Available online at www.sciencedirect.com
SCIENCE DIRECT®
automatica
Automatica 40 (2004) 701 – 708
www.elsevier.com/locate/automatica

Brief paper

Computation of the constrained infinite time linear quadratic regulator☆

Pascal Grieder*, Francesco Borrelli, Fabio Torrisi, Manfred Morari

Institut für Automatik, ETH, Swiss Federal Institute of Technology, Physikstrasse 3, CH-8092 Zürich, Switzerland

Received 11 September 2002; received in revised form 18 May 2003; accepted 27 November 2003

Abstract

This paper presents an efficient algorithm for computing the solution to the constrained infinite-time, linear quadratic regulator (CLQR) problem for discrete time systems. The algorithm combines multi-parametric quadratic programming with reachability analysis to obtain the optimal piecewise affine (PWA) feedback law. The algorithm reduces the time necessary to compute the PWA solution for the CLQR when compared to other approaches. It also determines the minimal finite horizon $\bar{N}_{\mathcal{S}}$, such that the constrained finite horizon LQR problem equals the CLQR problem for a compact set of states $\mathcal{S}$. The on-line computational effort for the implementation of the CLQR can be significantly reduced as well, either by evaluating the PWA solution or by solving the finite dimensional quadratic program associated with the CLQR for a horizon of $N = \bar{N}_{\mathcal{S}}$.

Keywords: Constrained infinite horizon control; Linear quadratic regulator; Model predictive control; Invariant Set

1. Introduction

Recently, extensive work has been carried out on constrained optimal control of linear, time-invariant systems. One main focus of research has been *finite horizon constrained optimal control* (FHCOC) (Richalet, Rault, Testud, & Papon, 1978; Cuttler & Ramaker, 1980; Mayne, Rawlings, Rao, & Scokaert, 2000). The solution to quadratic FHCOC can be found by solving a quadratic program (QP) of finite dimension or, as proposed by Bemporad, Morari, Dua, and Pistikopoulos (2002), by evaluating a PWA state feedback control law which was computed off-line by solving a multi-parametric quadratic program. It is current practice to approximate the *infinite time constrained LQR* (CLQR) problem by receding horizon control. For receding horizon control, a FHCOC problem is solved at each time step, and then only the initial value of the optimal input sequence is applied to the plant. The main problem of receding horizon control is that it does not, in general, guarantee stability. In order to make receding horizon control stable, conditions have to be added to the original problem which may result in degraded performance (Rawlings & Muske, 1993).

It is clear that a return to the infinite-horizon formulation is required to produce stabilizing control laws (Gevers, Bitmead, & Wertz, 1990) which guarantee global optimality. However, rather than address the full CLQR problem, all *model predictive control* (MPC) variants, with few exceptions (Scokaert & Rawlings, 1998; Chmielewski & Manousiouthakis, 1996), rely on approximations. Sznaier and Damborg (1987) showed that a finite horizon optimization over a horizon $\bar{N}$ can provide the solution to the infinite-horizon constrained optimal control problem. However, there is no technique to compute $\bar{N}$ for compact sets of points apart from the conservative upper bound by Chmielewski and Manousiouthakis (1996).

The contribution of this paper is a novel approach to compute the PWA state feedback solution to the CLQR problem. The presented algorithm combines multi-parametric quadratic programming (Bemporad et al., 2002) with reachability analysis to obtain the infinite-time optimal piecewise affine (PWA) feedback law. The algorithm significantly reduces the time necessary to compute the PWA solution for the CLQR when compared to other approaches (Baotić, 2002; Bemporad et al., 2002). Furthermore, the algorithm does not rely on estimates of $\bar{N}$ but instead computes $\bar{N}$ for compact sets. Thus, the on-line computation of the CLQR

☆ This paper was not presented of any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Marco Campi under the direction of Editor T. Başar

* Corresponding author. Tel.: +41-1-6327313; fax: +41-1-6321211.
E-mail addresses: grieder@control.ee.ethz.ch (P. Grieder), borrelli@control.ee.ethz.ch (F. Borrelli), torrisi@control.ee.ethz.ch (F. Torrisi), morari@control.ee.ethz.ch (M. Morari).

0005-1098/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2003.11.014

can be significantly reduced by either evaluating the PWA solution or by solving the FHCOC problem for a horizon of $N = \bar{N}$.

2. Problem statement and properties

In this paper, we will consider optimal control problems for discrete-time linear time-invariant systems

$$x(t+1) = Ax(t) + Bu(t) \tag{1}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$ and (A, B) controllable. Let $x(t)$ denote the measured state at time t and $x_{t+k|t}$ denote the predicted state at time $t+k$ given the state at time t. For brevity we will denote $x_{k|0}$ as x_k. In this section, we will give a brief overview of constrained optimal control problems discussed in the literature.

2.1. Finite-time constrained LQR

Assume that the states and the inputs of system (1) are subject to the following constraints:

$$x \in \mathbb{X} \subseteq \mathbb{R}^n, \quad u \in \mathbb{U} \subseteq \mathbb{R}^m, \tag{2}$$

where $\mathbb{X}$ and $\mathbb{U}$ are closed polyhedral sets containing the origin in their interiors, and consider the finite-time constrained optimal control problem

$$J_N^*(x(0)) = \min_{u_0,\ldots,u_{N-1}} \left\{ \sum_{k=0}^{N-1} (u_k' \mathscr{R} u_k + x_k' \mathscr{Q} x_k) + x_N' \mathscr{Q}_f x_N \right\}$$

$$\text{s. t.} \quad x_k \in \mathbb{X}, \quad k \in [1, \ldots, N], \tag{3a}$$

$$u_k \in \mathbb{U}, \quad k \in [0, \ldots, N-1], \tag{3b}$$

$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = x(0). \tag{3c}$$

$$\mathscr{Q} \succeq 0, \quad \mathscr{Q}_f \succeq 0, \quad \mathscr{R} \succ 0. \tag{3d}$$

Henceforth, we will assume the terminal weight matrix $\mathscr{Q}_f$ to be equal to the solution P of the algebraic Riccati equation (ARE). The solution to problem (3) has been studied by Bemporad et al. (2002). We will briefly summarize the main results. By substituting $x_k = A^k x(0) + \sum_{j=0}^{k-1} A^k B u_{k-1-j}$, problem (3) can be reformulated as

$$J_N^*(x(0)) = x(0)' \mathscr{Y} x(0) + \min_{U_N} \{U_N' \mathscr{H} U_N + x(0)' \mathscr{F} U_N\}$$

$$\text{s. t.} \quad GU_N \leqslant W + Ex(0), \tag{4}$$

where the column vector $U_N \triangleq [u_0', \ldots, u_{N-1}']' \in \mathbb{R}^s$ is the optimization vector, $s \triangleq mN$ and $\mathscr{H}$, $\mathscr{F}$, $\mathscr{Y}$, G, W, E are easily obtained from $\mathscr{Q}$, $\mathscr{R}$, $\mathscr{Q}_f$, (1) and (2) (see Bemporad et al. (2002) for details). We denote with $\mathscr{X}_f^N \subseteq \mathbb{R}^n$ the set of initial states $x(0)$ for which the optimal control problem (4) is feasible and the optimal cost $J_N^*(x(0))$ is finite, i.e.,

$$\mathscr{X}_f^N = \{x(0) \in \mathbb{R}^n \mid \exists U_N \in \mathbb{R}^s, GU_N \leqslant W + Ex(0)$$
$$\text{and } J_N^*(x(0)) < \infty\}.$$

Note that the condition $J_N^*(x(0)) < \infty$ is added for the case where $N = \infty$ (i.e., the set $\mathscr{X}_f^\infty$), in order to only consider points which converge to the origin. We denote with $U_N^*(x(0))$ the optimizer of (4).

Definition 1. The set of active constraints $\mathscr{A}^N(x)$ at point x of problem (4) is defined as

$$\mathscr{A}^N(x) = \{i \in I \mid G_i U_N^*(x) - W_i - E_i x = 0\},$$
$$I = \{1, 2, \ldots, m_G\},$$

where G_i, W_i and E_i denote the ith row of the matrices G, W and E, respectively, and $G \in \mathbb{R}^{m_G \times s}$. N is the horizon length used to solve (4).

Because problem (4) depends on $x(0)$, the implementation of the finite time constrained LQR can be performed either by solving the quadratic program (4) for a given $x(0)$, or as shown by Bemporad et al. (2002), by solving problem (4) for all $x(0)$ within a polyhedral set of values, i.e., by considering (4) as a *multi-parametric Quadratic Program* (mp-QP).

Theorem 1 (Bemporad et al., 2002; Borrelli, 2003). *Consider the finite time constrained LQR (3). Then the set of feasible parameters $\mathscr{X}_f^N$ is convex, the optimizer $U_N^* : \mathscr{X}_f^N \to \mathbb{R}^s$ is continuous and piecewise affine (PWA), i.e.,*

$$U_N^*(x(0)) = F_r x(0) + G_r \quad \textit{if } x(0) \in \mathscr{P}_r, \tag{5a}$$

$$\mathscr{P}_r = \{x \in \mathbb{R}^n \mid H_r x \leqslant K_r\}, \quad r = 0, \ldots, R \tag{5b}$$

and the value function $J^* : \mathscr{X}_f^N \to \mathbb{R}$ is continuous, convex and piecewise quadratic.

The solution of the mp-QP (4) (i.e., $\{\mathscr{P}_r\}_{r=0}^R$) is a polyhedral partition of $\mathscr{X}_f^N$ (Bemporad et al., 2002). Henceforth, we will denote the polyhedron $\mathscr{P}_r$ as region r.

2.2. Infinite-time constrained LQR (CLQR)

If in (3) we set $N = +\infty$, we obtain the infinite-time constrained LQR (CLQR) problem

$$J_\infty^*(x(0)) = \min_U \left\{ \sum_{k=0}^{+\infty} u_k' \mathscr{R} u_k + x_k' \mathscr{Q} x_k \right\}, \tag{6a}$$

$$\text{s.t.} \quad x_k \in \mathbb{X}, \quad k \in [1, 2, \ldots], \tag{6b}$$

$$u_k \in \mathbb{U}, \quad k \in [0, 1, \ldots], \tag{6c}$$

$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = x(0), \tag{6d}$$

$$\mathscr{Q} \succeq 0, \quad \mathscr{R} \succ 0, \tag{6e}$$

where the infinite dimensional vector $U_\infty \triangleq [u_0', u_1', \dots]$ is the optimization vector. We denote by U_∞^* the optimizer of (6). In order to show the equivalence of problems (3) and (6) we also define the following vector:

$$U^*(x(0), N) = [U_N^*(x(0)), \mathscr{K} x_N, \mathscr{K} x_{N+1}, \mathscr{K} x_{N+2}, \dots],$$

where $\mathscr{K}$ is the unconstrained optimal feedback law obtained from the ARE. The following theorems establish the solutions properties of the CLQR:

Theorem 2 (Chmielewski & Manousiouthakis, 1996; Scokaert & Rawlings, 1998). *Given system* (1), *there exists a finite horizon* $\hat{N}(x(0))$ *such that* $U_\infty^*(x(0)) = U^*(x(0), \hat{N}(x(0)))$ *for any* $x(0) \in \mathscr{X}_f^\infty$. *The equality also holds for all horizons* $N \geqslant \hat{N}(x(0))$.

Definition 2. We define $\bar{N}(x(0))$ to be the minimal horizon satisfying Theorem 2, i.e., $U_\infty^*(x(0)) = U^*(x(0), \bar{N}(x(0)))$ and $U_\infty^*(x(0)) \neq U^*(x(0), \bar{N}(x(0)) - 1)$.

Lemma 1 (Chmielewski & Manousiouthakis, 1996). *Consider a compact set* $\mathscr{S} \subset \mathbb{R}^n$ *of initial conditions* $x(0)$. *If the feasible set* $\mathscr{S}_F = \mathscr{S} \cap \mathscr{X}_f^\infty$ *is compact, there exists a finite horizon* $\bar{N}_\mathscr{S}$ *defined as*

$$\bar{N}_\mathscr{S} \triangleq \max_{x(0) \in \mathscr{S}_F} \bar{N}(x(0))$$

such that $U_\infty^*(x(0)) = U^*(x(0), \bar{N}_\mathscr{S})$ *for any* $x(0) \in \mathscr{S}_F$. *The equality also holds for all horizons* $N \geqslant \bar{N}_\mathscr{S}$. *If* $\mathscr{S}_F$ *is not closed or unbounded*, $\bar{N}(x(0))$ *may be unbounded.*

Theorem 3. *Consider a compact set* $\mathscr{S} \subset \mathbb{R}^n$ *of initial conditions* $x(0)$. *If the feasible set* $\mathscr{S}_F = \mathscr{S} \cap \mathscr{X}_f^\infty$ *is closed, the horizon* $\bar{N}_\mathscr{S}$ *is finite and therefore the state feedback solution* $U_\infty^* = U^*(x(0), \bar{N}_\mathscr{S})$ *of problem* (6) *defined over* $\mathscr{S}_F$ *is PWA, in particular*

$$U_{\bar{N}_\mathscr{S}}^*(x(0)) = F_r x(0) + G_r \text{ if } x(0) \in \mathscr{P}_r, \tag{7a}$$

$$\mathscr{P}_r = \{x \in \mathbb{R}^n | H_r x \leqslant K_r\},\ r = 0, \dots, R. \tag{7b}$$

Proof. Follows directly from Lemma 1 and Theorem 1 for $N = \bar{N}_\mathscr{S}$. Consequence of the results in Bemporad et al. (2002). □

Remark 1. If a PWA solution according to Theorem 1 is equal to the infinite-time PWA solution according to Theorem 3, the open-loop state trajectory is identical to the trajectory which is obtained if RHC is applied (Chisci & Zappa, 1999).

In view of the results of the previous section and Theorem 2, the implementation of CLQR can be performed either by solving (4) for a given $x(0)$ with $N = \bar{N}(x(0))$, or in a given compact set $\mathscr{S}$ of the initial conditions by solving the mp-QP (4) for $N \geqslant \bar{N}_\mathscr{S}$.

Various methods have been proposed in the literature for the computation of $\bar{N}(x(0))$ (Scokaert & Rawlings, 1998) and the estimation of $\bar{N}_\mathscr{S}$ (Chmielewski & Manousiouthakis, 1996). Note that $\bar{N}_\mathscr{S}$ is required for the computation of the infinite-time PWA solution presented in Theorem 1, using the techniques in Bemporad et al. (2002).

Chmielewski and Manousiouthakis (1996) presented an approach that provides a conservative estimate N_{est} of the finite horizon $\bar{N}_\mathscr{S}$ for a compact set $\mathscr{S}(N_{\text{est}} \geqslant \bar{N}_\mathscr{S})$. They solve a single, finite dimensional, convex program of known size to obtain N_{est}. Their estimate can be used either to compute the PWA solution of (6) for an arbitrary set $\mathscr{S}$ or, alternatively, a quadratic program with horizon N_{est} for any initial state $x(0) \in \mathscr{S}$. Chisci and Zappa (1999) presented a fast algorithm which is capable of speeding up the computation time for the CLQR problem by a factor of N_{est}/n, where n is the number of states. The procedure involves the solution of a QP as in (4) with horizon N_{est}. For a given initial state $x(0)$, Scokaert and Rawlings (1998) presented an algorithm that attempts to identify $\bar{N}(x(0))$ iteratively. The key definitions and theorems are reformulated here for completeness.

Definition 3. $\mathscr{X}_I \subseteq \mathbb{R}^n$ denotes the maximum invariant set of states for which the unconstrained LQR control law $\mathscr{K}$ obtained from the ARE satisfies the constraints in (2) for all time, i.e.,

$$\mathscr{X}_I = \{x(0) \in \mathbb{R}^n | x(k) \in \mathbb{X}, \mathscr{K} x(k) \in \mathbb{U}, \forall k \geqslant 0\}.$$

$\mathscr{X}_I$ is a positive invariant set containing an open neighborhood of the origin (Sznaier & Damborg, 1987), provided the origin is contained in the interior of the set described by (2).

Theorem 4 (Chmielewski & Manousiouthakis, 1996; Scokaert & Rawlings, 1998). *For any given initial state* $x(0)$, *the solution to* (4) *is equal to the infinite time solution* (6), *i.e.*, $J_N^*(x(0)) = J_\infty^*(x(0))$ *and* $U_\infty^*(x(0)) = U^*(x(0), N)$, *if the terminal state* x_N *of* (4) *lies in the unconstrained positive invariant set* $\mathscr{X}_I (x_N \in \mathscr{X}_I)$.

The method in Scokaert and Rawlings (1998) solves (4) for an initial horizon $N = N_0$. Then, until the final state lies in $\mathscr{X}_I$, N is increased according to a predefined iteration law which is initialized to $c = 0$. One approach would be to increment N by c $(N = N_0 + c)$, which yields the minimal horizon N such that $x_N \in \mathscr{X}_I$, i.e., $N_{\text{est}} = \bar{N}(x(0))$. An alternative is to increase N by a factor of 2 at each iteration, i.e., $N = 2^c N_0$, which results in fewer QPs to be solved at the cost of a larger N_{est} $(N_{\text{est}} \geqslant \bar{N}(x(0)))$. Note that this approach cannot be used to compute the PWA solution in Theorem 1 because it does not yield the horizon $\bar{N}_\mathscr{S}$ for a compact set $\mathscr{S}$.

Table 1
Time to compute the optimal control input on a 1.2 GHz PC with the approach in Scokaert and Rawlings (1998)

Example 1	Average-case (100 runs) (s)	Worst-case (s)
$N = N_0 + c$	5.29	10.13
$N = 2^c N_0$	0.80	2.48
$N = \bar{N}_{\mathscr{S}}$	0.47	0.56

N_0 is set to 1 and c is incremented by 1 at each iteration. The analysis is based on 100 random initial states.

3. Comparison of available techniques

In this section, we will discuss and compare available methods for solving CLQR and some of the drawbacks of the approaches of Scokaert and Rawlings (1998) and Chmielewski and Manousiouthakis (1996), that were introduced in the previous section, will be illustrated.

Example 1. Consider the system (Bemporad et al., 2002)

$$x(t+1) = \begin{pmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{pmatrix} x_t + \begin{pmatrix} 0.0609 \\ 0.0064 \end{pmatrix} u(t).$$

The task is to regulate the system to the origin while fulfilling the input constraint

$$-2 \leqslant u(t) \leqslant 2 \quad \forall t \geqslant 0.$$

We will solve this example in a set $\mathscr{S}$ of interest defined as

$$\mathscr{S} = \{x = (x_1\ x_2) \in \mathbb{R}^2 | -1000 \leqslant x_k \leqslant 1000, \quad i = 1, 2\}.$$

Note that this is not a constraint but merely an artificial bound on the state-space to be explored. The cost on the state is set to $Q = I$ and the input-cost is $R = 0.01$.

Applying the approach of Chmielewski and Manousiouthakis (1996) to Example 1, we obtain $N_{\text{est}} = 1.5 \times 10^7$ while the true minimal horizon is $\bar{N}_{\mathscr{S}} = 71$. Note that, to the authors' knowledge, there is currently no published technique to compute $\bar{N}_{\mathscr{S}}$. An algorithm for computing $\bar{N}_{\mathscr{S}}$ will be provided in Section 4. The optimization approach of Scokaert and Rawlings (1998) is well suited for small $\bar{N}_{\mathscr{S}}(x(0))$. However, in general, if a large section of the state-space is to be covered, this implies a large $\bar{N}_{\mathscr{S}}(x(0))$. For random values of $x(0) \in \mathscr{X}_f^\infty$ in Example 1, the run-times of the algorithm of Scokaert and Rawlings (1998) are presented in Table 1.

4. Computation of the constrained infinite time linear quadratic regulator

In this section, the main contribution of this paper is presented. We will provide an efficient algorithm to compute the PWA solution to the CLQR problem in (7) for a given set $\mathscr{S}$ of initial conditions. As a side product, the algorithm also computes $\bar{N}_{\mathscr{S}}$ defined in Lemma 1. The key idea of the algorithm is described next. We choose the terminal cost $\mathscr{Q}_f = P$, where P is the solution to the ARE, for the optimization problem in (3) and solve an mp-QP with prediction horizon N. From Theorem 4, we can conclude that for all states which enter the invariant set $\mathscr{X}_I$ introduced in Definition 3 in N steps, the infinite-horizon problem has been solved. Therefore the associated feedback law is infinite-horizon optimal. For the sake of clarity, we will first introduce our algorithm by applying it to a generic example, before we conclude this section with a more general description. We denote the set of feasible initial conditions of Problem (6) inside the compact set $\mathscr{S}$ as $\mathscr{S}_F = \mathscr{S} \cap \mathscr{X}_f^\infty$. The user defined set $\mathscr{S}$ is introduced as an artificial bound on the state-space to make sure $\mathscr{S}_F$ is bounded. In practice, $\mathscr{S}$ should be chosen to be very large.

We start the procedure by computing the positive-invariant unconstrained set $\mathscr{X}_I$ introduced in Definition 3. The polyhedron $\mathscr{X}_I = \mathscr{P}_0 = \{x \in \mathbb{R}^n | H_0 x \leqslant K_0\}$ can be computed as in Gilbert and Tan (1991) and is depicted in Fig. 1(a). Then, the algorithm finds a point $\bar{x}$ by stepping over a facet f of $\mathscr{X}_I$ with a small step ε. If (4) is feasible for horizon $N=1$ and $x(0)=\bar{x}$, the active constraints $\mathscr{A}_1^1(x(0))$ will define the neighboring polyhedron $\mathscr{P}_1 = \{x \in \mathbb{R}^n | H_1 x \leqslant K_1\}$ ($\bar{x} \in \mathscr{P}_1$, see Fig. 1(b)) (Bemporad et al., 2002). In order to avoid redundant exploration, one should keep track of the facets already explored. By Theorem 4, the finite time optimal solution computed above equals the infinite-time optimal solution if $x_N \in \mathscr{X}_I$. Therefore we extract from $\mathscr{P}_1$ the set of points that will enter $\mathscr{X}_I$ at the next time-step, provided that the optimal control law associated with $\mathscr{P}_1$ (i.e., $U_1^* = F_1 x(0) + G_1$) is applied. The *infinite-time polyhedron* ($\mathscr{ITP}$) is therefore defined by the intersection of the following polyhedra:

$$H_0 x_N \leqslant K_0, \tag{8}$$

$$H_r x(0) \leqslant K_r. \tag{9}$$

Eq. (8) is the reachability constraint (i.e., $x_N \in \mathscr{X}_I$) and (9) (with $r = 1$) defines the set of states for which the computed feedback law is feasible and optimal over N steps (see Bemporad et al., 2002 for details).

The identified region will be referred to as $\mathscr{ITP}_1^1$ ($\mathscr{ITP}_0^0 = \mathscr{X}_I$, see Fig. 1(c)) according to the following convention.

Definition 4. We define the rth infinite-time-polyhedron $\mathscr{ITP}_r^N$ as follows:

$$\forall x(0) \in \mathscr{ITP}_r^N, \quad \mathscr{A}_r^N(x(0)) = \text{constant},$$

and the reachability condition (8) holds. The optimal feedback law for $\mathscr{ITP}_r^N$ is defined by $\mathscr{A}_r^N$ (Bemporad et al., 2002) and ascertains that $x_N \in \mathscr{X}_I$. Once all redundant inequalities have been removed, this polyhedron has two types

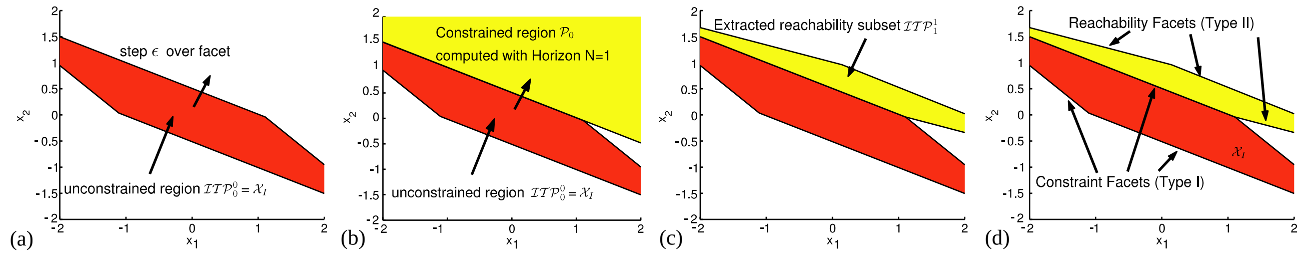

Fig. 1. Region exploration. (a) Compute positive invariant region $\mathcal{X}_I$ and step over facet with step-size ε. (b) Solve QP for new point with horizon N to create the first constrained region $\mathcal{P}_1$. (c) Compute reachability subset of $\mathcal{P}_1$ to obtain $\mathcal{ITP}_1$. (d) Identify facet types (Type I or II).

of facets:

Type I: The facet originated from constraint restrictions in (9).

Type II: The facet originated from reachability restrictions in (8).

The procedure for identifying the adjacent $\mathcal{ITP}$s is repeated for all facets f not previously explored, which originate from constraint restrictions (Type I). If a facet originates from the reachability restriction (Type II), we can conclude that the infinite-horizon optimal input sequence will not drive the states 'on the other side' of the facet into $\mathcal{X}_I$ in N steps. This distinction is depicted in Fig. 1(d). As depicted in Fig. 1(d), Type I facets are shared by all $\mathcal{P}$ and their associated reachability subsets $\mathcal{ITP}$, while all other facets are of Type II. Also note that we define all facets of $\mathcal{ITP}_0^0 = \mathcal{X}_I$ to be of Type II, i.e., all facets of $\mathcal{X}_I$ are explored at the first iteration (see Algorithm 1).

Once all facets have been explored for the finite horizon N, the horizon is increased to $N+1$ and the entire procedure is repeated. For all regions that have been previously computed, only facets of Type II are considered while for all newly identified regions, only facets of Type I are examined. This procedure is not applied to the facets which have already been explored. Note that the distinction between Type I and Type II facets merely serves to speed up the exploration procedure. It is not necessary for the algorithm to work correctly. The algorithm terminates as soon as all facets f of all $\mathcal{ITP}$s have been explored. A facet f is considered to be explored if the state $\bar{x}$ beyond f provides an $\mathcal{ITP}$ region or an infeasible problem (6) results for $x(0) = \bar{x}$. The presented procedure is summarized for a general problem in the following algorithm:

Algorithm 1.
function $\mathscr{C}$=computeC-LQR

(1) $\mathcal{ITP}_0^0 = \mathcal{X}_I$, $N = 0$, $r = 1$, $\mathscr{C} = \{\}$, $\mathscr{Z} = \{\}$;
(2) $\mathscr{C} = \mathscr{C} \cup \{\mathcal{ITP}_0^0, F_0 = K_{\text{LQR}}, G_0 = 0\}$;
(3) **repeat:**
(4) $N = N + 1$
(5) forall $(\mathcal{ITP}_i^q \in \mathscr{C}$ & $q < N)$, explore($\mathcal{ITP}_i^q$, *Type* II);
(6) forall $(\mathcal{ITP}_i^q \in \mathscr{C}$ & $q = N)$, explore($\mathcal{ITP}_i^q$, *Type* I);
(7) **until:** all facets f of all $\mathcal{ITP}_i^q \in \mathscr{C}$ are contained in $\mathscr{Z}$;
(8) $\bar{N}_{\mathscr{S}} = N$; return $\mathscr{C}$;

function explore($\mathcal{ITP}_i^q$, *facetType*);

(1) forall ($f \in$ facets($\mathcal{ITP}_i^q$) & $f \notin \mathscr{Z}$),
(2) if facet type of f is not *facetType*, goto 1;
(3) step over f and get $\bar{x}$;
(4) if $\exists \mathcal{ITP}_i^m \in \mathscr{C}$ s.t. $\bar{x} \in \mathcal{ITP}_i^m$, goto (10);
(5) solve (4) for $x(0) = \bar{x}$ with horizon N; if infeasible, goto (10);
(6) compute $\mathcal{ITP}_r^N$ according to (9) and (8);
(7) If $\mathcal{ITP}_r^N = \emptyset$, goto 1;
(8) Use $\mathscr{A}_r^N$ to compute F_r and G_r according to [3];
(9) $\mathscr{C} = \mathscr{C} \cup \{\mathcal{ITP}_r^N, F_r, G_r\}$; $r = r + 1$;
(10) $\mathscr{Z} = \mathscr{Z} \cup \{f\}$;
(11) end forall.

As before, N denotes the horizon for solving (4) and the integer r is a counter for the region number. The generated structure $\mathscr{C}$ is a list of all regions with their associated control law as well as a list containing all explored facets.

Theorem 5. *Algorithm* 1 *always converges in finite time, provided* $\mathscr{S}_F$ *is bounded and closed.*

Proof. If $\mathcal{X}_f^\infty$ is compact, a finite $\bar{N}_{\mathscr{S}}$ exists (Chmielewski & Manousiouthakis 1996). For a finite prediction horizon, the number of possible active constraint combinations is also finite. Since a region is uniquely identified by the active constraints, the associated region partition will consist of a finite number of regions. Since the algorithm increases the horizon N if no more regions are identified, the prediction horizon will eventually reach $\bar{N}_{\mathscr{S}}$. At this point, Algorithm 1 is identical to the one in Bemporad et al. (2002) and will therefore converge in finite time. □

It should be noted that in theory, $\bar{N}_{\mathscr{S}}$ and the convergence time might not be finite if $\mathscr{S}_F$ is open or unbounded. However, in practice this is not an issue. First, any compact set $\mathscr{S}$ will make $\mathscr{S}_F$ bounded. Second, if $\mathscr{S}_F$ has open boundaries, accumulation points will occur near those boundaries, i.e.

an infinite number of regions will be located in a bounded subset of the state-space. Since the step-size ε which is taken in the algorithm is finite, the point $\bar{x}$ will not provide a feasible solution to (4) once the regions are close to the open boundaries. The resulting partition will be an inner approximation of $\mathscr{X}_f^\infty$, whereby the size of the set can be adjusted by choice of the step-size ε. Therefore in practice the algorithm always converges in finite time, though not the entire set $\mathscr{X}_f^\infty$ may be covered by $\mathscr{ITP}$s if $\mathscr{X}_f^\infty$ has open boundaries, i.e. $\bigcup \mathscr{ITP}_r \subset \mathscr{X}_f^\infty$.

Theorem 6 shows that each state $x \in \mathscr{X}_f^\infty$ is unambiguously associated with one $\mathscr{ITP}$.

Theorem 6. *The intersection of the interior of $\mathscr{ITP}_i^N$ and $\mathscr{ITP}_j^k$ is non-empty, if and only if $i = j$ and $K = N$.*

Proof. "$\Leftarrow$" trivial.

"$\Rightarrow$", from Theorem 4, (9) and (8), we can conclude that the $\mathscr{ITP}$ region partition is identical to the finite-time region partition computed for a horizon of $N \geqslant \bar{N}_{\mathscr{S}}$. Therefore each region has a distinct set of active constraints and from Bemporad et al. (2002) we can conclude that if two $\mathscr{ITP}$s have a non-empty intersection then they are identical. □

The following theorems state some properties of the solution provided by Algorithm 1.

Theorem 7. *If we explore any given compact set $\mathscr{S}$ with Algorithm 1, the largest resulting horizon is equal to $\bar{N}_{\mathscr{S}}$, i.e.,*

$$\bar{N}_{\mathscr{S}} = \max_{\substack{\mathscr{ITP}_r^N \\ r=0,\dots,R}} N.$$

Proof. Since $\bar{N}_{\mathscr{S}}$ is defined as $\bar{N}_{\mathscr{S}} \triangleq \max_{x(0)\in\mathscr{S}_F} \bar{N}(x(0))$ with $\mathscr{S}_F = \mathscr{S} \cap \mathscr{X}_f^\infty$, we need to show that

$$\max_{x(0)\in\mathscr{S}_F} \bar{N}(x(0)) = \max_{\substack{\mathscr{ITP}_r^N \\ r=0,\dots,R}} N.$$

We will denote the maximum horizon of all $\mathscr{ITP}$s as $N_{\max}$. Consider an initial feasible state $\tilde{x} \in \mathscr{S}$ which reaches $\mathscr{X}_1$ in exactly $\bar{N}_{\mathscr{S}}$ steps if the optimal PWA control law is applied. This state would not be covered by any $\mathscr{ITP}$, if $N_{\max} < \bar{N}_{\mathscr{S}}$, since (8) would be violated. Since Algorithm 1 always converges, the entire feasible set $\mathscr{X}_f^\infty$ is covered by $\mathscr{ITP}$s and therefore $N_{\max} \geqslant \bar{N}_{\mathscr{S}}$. However, $N_{\max}$ can only be greater than $\bar{N}_{\mathscr{S}}$ if a region with horizon $N = \bar{N}_{\mathscr{S}}$ is bounded by a reachability facet (Type II). Only then would the algorithm increase N further; otherwise all facets would be covered for a horizon $N = \bar{N}_{\mathscr{S}}$ and the exploration would end. Since, by definition, all $x \in \mathscr{X}_f^\infty$ can reach $\mathscr{X}_I$ in at most $\bar{N}_{\mathscr{S}}$ steps, a region with horizon $N = \bar{N}_{\mathscr{S}}$ cannot be bounded by a Type II facet. These states have no impact on the result because $\bar{N}(x)$ is not defined for infeasible states (i.e., $x \in (\mathscr{S} \setminus \mathscr{X}_f^\infty)$). Therefore $N_{\max} = \bar{N}_{\mathscr{S}}$. □

Lemma 2. *In the infinite-horizon polyhedral state-space region partition a state can only remain within one region for at most one time step (except for $\mathscr{X}_I$).*

Proof. Follows from Remark 1 and the implementation of Algorithm 1. □

Theorem 8. *The union of all $\mathscr{ITP}$'s computed with Algorithm 1 is positive invariant if $\mathscr{S}_F$ is bounded and closed.*

Proof. Follows from Lemma 2 and Theorems 1 and 5. The state will always move to a region with horizon $N-1$ at the next time step until the unconstrained region $\mathscr{X}_I$ is reached. □

Remark 2. As previously stated, $\bigcup \mathscr{ITP}_r \subset \mathscr{X}_f^\infty$ if $\mathscr{X}_f^\infty$ has open boundaries or $\mathscr{S} \subset \mathscr{X}_f^\infty$. The union of all regions is generally not invariant in this case. However, the union of regions can easily be made invariant by modifying the on-line application of the feedback law. If the trajectory enters part of the state space where no region was found, the open loop solution of the previous region is applied. Since the open-loop is equal to the RHC closed-loop solution for the infinite horizon controller (Remark 1), optimality and invariance is preserved.

Algorithm 1 has two main advantages over the algorithm presented by Bemporad et al. (2002). First, Algorithm 1 does not artificially partition regions with unique active constraints. Therefore excessive partitioning is avoided which may result in fewer regions. Furthermore, the artificially partitioned regions do not need to be merged a posteriori. Second, extensive simulation has shown that the removal of redundant constraints may take up more than half of the overall computation time of the multi-parametric solution to (4) (Tøndel, Johansen & Bemporad, 2001). The initial polyhedral representation $\mathscr{P}_r$ contains redundant constraints which need to be removed in order to obtain a minimal representation of the polyhedra. For any given region, the number of redundant constraints in Bemporad et al. (2002) is always greater or equal to the number of redundant constraints obtained with Algorithm 1, since the prediction horizon is iteratively increased. This is because the number of constraints is linear in the prediction horizon. The effort of removing redundant constraints is therefore greatly reduced by using the smallest possible horizon length for each region.

It is possible to extend Algorithm 1 to speed up the identification of the active PWA feedback law for a given $x(0)$. Since the closed-loop solution is equal to the open-loop solution (Remark 1) and a state only remains in one region for one time-step (Lemma 2), it is possible to merge regions according to Bemporad, Fukuda and Torrisi (2001). With this method, regions with the same PWA control law on the first input are joined. If this procedure is applied to the PWA controller partition obtained for Example 1, the number of regions is reduced from 185 to 53 (see Fig. 2(b)).

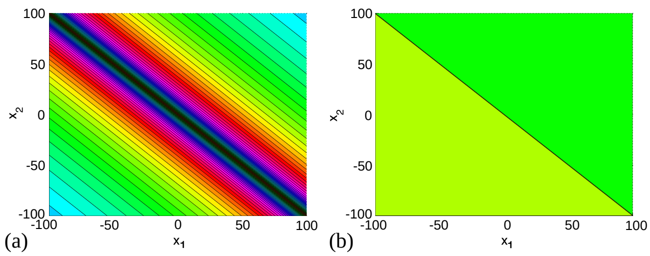

Fig. 2. Infinite-time region partitions obtained by applying Algorithm 1 on Example 1. (a) Close-Up of the controller partition. The number of regions is $R = 185$. (b) Close-Up of the reduced controller partition. The number of regions is $R = 45$.

It should be noted that the PWA controller may consist of a very large number of regions. The resulting partition may therefore be computationally prohibitive for on-line implementation in the form of a look-up table. However, as the next section will show, the CLQR obtained with Algorithm 1 may also be of relatively low complexity. Furthermore, the procedure described in Algorithm 1 can easily be adjusted to compute the finite horizon controller which may yield a considerable off-line speedup compared to other algorithms (Bemporad et al., 2002; Baotić, 2002).

5. Results

In this section, we will compare the computation time needed to obtain the PWA solution using Algorithm 1 with other comparable approaches. Subsequently, we will compare the necessary effort to identify the active feedback law with the time needed to solve a QP. The region partitions that were obtained for Example 1 can be seen in Fig. 2(a). The times needed to compute the PWA solution for Example 1 using various algorithms are given in Table 2. The abbreviation mp-QP in Table 2 signifies that (4) is solved explicitly for horizon $N = \bar{N}_{\mathscr{S}}$, as described in Section 2. Note that there is currently no algorithm to compute $\bar{N}_{\mathscr{S}}$. Also note that the run-time for both algorithms could be further reduced by almost 50% by taking into account that symmetric constraints produce symmetric region partitions (Tøndel et al., 2001). To the authors' knowledge, all comparable algorithms which were published to date, have significantly larger run-times, even under the assumption that the horizon $\bar{N}_{\mathscr{S}}$ is known. This is shown in Table 2 and was verified on numerous other examples Table 3.

Assuming the PWA feedback partition has been computed, we will now compare the on-line times necessary to extract the PWA feedback law with the iterative QP algorithm of Scokaert and Rawlings (1998) which was presented in Section 3. Also note that the PWA solution can give hard bounds on the worst case run-time, whereas a QP-based solution cannot provide such a bound without knowledge of $\bar{N}_{\mathscr{S}}$.

Table 2
Comparisons of computation times to compute the PWA solution for Example 1 on a Pentium III, 1.2 GHz

Algorithm 1	22.43 s
mp-QP [1], $N = \bar{N}_{\mathscr{S}} = 71$	37.99 s

The solution consists of 185 regions.

Table 3
Comparisons of the computation time necessary to identify the optimal input on a Pentium III, 1.2 GHz

Example 1	Average-case (s)	Worst-case (s)
$N = N_0 + k$	5.29	10.13
$N = 2^k N_0$	0.88	2.55
$N = \bar{N}_{\mathscr{S}}$	0.47	0.56
Algorithm 1 (185 regions)	0.0042	0.0056

N_0 is set to 1 and k is incremented by 1 at each iteration. The analysis is based on 100 random initial states.

A scheme to decrease the time necessary to find the PWA feedback law further, was published by Borrelli, Baotić, Bemporad, and Morari (2001). The authors use a cost-function to identify the optimal feedback law, greatly decreasing the required storage-space and additionally reducing identification times for the controller by a factor of 2 for the examples given here. Note that, according to Theorem 7, Algorithm 1 can be used to compute $\bar{N}_{\mathscr{S}}$ exactly. As an alternative to the look-up table, this value could subsequently be used to speed up the algorithm in Sznaier and Damborg (1987), Scokaert and Rawlings (1998).

6. Conclusions

An efficient algorithm for solving the infinite-horizon constrained linear quadratic regulator (CLQR) problem was presented. The algorithm is based on multi-parametric quadratic programming and reachability analysis. This combination outperforms all comparable mp-QP approaches in terms of off-line computation speed. Furthermore, when compared to on-line computation procedures, the time necessary to obtain the optimal input was significantly decreased, making CLQR an attractive solution even for fast processes. Moreover, a worst-case run-time can be guaranteed. In addition, a method to compute the horizon $\bar{N}_{\mathscr{S}}$ for compact sets has been presented for the first time. Exact knowledge of $\bar{N}_{\mathscr{S}}$ can serve to improve the performance of a wide array of algorithms presented in literature.

The presented algorithm as well as a more detailed technical report is available from http://control.ee.ethz.ch

Acknowledgements

The authors would like to acknowledge the contributions of an anonymous reviewer, who pointed out the need to introduce the set $\mathscr{S}_F = \mathscr{S} \cap \mathscr{X}_f^{\infty}$ in addition to $\mathscr{S}$.

References

Baotić, M. (2002). *An efficient algorithm for multi-parametric quadratic programming*. Technical report AUT02-04, Automatic Control Laboratory, ETH Zurich, Switzerland.

Bemporad, A., Fukuda, K., & Torrisi, F. D. (2001). Convexity recognition of the union of polyhedra. *Computational Geometry*, *18*, 141–154.

Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, *38*(1), 3–20.

Borrelli, F. (2003). *Constrained optimal control of linear and hybrid systems, Lecture notes in control and information sciences*, Vol. 290. Berlin: Springer.

Borrelli, F., Baotić, M., Bemporad, A., & Morari, M. (2001). Efficient on-line computation of constrained optimal control. In *Proceedings of the 40th IEEE conference on decision and control*.

Chisci, L., & Zappa, G. (1999). Fast algorithm for a constrained infinite horizon LQ problem. *International Journal of Control*, *72*(11), 1020–1026.

Chmielewski, D., & Manousiouthakis, V. (1996). On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, *29*(3), 121–130.

Cuttler, C.R., & Ramaker, B.C. (1980). Dynamic matrix control—a computer control algorithm. In *Proceedings of the american control conference*, Vol. WP5-B, San Francisco, USA.

Gevers, M., Bitmead, R. R., & Wertz, V. (1990). *Adaptive optimal control, the thinking man's GPC*. Englewood Cliffs, NJ: Prentice-Hall.

Gilbert, E. G., & Tan, K. (1991). Linear systems with state and control constraints: The theory and applications of maximal output admissible sets. *IEEE Transactions on Automatic Control*, *36*(9), 1008– 1020.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814.

Rawlings, J. B., & Muske, K. R. (1993). The stability of constrained receding-horizon control. *IEEE Transactions on Automatic Control*, *38*, 1512–1516.

Richalet, J., Rault, A., Testud, A. L., & Papon, J. (1978). Model predictive heuristic control-application to industrial processes. *Automatica*, *14*, 413–428.

Scokaert, P. O. M., & Rawlings, J. B. (1998). Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, *43*(8), 1163–1169.

Sznaier, M., & Damborg, M. J. (1987). Suboptimal control of linear systems with state and control inequality constraints. In *Proceedings of the 26th IEEE Conference on decision and control*, Vol. 1 (p. 761–762).

Tøndel, P., Johansen, T. A., & Bemporad, A. (2001). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In *Proceedings of the 40th IEEE conference on decision and control*.

Pascal Grieder received a diploma in electrical engineering in 2001 from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. He conducted his diploma thesis research at Stanford University before returning to Zurich as a Dr.sc.tech. student at the Automatic Control Laboratory of ETH. His research interests are in receding horizon control, hybrid systems and computational geometry.

Francesco Borrelli was born in Milano, Italy in 1974. He received the "Laurea" degree in computer science engineering in 1998 from the University of Naples "Federico II", Italy. In 2002 he received the Ph.D. from the ETH Zurich. He has been a post-doc at the Automatic Control Laboratory of the ETH Zurich. Currently he is a contract assistant professor at the Department of Aerospace Engineering and Mechanics at the University of Minnesota and a consultant for Honeywell Laboratories in Minneapolis (USA). His research interests include constrained optimal control, model predictive control, robust control, parametric programming, singularly perturbed systems and automotive applications of automatic control.

Fabio Torrisi received the master degree in Computer Engineering in 1999 from the University of Florence, Italy and the Ph.D. in 2003 from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. His research interest is in the area of hybrid systems, computational geometry, model predictive control, software engineering, and embedded systems.

Manfred Morari was appointed head of the Automatic Control Laboratory at the Swiss Federal Institute of Technology (ETH) in Zurich. Before that he was the McCollum-Corcoran Professor of Chemical Engineering and Executive Officer for Control and Dynamical Systems at the California Institute of Technology. He obtained the diploma from ETH Zurich and the Ph.D. from the University of Minnesota, both in chemical engineering. His interests are in hybrid systems and the control of biomedical systems. In recognition of his research contributions, he received numerous awards, among them the Donald P. Eckman Award of the Automatic Control Council, the Allan P. Colburn Award and the Professional Progress Award of the AIChE, the Curtis W. McGraw Research Award of the ASEE, Doctor Honoris Causa from Babes-Bolyai University, and was elected to the National Academy of Engineering (U.S.). Professor Morari has held appointments with Exxon and ICI plc and serves on the technical advisory board of several major corporations.