8}$$

$$\dot{\tau}_c \approx \frac{\tau_c(k) - \tau_c(k-1)}{T_s} \leq 2000 \text{ Nm/s} \tag{10.9}$$

$$\Delta\omega \geq 0. \tag{10.10}$$

In order to constrain the derivative of the input, the state vector is augmented by including the previous torque $\tau_c(t-1)$. The variation of the combustion torque $\Delta\tau_c(t) = \tau_c(t) - \tau_c(t-1)$ will be the new input variable.

The resulting hybrid discrete-time model has three states (x_1 = previous τ_c, $x_2 = \omega_e$, $x_3 = v_v$), one control input ($u_1 = \Delta\tau_c$), one uncontrollable input ($u_2 = \mu$), one regulated output ($y = \Delta\omega$), one auxiliary binary variable $\delta \in \{0, 1\}$ indicating the affine region where the system is operating, $[\delta = 0] \iff [0.21\Delta\omega - 5.37\mu \leq -0.61]$, and two auxiliary continuous variables $z \in \mathbb{R}^2$ describing the dynamics in (10.6), i.e.,

$$z = \begin{cases} A_1\tilde{x} + B_{12}\mu + f_1 & \text{if } \delta = 0 \\ A_2\tilde{x} + B_{22}\mu + f_2 & \text{otherwise,} \end{cases}$$

where A_1, A_2, B_{11}, B_{12}, B_{21}, B_{22}, f_1, f_2 are the matrices in (10.6). The resulting MLD model is obtained by processing the description list reported in Section 10.7 through the HYSDEL compiler:

$$x(t+1) = \begin{bmatrix} 0 & 0 & 0 \\ 0.0484 & 0 & 0 \\ 0.0897 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1.0000 & 0 \\ 0.0484 & 0 \\ 0.0897 & 0 \end{bmatrix} \begin{bmatrix} \Delta\tau_c(t) \\ \mu(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} z(t) \tag{10.11a}$$

$$E_2\delta(t) + E_3 z(t) \leq E_1 \begin{bmatrix} \Delta\tau_c(t) \\ \mu(t) \end{bmatrix} + E_4 x(t) + E_5 \tag{10.11b}$$

where

$$E_2 = 10^3 \begin{bmatrix} 1.1 \\ 1.1 \\ -1.1 \\ -1.1 \\ 0.18 \\ 0.18 \\ -0.18 \\ -0.18 \\ -0.40 \\ 0.40 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad E_3 = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad E_1 = \begin{bmatrix} 0 & 0.35 \\ 0 & -0.35 \\ 0 & 6.52 \\ 0 & -6.52 \\ 0 & -0.004 \\ 0 & 0.004 \\ 0 & -0.08 \\ 0 & 0.08 \\ 0 & 0.21 \\ 0 & -0.21 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$E_4 = \begin{bmatrix} 0 & -0.98 & -0.78 \\ 0 & 0.98 & 0.78 \\ 0 & -1.00 & 0.02 \\ 0 & 1.00 & -0.02 \\ 0 & -0.00 & -0.98 \\ 0 & 0.00 & 0.98 \\ 0 & 0.00 & -1.00 \\ 0 & -0.00 & 1.00 \\ 0 & -0.38 & 18.04 \\ 0 & 0.38 & -18.04 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0.07 & -3.35 \end{bmatrix} \quad E_5 = 10^3 \begin{bmatrix} 1.10 \\ 1.10 \\ -0.00 \\ 0.00 \\ 0.18 \\ 0.18 \\ 0.00 \\ -0.00 \\ 0.00 \\ 0.39 \\ 2.00 \\ 0.17 \\ 0.02 \end{bmatrix} \tag{10.11c}$$

Note that the constraints in (10.10) are included in the system matrices (10.11c).

10.3 Constrained Optimal Control

Figure 10.3 depicts the lateral and longitudinal traction torque as a function of the wheel slip. It is clear that if the wheel slip increases beyond a certain value,

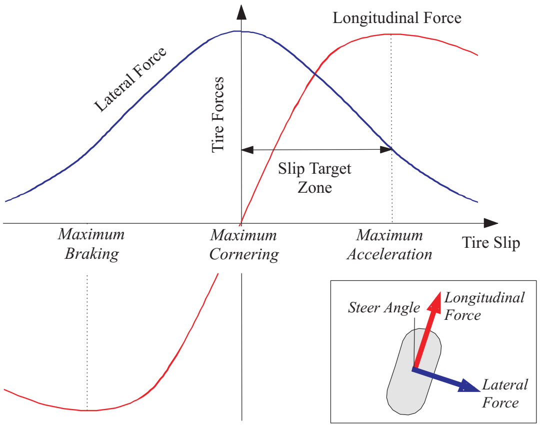


Fig. 10.3. Typical behaviour of lateral and longitudinal tire forces

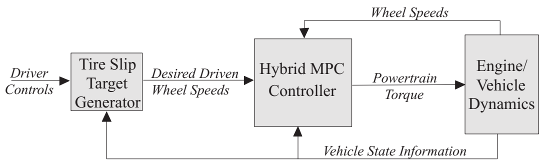


Fig. 10.4. Overall traction control scheme. In this paper we focus on the design of the hybrid MPC controller

the longitudinal and lateral driving forces on the tire decrease considerably and the vehicle cannot speed up and steer as desired.

By maximizing the tractive force between the vehicle's tire and the road, a traction controller prevents the wheel from slipping and at the same time improves vehicle stability and steerability. The overall control scheme is depicted in Figure 10.4 and is composed of two parts: a device that estimates the road surface condition μ and consequently generates a desired wheel slip $\Delta\omega_d$, and a traction controller that regulates the wheel slip at the desired value $\Delta\omega_d$. In this paper we only focus on the second part, as the first one is already available from previous projects at Ford Research Laboratories.

The control system receives the desired wheel slip $\Delta\omega_d$, the estimated road coefficient adhesion μ, the measured front and rear wheel speeds as input and generates the desired engine torque τ_c (the time delay between τ_c and τ_d will be compensated a posteriori as described in Section 10.4.2). A receding horizon controller can be designed for the posed traction control problem. At each time step t the following finite horizon optimal control problem is solved:

$$\min_{V} \sum_{k=0}^{N-1} |Q(\Delta\omega_k - \Delta\omega_d(t))| + |R\Delta\tau_{c,k}| \tag{10.12}$$

$$\text{subj. to} \begin{cases} x_{k+1} = Ax_k + B_1 \begin{bmatrix} \Delta\tau_{c,k} \\ \mu(t) \end{bmatrix} + B_2\delta_k + B_3 z_k \\ E_2\delta_k + E_3 z_k \leq E_1 \begin{bmatrix} \Delta\tau_{c,k} \\ \mu(t) \end{bmatrix} + E_4 x_k + E_5 \\ x_0 = x(t) \\ \Delta\tau_{c,k} = \tau_{c,k} - \tau_{c,k-1},\ k = 0,\dots,N-1, \quad \tau_{c,-1} \triangleq \tau_c(t-1) \end{cases} \tag{10.13}$$

where matrices $A, B_1, B_2, B_3, E_2, E_3, E_1, E_4, E_5$ are given in (10.11), and $V \triangleq [\Delta\tau_{c,0},\dots,\Delta\tau_{c,N-1}]'$ is the optimization vector. Note that the optimization variables are the torque variations $\Delta\tau_{c,k} = \tau_{c,k} - \tau_{c,k-1}$, and that the set point $\Delta\omega_d(t)$ and the current road coefficient of adhesion $\mu(t)$ are considered constant over the horizon N.

As explained in Section 8.5, problem (10.12)-(10.13) is translated into a mixed integer linear program of the form:

$$\min_{\mathcal{E},V,Z,\delta} \sum_{k=0}^{N-1} \varepsilon_k^w + \varepsilon_k^u \tag{10.14a}$$

$$\text{subj. to} \quad G^\varepsilon \mathcal{E} + G^u V + G^Z z + G^\delta \delta \leq S + F \begin{bmatrix} \omega_e(t) \\ v_v(t) \\ \tau_c(t-1) \\ \mu(t) \\ \Delta\omega_d(t) \end{bmatrix}, \tag{10.14b}$$

where $Z = [z_0',\dots,z_{N-1}']' \in \mathbb{R}^{2N}$, $\delta = [\delta_0,\dots,\delta_{N-1}]' \in \{0,1\}^N$ and $\mathcal{E} = [\varepsilon_0^w,\dots,\varepsilon_{N-1}^w,\varepsilon_0^u,\dots,\varepsilon_{N-1}^u]' \in \mathbb{R}^{2N}$ is a vector of additional slack variables introduced in order to translate the cost function (10.12) into the linear cost function (10.14a). Matrices $G^\varepsilon, G^u, G^z, G^\delta, S, F$ are matrices of suitable dimension that can be constructed from Q, R, N and A, B_1, B_2, B_3, E_2, E_3, E_1, E_4, E_5. The resulting control law is

$$\tau_c(t) = \tau_c(t-1) + \Delta\tau_{c,0}^*, \tag{10.15}$$

where $V^* \triangleq [\Delta\tau_{c,0}^*,\dots,\Delta\tau_{c,N-1}^*]'$ denotes the sequence of optimal input increments computed at time t by solving (10.14) for the current measurements $\omega_e(t)$, $v_v(t)$, set point $\Delta\omega_d(t)$, and estimate of the road coefficient $\mu(t)$.

10.4 Controller Design

The design of the controller is performed in two steps. First, the optimal control law (10.12)-(10.13) is tuned in simulation until the desired performance is achieved. The receding horizon controller is not directly implementable, as it would require the MILP (10.14) to be solved on-line, which is clearly prohibitive on standard automotive control hardware. Therefore, for implementation, in the second phase the PPWA solution of problem (10.12)-(10.13) is computed off-line by using the multiparametric mixed integer program solver described in Section 1.5. The resulting control law has the piecewise affine form

$$\tau_c(t) = F^i\theta(t) + g^i \quad \text{if} \quad H^i\theta(t) \leq K^i,\ i = 1, \ldots, N_r, \tag{10.16}$$

where $\theta(t) = [\omega_e(t)\ v_v(t)\ \tau_c(t-1)\ \mu(t)\ \Delta\omega_d(t)]'$. Therefore, the set of states + references is partitioned into N_r polyhedral cells, and an affine control law is defined in each one of them. The control law can be implemented on-line in the following simple way: (i) determine the i-th region that contains the current vector $\theta(t)$ (current measurements and references); (ii) compute $u(t) = F^i\theta(t) + g^i$ according to the corresponding i-th control law.

10.4.1 Tuning

The parameters of the controller (10.12)-(10.13) to be tuned are the horizon length N and the weights Q and R. By increasing the prediction horizon N the controller performance improves, but at the same time the number of constraints in (10.13) increases. As in general the complexity of the final piecewise affine controller increases dramatically with the number of constraints in (10.13), tuning N amounts to finding the smallest N which leads to a satisfactory closed-loop behaviour. Simulations were carried out to test the controller against changes to model parameters. A satisfactory performance was achieved with $N = 5, Q = 50, R = 1$ which corresponds to a PPWA controller consisting of 137 regions.

10.4.2 Combustion Torque Delay

The vehicle model in Section 10.2 is affected by a time delay of $\sigma = \frac{\tau_f}{T_s} = 12$ sampling intervals between the desired commanded torque τ_d and the combustion torque τ_c. To avoid the introduction of σ auxiliary states in the hybrid model (7.8), we take such a delay into account only during implementation of the control law.

Let the current time be $t \geq \sigma$ and let the explicit optimal control law in (10.16) be denoted as $\tau_c(t) = f_{PWA}(\theta(t))$. Then, we compensate for the delay by setting

$$\tau_d(t) = \tau_c(t+\sigma) = f_{PWA}(\hat{\theta}(t+\sigma)), \tag{10.17}$$

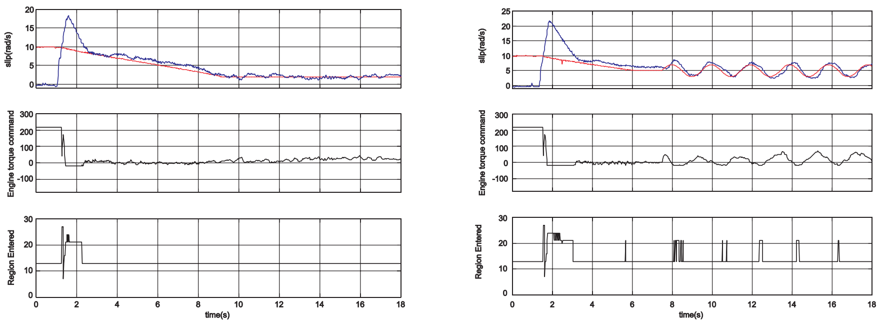
where $\hat{\theta}(t+\sigma)$ is the σ-step ahead predictor of $\theta(t)$. Since at time t, the inputs $\tau_d(t-i),\ i = 1, \ldots, \sigma$ and therefore $\tau_c(t-i+\sigma),\ i = 1, \ldots, \sigma$ are available, $\hat{\theta}(t+\sigma)$ can be computed from $\omega_e(t)$, $v_v(t)$ by iterating the PWA model (10.6), by assuming $\mu(t+\sigma) = \mu(t)$ and $\Delta\omega_d(t+\sigma) = \Delta\omega_d(t)$.

10.5 Motivation for Hybrid Control

There are several reasons that led us to solve the traction control problem by using a hybrid approach. First of all the nonlinear frictional torque τ_t in (10.4) has a clear piecewise-linear behaviour [143]: The traction torque increases almost linearly for low values of the slip, until it reaches a certain peak after which it start decreasing. For various road conditions the curves have different peaks and slopes. By including such a piecewise linearity in the model we obtained a single control law that is able to achieve the control task for a wide range of road conditions. Moreover, the design flow has the following advantages: (i) From the definition of the control objective to its solution, the problem is tackled in a systematic way by using the HYSDEL compiler and multiparametric programming algorithms; (ii) Constraints are embedded in the control problem in a natural and effective way; (iii) The resulting control law is piecewise affine and requires much less supervision by logical constructs than controllers developed with traditional techniques (e.g. PID control); (iv) It is easy to extend the design to handle more accurate models and include additional constraints without changing the design flow. For example, one can use a better piecewise-linear approximation of the traction torque, a more detailed model of the dynamics and include logic constructs in the model such as an hysteresis for the controller activation as a function of the slip.

10.6 Experimental Setup and Results

The hybrid traction controller was tested in a small (1390 kg) front-wheel-drive passenger vehicle with manual transmission. The explicit controller was run with a 20 ms timebase in a 266 MHz Pentium II-based laptop. Vehicle wheel speeds were measured directly by the laptop computer, and the calculated engine torque command was passed to the powertrain control module through a serial bus. Torque control in the engine was accomplished through spark retard, cylinder air/fuel ratio adjustment, and cylinder fuel shutoff where needed. The overall system latency from issuance of the torque command to production of the actual torque by the engine was 0.25 seconds. The vehicle was tested on a polished ice surface (indoor ice arena, $\mu \simeq 0.2$) with a variety of ramp, step, and sinusoidal tracking reference signals. Control intervention was initiated when the average driven wheel speed exceeded the reference wheel speed for the first time. Several hybrid models (up to 7-th order) and several PPWA controllers of different complexity (up to 1500 regions) have been tested. Figures 10.5 illustrate two test results with a PPWA optimal controller based on the simplest model described in Section 10.3 and consisting of 40 regions.



(a) Ramp and step slip reference (b) Sinusoidal slip reference

Fig. 10.5. Controller 1

10.7 HYSDEL Hybrid Model

Below we report the description list in HYSDEL of the traction control model described in Section 10.2.1.

```
SYSTEM FordCar  {

INTERFACE {
/* Description of variables and constants */
    STATE {
            REAL taotold ;
            REAL we      ;
            REAL vv      ;
    }
    INPUT { REAL deltataot; REAL mu;
    }
    PARAMETER {
            /* Region of the PWA linearization */
            /* ar * mu + br * deltaw <= cr */
            REAL ar = -5.3781;
            REAL br = 53.127/250;
            REAL cr = -0.61532;
            /* Other parameters */
            REAL deltawmax = 400;
            REAL deltawmin = -400;
            REAL zwemax = 1000;
            REAL zwemin = -100;
            REAL zvvmax = 80;
            REAL zvvmin = -100;
            REAL gr = 13.89;
            REAL rt = 0.298;
            REAL e = 1e-6;
            /* Dynamic behaviour of the model (Matlab generated) */
            REAL a11a = 0.98316      ;
            REAL a12a = 0.78486      ;
            REAL a21a = 0.00023134   ;
            REAL a22a = 0.989220     ;
            REAL b11a = 0.048368     ;
            REAL b12a = -0.35415     ;
            REAL b21a = 0.089695     ;
            REAL b22a = 0.0048655    ;
```

```
            REAL f1a = 0.048792      ;
            REAL f2a = -1.5695e-007  ;
            REAL a11b = 1.0005       ;
            REAL a12b = -0.021835    ;
            REAL a21b = -6.4359e-006 ;
            REAL a22b = 1.00030      ;
            REAL b11b = 0.048792     ;
            REAL b12b = -6.5287      ;
            REAL b21b = -1.5695e-007 ;
            REAL b22b = 0.089695     ;
            REAL f1b = 0.81687       ;
            REAL f2b = -0.011223     ;
            }
    }
    IMPLEMENTATION {
        AUX {
                REAL zwe, zvv;
                BOOL region;
        }
        AD {
                /* PWA Domain */
                region = ar * ((we / gr) - (vv / rt)) + br * mu - cr <= 0
                /*  region=  ((we / gr) - (vv / rt))-1.6 <= 0 */
                [deltawmax,deltawmin,e];
        }
        DA {
                zwe = { IF region THEN a11a * we + a12a * vv + b12a* mu+f1a   [zwemax, zwemin, e]
                        ELSE a11b * we + a12b * vv + b12b* mu+f1b    [zwemax, zwemin, e]
                };
                zvv = { IF region THEN a21a * we + a22a * vv + b22a* mu +f2a [zvvmax, zvvmin, e]
                        ELSE a21b * we + a22b * vv + b22b* mu+f2b           [zvvmax, zvvmin, e]
        };
                }
    CONTINUOUS {
                    taotold     = deltataot;
                    we          = zwe+b11a* taotold+b11a* deltataot ;
                    vv          = zvv+b21a* taotold+b21a* deltataot;
               }
    MUST { deltataot <= 2000;
           taotold <=176;
           -taotold <=20;
           (we/gr)-(vv/rt)>=0;
    }
}
}
```

References

1. J. Acevedo and E.N. Pistikopoulos. A multiparametric programming approach for linear process engineering problems under uncertainty. *Ind. Eng. Chem. Res.*, 36:717–728, 1997.
2. I. Adler and R.D.C. Monteiro. A geometric view of parametric linear programming. *Algorithmica*, 8(2):161–176, 1992.
3. J.C. Allwright and G.C. Papavasiliou. On linear programming and robust model-predictive control using impulse-responses. *Systems & Control Letters*, 18:159–164, 1992.
4. S. Alther and A. Kaufman. Regulation and trajectory tracking of a ball on a plate by using MPC. Technical report, Automatic Control Laboratory, ETH Zurich, 2001. Semester Thesis.
5. R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In A.P. Ravn R.L. Grossman, A. Nerode and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer Verlag, 1993.
6. P.J. Antsaklis. A brief introduction to the theory and applications of hybrid systems. *Proc. IEEE, Special Issue on Hybrid Systems: Theory and Applications*, 88(7):879–886, July 2000.
7. A. Asarin, O. Maler, and A. Pnueli. On the analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138:35–65, 1995.
8. R. Balakrishna and A. Ghosal. Modeling of slip for wheeled mobile robots. *IEEE Trans. Robotics and automation*, 11(1):349–370, February 1995.
9. A. Balluchi, L. Benvenuti, M. Di Benedetto, C. Pinello, and A. Sangiovanni-Vincentelli. Automotive engine control and hybrid systems: Challenges and opportunities. *Proc. IEEE*, 88(7):888–912, 2000.
10. M. Baotic. An efficient algorithm for multi-parametric quadratic programming. Technical Report AUT02-04, Automatic Control Laboratory, ETH Zurich, Switzerland, February 2002.
11. M. Bauer and M. Tomizuka. Fuzzy logic traction controllers and their effect on longitudinal vehicle platoon. *Vehicle System Dynamics*, 25(4):277–303, April 1996.

12. M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming — Theory and Algorithms*. John Wiley & Sons, Inc., New York, second edition, 1993.
13. A. Bellini, A. Bemporad, E. Franchi, N. Manaresi, R. Rovatti, and G. Torrini. Analog fuzzy implementation of a vehicle traction sliding-mode control. In *Proc. ISATA 29th International Symposium on Automotive Technology and Automation. Automotive Autom, Croydon, UK*, pages 275–282, 1996.
14. A. Bemporad. Reducing conservativeness in predictive control of constrained systems with disturbances. In *Proc. 37th IEEE Conf. on Decision and Control*, pages 1384–1391, Tampa, FL, 1998.
15. A. Bemporad, F. Borrelli, L. Glielmo, and F. Vasca. Optimal piecewise-linear control of dry clutch engagement. In *IFAC Workshop Advances in Automotive Control*, Karlsruhe, Germany, March 2001.
16. A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear optimal controllers for hybrid systems. In *Proc. American Contr. Conf.*, pages 1190–1194, Chicago, IL, June 2000.
17. A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear robust model predictive control. In *Proc. European Control Conf.*, Porto, Portugal, October 2001.
18. A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Trans. Automatic Control*, 47(12):1974–1985, December 2002.
19. A. Bemporad, L. Chisci, and E. Mosca. On the stabilizing property of SIORHC. *Automatica*, 30(12):2013–2015, 1994.
20. A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Trans. Automatic Control*, 45(10):1864–1876, 2000.
21. A. Bemporad, K. Fukuda, and F.D. Torrisi. Convexity recognition of the union of polyhedra. *Computational Geometry*, 18:141–154, 2001.
22. A. Bemporad, N. Giorgetti, I.V. Kolmanovsky, and D. Hrovat. Hybrid modeling and control of a direct injection stratified charge engine. In *Symposium on Advanced Automotive Technologies, ASME International Mechanical Engineering Congress and Exposition*, New Orleans, Louisiana, November 2002.
23. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.
24. A. Bemporad and M. Morari. Robust model predictive control: A survey. In A. Garulli, A. Tesi, and A. Vicino, editors, *Robustness in Identification and Control*, number 245 in Lecture Notes in Control and Information Sciences, pages 207–226. Springer-Verlag, 1999.
25. A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
26. A. Bemporad, M. Morari, and N.L. Ricker. MPC Simulink Block. Technical Report and software available upon request from the authors, ETH Zurich, Switzerland, 2000.
27. Claude Berge. *Topological Spaces*. Dover Publications, inc., Mineola, New York, 1997.
28. A.B. Berkelaar, K. Roos, and T. Terlaky. The optimal set and optimal partition approach to linear and quadratic programming. In T. Gal and H.J. Greenberg, editors, *Advances in Sensitivity Analysis and Parametric Programming*, volume 6 of *International Series in Operations Research & Management Science*, chapter 6. Kluwer Academic Publishers, 1997.

29. D.P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, Massachusetts, 1995.
30. G. Bitsoris. On the positive invariance of polyhedral sets for discrete-time systems. *Systems & Control Letters*, 11:243–248, 1988.
31. F. Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Trans. Automatic Control*, 39(2):428–433, February 1994.
32. F. Blanchini. Set invariance in control — a survey. *Automatica*, 35(11):1747–1768, November 1999.
33. V.D. Blondel, O. Bournez, P. Koiran, and J.N. Tsitsiklis. The stability of saturated linear systems is undecidable. *Journal of Computer Systems Science,* submitted, July 1999. `http://truth.mit.edu/~jnt/sat.ps`.
34. V.D. Blondel and J.N. Tsitsiklis. Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35:479–489, March 1999.
35. F. Borrelli, M. Baotic, A. Bemporad, and M. Morari. Efficient on-line computation of constrained optimal control. In *Proc. 40th IEEE Conf. on Decision and Control*, December 2001.
36. F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat. A hybrid approach to traction control. In A. Sangiovanni-Vincentelli and M.D. Di Benedetto, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer Verlag, 2001.
37. F. Borrelli, A. Bemporad, and M. Morari. A geometric algorithm for multi-parametric linear programming. *J. Opt. Theory and Applications*, 118(3), September 2003.
38. M. Branicky. Stability of switched and hybrid systems. In *Proc. Conf. Dec. Contr.*, pages 3498–3503, 1994.
39. M.S. Branicky. *Studies in hybrid systems: modeling, analysis, and control.* PhD thesis, LIDS-TH 2304, Massachusetts Institute of Technology, Cambridge, MA, 1995.
40. M.S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automatic Control*, 43(4):475–482, April 1998.
41. M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Trans. Automatic Control*, 43(1):31–45, 1998.
42. M.S. Branicky and S.K. Mitter. Algorithms for optimal hybrid control. In *Proc. 34th IEEE Conf. on Decision and Control*, New Orleans, USA, December 1995.
43. M.S. Branicky and G. Zhang. Solving hybrid control problems: Level sets and behavioral programming. In *Proc. American Contr. Conf.*, Chicago, Illinois USA, June 2000.
44. M. Buss, O. von Stryk, R. Bulirsch, and G. Schmidt. Towards hybrid optimal control. *AT - Automatisierungstechnik*, 48:448–459, 2000.
45. P.J. Campo and M. Morari. Robust model predictive control. In *Proc. American Contr. Conf.*, volume 2, pages 1021–1026, 1987.
46. P.J. Campo and M. Morari. Model predictive optimal averaging level control. *AIChE Journal*, 35(4):579–591, 1989.
47. T.M. Cavalier, P.M. Pardalos, and A.L. Soyster. Modeling and integer programming techniques applied to propositional calculus. *Computers Opns Res.*, 17(6):561–570, 1990.

48. T. S. Chang and D. E. Seborg. A linear programming approach for multivariable feedback control with inequality constraints. *Int. J. Control*, 37(3):583–597, 1983.
49. L. Chisci and G. Zappa. Fast algorithm for a constrained infinite horizon LQ problem. *Int. J. Control*, 72(11):1020–1026, 1999.
50. D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, 29(3):121–130, November 1996.
51. C.R. Cuttler and B.C. Ramaker. Dynamic matrix control—a computer control algorithm. In *Proc. American Contr. Conf.*, volume WP5-B, San Francisco, USA, 1980.
52. G. B. Dantzig, J. Folkman, and N. Shapiro. On the continuity of the minimum set of a continuous function. *Journal of Mathematical Analysis and Applications*, 17:519–548, 1967.
53. A. Dario and T. Kueng. Graphical user interface for the ball and plate system. Technical report, Automatic Control Laboratory, ETH Zurich, 2001. Semester Thesis.
54. B. De Schutter and B. De Moor. The extended linear complementarity problem and the modeling and analysis of hybrid systems. In P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, editors, *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 1999.
55. B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear systems. In *Proc. American Contr. Conf.*, pages 4046–4050, 2000.
56. J. A. De Doná. *Input Constrained Linear Control.* PhD thesis, Control Group, Department of Engineering, University of Cambridge, Cambridge, 2000.
57. V. Dua and E.N. Pistikopoulos. An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, to appear.
58. J. Ezzine and A.H. Haddad. On the controllability and observability of hybrid systems. *Int. J. Control*, 49:2045–2055, 1989.
59. G. Ferrari-Trecate, F.A. Cuzzola, and M. Morari. Analysis of piecewise affine systems with logic states. In C. J. Tomlin and M. R. Greenstreet, editors, *Proc. 5th International Workshop on Hybrid Systems: Computation and Control*, volume 2289 of *Lecture Notes in Computer Science*, pages 194–208. Springer-Verlag, 2002.
60. G. Ferrari-Trecate, E. Gallestey, P. Letizia, M. Spedicato, M. Morari, and M. Antoine. Modeling and control of co-generation power plants: A hybrid system approach. Technical Report AUT02-01, Automatic Control Laboratory, ETH Zurich, 2002.
61. G. Ferrari-Trecate, E. Gallestey, A. Stothert, P. Letizia, M. Spedicato, M. Morari, and M. Antoine. Modeling and control of co-generation power plants: A hybrid system approach. In C. J. Tomlin and M. R. Greenstreet, editors, *Proc. 5th International Workshop on Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer-Verlag, 2002.
62. G. Ferrari-Trecate, D. Mignone, and M. Morari. Moving horizon estimation for hybrid systems. In *Proc. American Contr. Conf.*, 2000.
63. G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Proc. 4th International Workshop on*

Hybrid Systems: Computation and Control, volume 2034 of *Lecture Notes in Computer Science*, pages 218–231. Springer-Verlag, 2001.
64. A V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, London, U.K., 1983.
65. C. Filippi. On the geometry of optimal partition sets in multiparametric linear programming. Technical Report 12, Department of Pure and Applied Mathematics, University of Padova, Italy, June 1997.
66. M. Fodor, J. Yester, and D. Hrovat. Active control of vehicle dynamics. In *Proceedings of the 17th Digital Avionics Systems Conference*, Seattle, Washington, November 1999.
67. K. Fukuda. *cdd/cdd+ Reference Manual*. Institute for operations Research ETH-Zentrum, ETH-Zentrum, CH-8092 Zurich, Switzerland, 0.61 (cdd) 0.75 (cdd+) edition, December 1997.
68. T. Gal. *Postoptimal Analyses, Parametric Programming, and Related Topics*. de Gruyter, Berlin, 2nd edition, 1995.
69. T. Gal and H.J. Greenberg (Eds.). *Advances in Sensitivity Analysis and Parametric Programming*, volume 6 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, 1997.
70. T. Gal and J. Nedoma. Multiparametric linear programming. *Management Science*, 18:406–442, 1972.
71. C.E. García and A.M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chem. Eng. Communications*, 46:73–87, 1986.
72. S.I. Gass and T.L. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2:39–45, 1955.
73. H. Genceli and M. Nikolaou. Robust stability analysis of constrained ℓ_1-norm model predictive control. *AIChE J.*, 39(12):1954–1965, 1993.
74. A. Gentilini, C. Schaniel, M. Morari, C. Bieniok, R. Wymann, and T. Schnider. A new paradigm for the closed-loop intraoperative administration of analgesics in humans. Technical Report AUT01-01, Automatic Control Lab, ETH Zurich, 2001.
75. E.G. Gilbert and K. Tin Tan. Linear systems with state and control constraints: the theory and applications of maximal output admissible sets. *IEEE Trans. Automatic Control*, 36(9):1008–1020, 1991.
76. K. Gokbayrak and C.G. Cassandras. A hierarchical decomposition method for optimal control of hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 1816–1821, Phoenix, AZ, December 1999.
77. J.E. Goodman and Joseph O'Rourke (Eds.). *Handbook of Discrete and Computational Geometry*. Discrete Mathematics and Its Applications. CRC Press, New York, 1997.
78. D. Goshen-Meskin and I. Y. Bar-Itzhack. Observability analysis of piece-wise constant systems—part I: Theory. *IEEE Trans. Aerosp. Electr. Syst.*, 28(4):1056–1067, October 1992.
79. P. Grieder, F. Borrelli, F. Torrisi, and M. Morari. Constrained infinite horizon linear quadratic regulator. Technical Report AUT02-66, Automatic Control Laboratory, ETH Zurich, Switzerland, July 2002. `http://control.ee.ethz.ch`.
80. P.O. Gutman and M. Cwikel. Admissible sets and feedback control for discrete-time linear dynamical systems with bounded control and states. *IEEE Trans. Automatic Control*, AC-31(4):373–376, 1986.

81. A Hassibi and S. Boyd. Quadratic stabilization and control of piecewise-linear systems. In *Proc. American Contr. Conf.*, Philadelphia, Pennsylvania USA, June 1998.
82. S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 3972–3976, Phoenix, AZ, December 1999.
83. W.P.M.H. Heemels. *Linear complementarity systems: a study in hybrid dynamics*. PhD thesis, Dept. of Electrical Engineering, Eindhoven University of Technology, The Netherlands, 1999.
84. W.P.M.H. Heemels, J.M. Schumacher, and S. Weiland. Linear complementarity systems. *SIAM Journal on Applied Mathematics*, 60(4):1234–1269, 2000.
85. W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.
86. T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata ? In *Proc. 27th ACM Symposium on the Theory of Computing*, pages 373–381, 1995.
87. O. Hermann. Regelung eines ball and plate systems. Technical report, Automatic Control Laboratory, ETH Zurich, 1996. Diploma Thesis.
88. W. M. Hogan. Point-to-set maps in mathematical programming. *SIAM Review*, 15(3):591–603, July 1973.
89. D. Hrovat. Survey of advanced suspension developments and related optimal control applications. *Automatica*, 33:1781–1817, 1997.
90. D. Hrovat, J. Asgari, and M. Fodor. Automotive mechatronic systems. In Cornelius T. Leondes, editor, *Chapter 1 of "Mechatronic Systems Techniques and Applications Volume 2 - Transportation and Vehicular Systems"*. Gordon and Breach Science Publishers, 2000.
91. A. Jadbabaie, J. Yu, and J. Hauser. Stabilizing receding horizon control of nonlinear systems: a control lyapunov function approach. In *Proc. American Contr. Conf.*, June 1999.
92. M. Johannson and A. Rantzer. Computation of piece-wise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Automatic Control*, 43(4):555–559, 1998.
93. T.A. Johansen, J. Kalkkuhl, J. Lüdemann, and I. Petersen. Hybrid control strategies in ABS. In *Proc. American Contr. Conf.*, Arlington, Virginia, June 2001.
94. K H Johansson, M Egerstedt, J Lygeros, and S Sastry. On the regularization of Zeno hybrid automata. *System & Control Letters*, 38:141–150, 1999.
95. P. Kachroo and M. Tomizuka. An adaptive sliding mode vehicle traction controller design. In *Proc. IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, volume 1, pages 777–782, 1995.
96. S.S. Keerthi and E.G. Gilbert. Optimal infinite-horizon feedback control laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. *J. Opt. Theory and Applications*, 57:265–293, 1988.
97. E. C. Kerrigan. *Robust Constraints Satisfaction: Invariant Sets and Predictive Control*. PhD thesis, Department of Engineering, University of Cambridge, Cambridge, England, 2000.
98. E.C. Kerrigan and J.M. Maciejowski. Towards the explicit solution of a class of constrained min-max problems with non-quadratic costs. Technical report, Dept. of Engineering, University of Cambridge, June 2001.

99. Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: a class of decidable hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 179–208. Springer Verlag, 1993.
100. H. Kiendl, J. Adamy, and P. Stelzner. Vector norms as Lyapunov functions for linear systems. *IEEE Trans. Automatic Control*, 37(6):839–842, 1992.
101. M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.
102. B. Kouvaritakis, J.A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Trans. Automatic Control*, 45(8):1545–1549, 2000.
103. G. Labinaz, M.M. Bayoumi, and K. Rudie. A survey of modeling and control of hybrid systems. *Annual Reviews of Control*, 21:79–92, 1997.
104. J. H. Lee and Z. Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997.
105. J. Lygeros, D.N. Godbole, and S. Sastry. A game theoretic approach to hybrid system design. In R. Alur and T. Henzinger, editors, *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, 1996.
106. J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.
107. J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.
108. M. Mannelli. Gestione integrata del cambio robotizzato per la riduzione di consumi ed emissioni mediante tecniche di controllo ibrido. Technical report, Universita' degli Studi di Firenze, Facolta' di Ingegneria, Firenze, April 2002. Diploma Thesis.
109. G. F. Mauer. A fuzzy logic controller for an ABS braking system. *IEEE Transaction on Fuzzy Systems*, November 1995.
110. D.Q. Mayne. Constrained optimal control. *European Control Conference, Plenary Lecture*, September 2001.
111. D.Q. Mayne. Control of constrained dynamic systems. *European Jornal of Control*, 7:87–99, 2001.
112. D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.
113. S. Mehrotra and R.D.C. Monteiro. Parametric and range analysis for interior point methods. Technical report, Dept. of Systems and Industrial Engineering, University of Arizona, Tucson, USA, 1992.
114. W. Michael and F. Philipp. Regulation and trajectory tracking of a ball on a plate by using mpc. Technical report, Automatic Control Laboratory, ETH Zurich, 2000. Semester Thesis.
115. D. Mignone. *Control and Estimation of Hybrid systems via Mathematical Optimization*. PhD thesis, Automatic Control Labotatory - ETH, Zurich, 2002.
116. D. Mignone, G. Ferrari-Trecate, and M. Morari. Stability and stabilization of piecewise affine and hybrid systems: An LMI approach. In *Proc. 39th IEEE Conf. on Decision and Control*, December 2000.
117. M. Morari and J.H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4–5):667–682, 1999.
118. U. Muntwyler and S. Moscibroda. Regulation and trajectory tracking of a ball on a plate. Technical report, Automatic Control Laboratory, ETH Zurich, 2000. Semester Thesis.

119. K. G. Murty. Computational complexity of parametric linear programming. *Mathematical Programming*, 19:213–219, 1980.
120. G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.
121. B. Piccoli. Necessary conditions for hybrid optimization. In *Proc. 38th IEEE Conf. on Decision and Control*, Phoenix, Arizona USA, December 1999.
122. A. Polański. On infinity norms as Lyapunov functions for linear systems. *IEEE Trans. Automatic Control*, 40(7):1270–1273, 1995.
123. A.I. Propoi. Use of linear programming methods for synthesizing sampled-data automatic systems. *Automation and Remote Control*, 24(7):837–844, 1963.
124. S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. In *Chemical Process Control - V*, volume 93, no. 316, pages 232–256. AIChe Symposium Series - American Institute of Chemical Engineers, 1997.
125. R. Raman and I.E. Grossmann. Relation between MILP modeling and logical inference for chemical process synthesis. *Computers & Chemical Engineering*, 15(2):73–84, 1991.
126. C.V. Rao and J.B. Rawlings. Linear programming and model predictive control. *J. Process Control*, 10:283–289, 2000.
127. J.B. Rawlings and K.R. Muske. The stability of constrained receding-horizon control. *IEEE Trans. Automatic Control*, 38:1512–1516, 1993.
128. J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control-application to industrial processes. *Automatica*, 14:413–428, 1978.
129. P. Riedinger, F.Kratz, C. Iung, and C.Zanne. Linear quadratic optimization for hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, Phoenix, Arizona USA, December 1999.
130. R.L.Grossmann, A. Nerode, A.P.Ravn, and H. Rischel (Eds.). *Hybrid Systems*. Springer Verlag, New York, 1993. no. 736 in LCNS.
131. S. M. Robinson and R.H. Day. A sufficient condition for continuity of optimal sets in mathematical programming. *Journal of Mathematical Analysis and Applications*, 45:506–511, 1974.
132. M. Schechter. Polyhedral functions and multiparametric linear programming. *J. Opt. Theory and Applications*, 53(2):269–280, May 1987.
133. P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. Automatic Control*, 43(8):1136–1142, 1998.
134. P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *IEEE Trans. Automatic Control*, 43(8):1163–1169, 1998.
135. M.M. Seron, J.A. DeDoná, and G.C. Goodwin. Global analytical model predictive control with input constraints. In *Proc. 39th IEEE Conf. on Decision and Control*, pages 154–159, 2000.
136. E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automatic Control*, 26(2):346–358, April 1981.
137. E.D. Sontag. Real addition and the polynomial hierarchy. *Inform. Proc. Letters*, 20:115–120, 1985.
138. E.D. Sontag. Controllability is harder to decide than accessibility. *SIAM J. Control and Opt.*, 26:1106–1118, 1988.
139. E.D. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete-time. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III - Verification and Control*, number 1066 in Lecture Notes in Computer Science, pages 436–448. Springer-Verlag, 1996.

140. H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *Proc. 38th IEEE Conf. on Decision and Control*, Phoenix, Arizona USA, December 1999.
141. M. Sznaier and M.J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Proc. 26th IEEE Conf. on Decision and Control*, volume 1, pages 761–762, 1987.
142. H. S. Tan. *Adaptive and Robust Controls with Application to Vehicle Traction Control*. PhD thesis, Univ. of California, Berkeley, 1988.
143. H. S. Tan and M. Tomizuka. Discrete time controller design for robust vehicle traction. *IEEE Control System Magazine*, 10(3):107–113, April 1990.
144. H.S. Tan and M. Tomizuka. An adaptive sliding mode vehicle traction controller design. In *Proc. American Contr. Conf.*, volume 2, pages 1856–1861, 1990.
145. K.T. Tan. *Maximal output admisible sets and the nonlinear control of linear discrete-time systems with state and control constraints*. PhD thesis, University of Michigan, 1991.
146. C.J. Tomlin, J. Lygeros, and S.S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceeding of IEEE*, 88, July 2000.
147. P. Tøndel, T.A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In *Proc. 40th IEEE Conf. on Decision and Control*, December 2001.
148. P. Tøndel, T.A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In *Proc. 40th IEEE Conf. on Decision and Control*, Orlando, Florida, 2001.
149. P. Tøndel, T.A. Johansen, and A. Bemporad. Computation and approximation of piecewise affine control via binary search trees. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 3144–3149, 2002.
150. F.D. Torrisi and A. Bemporad. HYSDEL — a tool for generating computational hybrid models. *IEEE Trans. Control Systems Technology*, 2003. to appear.
151. M.L. Tyler and M. Morari. Propositional logic in control and monitoring problems. *Automatica*, 35(4):565–582, 1999.
152. V.I. Utkin. Variable structure systems with sliding modes. *IEEE Trans. Automatic Control*, 22(2):212–222, April 1977.
153. A.J. van der Schaft and J.M. Schumacher. Complementarity modelling of hybrid systems. *IEEE Trans. Automatic Control*, 43:483–490, 1998.
154. M. Vidyasagar. *Nonlinear Systems Analysis - Second Edition*. Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1993.
155. L. Wenger and F. Borrelli. The application of constrained optimal control to active automotive suspensions. In *Proc. 41th IEEE Conf. on Decision and Control*, 2002.
156. M. A. Wicks and P. Peleties. Construction of piecewise lyapunov functions for stabilizing switched systems. In *Proc. Conf. Dec. Contr.*, pages 3492–3497, 1994.
157. H.P. Williams. Logical problems and integer programming. *Bulletin of the Institute of Mathematics and Its Applications*, 13:18–20, 1977.
158. H.P. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, Third Edition, 1993.
159. H.S. Witsenhausen. A min-max control problem for sampled linear systems. *IEEE Trans. Automatic Control*, 13(1):5–21, 1968.
160. L.A. Zadeh and L.H. Whalen. On optimal control and linear programming. *IRE Trans. Automatic Control*, 7:45–46, 1962.

161. A. Zheng and M. Morari. Stability of model predictive control with mixed constraints. *IEEE Trans. Automatic Control*, 40:1818–1823, 1995.