Explicit Sub-optimal Linear Quadratic Regulation with State and Input Constraints

Tor A. Johansen [a] Idar Petersen [b] Olav Slupphaug [a,c]

[a] *Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.*

[b] *SINTEF Electronics and Cybernetics, N-7465 Trondheim, Norway.*

[c] *ABB Industry, Oslo, Norway.*

Abstract

Optimal feedback solutions to the infinite horizon LQR problem with state and input constraints based on receding horizon real-time quadratic programming are well known. In this paper we develop an explicit solution to the same problem, eliminating the need for real-time optimization. It is shown that the resulting feedback controller is piecewise linear. This explicit functional structure is exploited for efficient real-time implementation. A suboptimal strategy, based on a suboptimal choice of a finite horizon and imposing additional limitations on the allowed switching between active constraint sets on the horizon, is suggested in order to address the computer memory and processing capacity requirements of the explicit solution.

Key words: Constrained control; optimal control; linear systems

1 Introduction

Consider the linear time-invariant system

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

where $x \in R^n$, and $u \in R^r$. The optimal constrained LQ feedback controller minimizes the infinite horizon quadratic cost

⋆ Corresponding author: Tor.Arne.Johansen@itk.ntnu.no

⋆⋆The work of the first and second authors were in part supported by the European Commission under the ESPRIT Long Term Research project 28104 H^2C.

$$J(u(t), u(t+1), u(t+2), ...; x(t)) = \sum_{\tau=t}^{\infty} l_{QR}(x(\tau), u(\tau)) \quad (2)$$

$$l_{QR}(x, u) = x^T Q x + u^T R u \quad (3)$$

subject to the linear constraints

$$Gx(\tau + 1) \leq g \quad (4)$$

$$Hu(\tau) \leq h \quad (5)$$

for all $\tau \geq t$, where $R > 0$, $Q \geq 0$, $G \in R^{q \times n}$, and $H \in R^{p \times r}$. It is assumed that $g > 0$ and $h > 0$ (where the inequalities are elementwise since g and h are vectors) to ensure that the origin is an interior point in the admissible region. The *optimal cost function* is defined as

$$V(x(t)) = \min_{u(t), u(t+1), ...} J(u(t), u(t+1), u(t+2), ...; x(t)) \quad (6)$$

where the minimization is subject to the dynamics of the system (1), and the constraints (4)-(5) are imposed at every time instant $\tau \in \{t, t+1, t+2, ...\}$ on the trajectory. The cost of moving from the state $x(t)$ to the origin in an optimal manner is given by $V(x(t))$. Consider the following Hamilton-Jacobi-Bellman (HJB) equation

$$0 = \min_{\substack{u(\tau) \in R^r, Gx(\tau+1) \leq g, Hu(\tau) \leq h \\ \tau \in \{t, t+1, t+2, ..., t+N-1\}}} \left(V(x(t+N)) - V(x(t)) + \sum_{\tau=t}^{t+N-1} l_{QR}(x(\tau), u(\tau)) \right) \quad (7)$$

where $N \geq 1$ is some horizon, and $V(0) = 0$. This equation characterizes the optimal cost function and optimal control action for the problem when N is so large that there are no active or violated constraints beyond this horizon, since the unconstrained LQ solution is optimal beyond the horizon, (Sznaier and Damborg 1987, Chmielewski and Manousiouthakis 1996, Scokaert and Rawlings 1998). Under the assumptions of feasibility, non-explicit optimal solutions to the HJB (7) can be computed using real-time quadratic programming, where a finite-dimensional optimization problem is achieved since $V(x(t+N)) = x^T(t+N)Px(t+N)$, where P is the solution to the algebraic Riccati equation associated with the unconstrained LQR. This is an optimal approach, in contrast to common suboptimal (approximate) approaches used in model predictive control with a finite horizon cost function approximation or a finite input move horizon, e.g. (Keerthi and Gilbert 1988, Rawlings and Muske 1993). In any case, the real-time quadratic programming imposes severe limitations on the achievable sample rate that may discourage the application of this approach in many cases.

Recently, Bemporad *et al.* (2000) (see also (Bemporad *et al.* 1999) for further details) derived an optimal explicit solution to the constrained LQR problem, in the sense that no real-time quadratic program needs to be solved. The explicit controller was computed offline using multi-parametric quadratic programming. The constrained LQR problem is viewed as a quadratic program parameterized by the state x, and the multi-parametric quadratic programming approach essentially finds an explicit solution for all x within an arbitrary subset of the state space. The resulting optimal controller was proved to be a continuous piecewise linear function defined on a polyhedral partitioning of the state-space. Related characterizations of the piecewise linear nature of constrained LQ control are derived for some cases in (Seron *et al.* 2000).

In this paper we also seek an explicit solution to this problem in order to reduce the demand for real-time computations. However, in order to address the restrictions imposed by real-time applications on both computer memory and processing capacity, a (possibly) suboptimal strategy is developed. Hence, we introduce a mechanism to trade performance for computational advantages. The present approach can be seen as an extension of (Bemporad *et al.* 1999) (although the main parts were developed independently, see also (Johansen *et al.* 2000*b*)), with the following main differences:

- Here we consider a suboptimal strategy where an approximation to the optimal cost function is utilized and we impose restrictions on the allowed switching between the active constraint sets during the prediction horizon. As a limiting case, the presented approach is equivalent to the optimal explicit LQR of Bemporad *et al.* (1999).
- Due to the sub-optimality of the controller, its performance is not known a priori, so one may rely on computational analysis tools which can be used to compute upper and lower bounds on suboptimal performance as well as assess stability (Johansson and Rantzer 1998, Rantzer and Johansson 2000).
- The solution strategies are different; the present approach is not based on multi-parametric quadratic programming. Both strategies leads to a piecewise linear (PWL) controller. While the exact approach leads to a continuous PWL function on a polyhedral partitioning, the suboptimal approach will not do so in all cases.
- The present approach explicitly addresses the possibility of infeasibility in the design by minimizing the constraint violation, while in the approach of Bemporad *et al.* (1999) a method based on slack variables is used (Zheng and Morari 1995).
- The present design approach includes practical modifications to avoid high gain feedback (which may result in sliding mode like behavior and chattering control) at the boundary of the state constraints due to the choice of a short horizon.

In (Chisci 1999), the structure of the finite-dimensional real-time quadratic program is utilized to developed a fast QP algorithm based on active sets for constrained LQR. In an alternative approach, (Sznaier and Damborg 1990), a finite discrete search problem is achieved by quantization of the set of admissible inputs. The computational complexity of their search problem is typically much larger, and a different type of approximation is introduced due to the quantization of the inputs. The approach of (Wredenhagen and Belanger 1994) defines a nested set of elliptic regions of the state space, each containing the origin, where different LQ optimal feedback laws are designed with different Q matrices. The idea is to reduce the gain of the feedback in order to avoid saturation when far away from the origin. However, only input constraints are considered.

This paper is organized as follows: In section 2 it is shown that the HJB equation can be decomposed into two nested parts by considering the finite number of combinations of active constraint sets. It is shown in section 3 that the solution of the innermost part of the HJB equation is an affine state feedback, when the active constraint set is given. The outer part of the HJB equation, addressed in section 4, is to determine which constraints should be active at any current state $x(t)$. Some aspects of sub-optimality, computational complexity and real-time implementation are discussed in section 5.

2 Controller decomposition

The main idea is to introduce active constraint set sequences as a formalism to decompose the HJB equation. This decomposition is discussed in this section.

2.1 Active constraint set sequences

A single inequality constraint $d_i^T z \geq e_i$ is said to be an *active constraint* if $d_i^T z = e_i$, where d_i is a vector, e_i is a scalar and the vector z is the design variable. Let $D^T = (d_1, d_2, ..., d_m)$ and $e^T = (e_1, e_2, ..., e_n)$. An *active constraint set* associated with some set of inequality constraints $Dz \leq e$ is the set of indices to those constraints that are active. The active constraint set may be empty, meaning that no constraints are active. At each sample one may impose a number of equality constraints (selected from the inequality constraints (4) and (5)) on the states and inputs that, except for degenerate cases, is less than or equal to the number of inputs r. This selection of constraints is the *active constraint set* associated with that sample. A sequence of active constraint sets imposed at each sample on the horizon finite N is called an *active constraint set sequence.*

A naive solution strategy to the optimal explicit LQR problem is simply to evaluate all feasible active constraint set sequences on a sufficiently large horizon N. This naive solution strategy to the optimal explicit LQR will indeed have offline computational disadvantages compared to the multi-parametric quadratic programming approach of (Bemporad *et al.* 1999) since the number of candidate active constraint set sequences increases very rapidly with the horizon N and the number of inputs r and states n. However, it has the advantage that it can be easily modified to determine suboptimal explicit LQR solutions with drastically reduced offline and real-time computational demands. The main idea in the present work is to use a smaller horizon N than optimal, and in addition to reduce the flexibility in the active constraint set sequence by allowing changes in the active constraint set to be made only at a limited number of predetermined samples.

Suppose the set of indices α is associated with the active input constraints in (5) at some sample, and the set of indices β is associated with the active state constraints (4) at the same sample. Then (α, β) is an active constraint set. Next, suppose we define allowed switching times as follows: $0 = N_1 < N_2 < \cdots < N_S < N$. For example, if $S = 3$, $N_1 = 0$, $N_2 = 3$ and $N_3 = 7$ there will be 3 subintervals $\{t, t+1, t+2\}$, $\{t+3, t+4, t+5, t+6\}$, and $\{t+7, t+8, t+9\}$ with associated fixed active constraint sets $(\alpha_1, \beta_1), (\alpha_2, \beta_2), (\alpha_3, \beta_3)$, respectively. In general, these active constraint sets lead to an *active constraint set sequence* $((\alpha_1, \beta_1), (\alpha_2, \beta_2), ..., (\alpha_{N_S}, \beta_{N_S}))$ that together with $(N_1, N_2, ..., N_S)$ and N define the active constraint set imposed at each sample on the horizon. This means that the constraints indexed by each active constraint set are imposed on the associated interval, leading to the following set of equality constraints:

$$\left.\begin{array}{c} H_{\alpha_1} u(t) = h_{\alpha_1} \\ H_{\alpha_1} u(t+1) = h_{\alpha_1} \\ \vdots \\ H_{\alpha_{N_S}} u(t+N-1) = h_{\alpha_{N_S}} \end{array}\right\} \tag{8}$$

and

$$\left.\begin{array}{c} G_{\beta_1}\left(Ax(t) + C_N \tilde{E}_1 \tilde{u}(t)\right) = g_{\beta_1} \\ G_{\beta_1}\left(A^2 x(t) + C_N \tilde{E}_2 \tilde{u}(t)\right) = g_{\beta_1} \\ \vdots \\ G_{\beta_{N_S}}\left(A^N x(t) + C_N \tilde{E}_N \tilde{u}(t)\right) = g_{\beta_{N_S}} \end{array}\right\} \tag{9}$$

We have introduced the matrix $C_\tau = (A^{\tau-1}B | A^{\tau-2}B | \cdots | B)$, the $rN \times rN$-

matrix $\tilde{E}_\tau$ defined by

$$\tilde{E}_\tau = \begin{pmatrix} 0 & 0 \\ I_{r\tau \times r\tau} & 0 \end{pmatrix} \tag{10}$$

and applied the well known formula $x(t + \tau) = A^\tau x(t) + C_N \tilde{E}_\tau \tilde{u}(t)$ where $\tilde{u}(t) = (u^T(t),\ u^T(t+1),\ \cdots,\ u^T(t+N-1))^T$. Removing from (8) and (9) equations that are a priori known to be infeasible and duplicated equations, (8) and (9) may be stacked into the following set of equations

$$L_k \tilde{u}(t) = M_k x(t) + M_k^g g + M_k^h h \tag{11}$$

where k is an index in the index set $\mathcal{C} = \{0, 1, 2, ..., N_K - 1\}$ enumerating the finite set of all active constraint set sequences generated by the constraints (5), (4) and the division into subintervals. For later use, let $k_0 \in \mathcal{C}$ be the index to the active constraint set sequence with no active constraints, and define the $r \times rN$ matrix $E_\tau = (0_{r\times r}, ..., 0_{r\times r}, I_{r\times r}, 0_{r\times r}, ...0_{r\times r})$ where the $I_{r\times r}$ is at the τ-th $r \times r$ block.

2.2 Decomposition of the HJB equation

In this section we consider the minimization problem on the RHS of (7) with the stated constraints, which is a strictly convex problem whose solution is characterized by the Karush-Kuhn-Tucker conditions. However, since these conditions involve inequalities, the Karush-Kuhn-Tucker conditions provide an implicit solution that does not lead to an explicit state-feedback implementation of the controller. This motivates a simple decomposition of the minimization in (7) into two nested parts where one part only involves equality constraints and the other part is a discrete optimization problem over all allowed active constraint set sequences. The part that involves equality constraints can then be solved explicitly offline, while the discrete optimization problem can also be solved offline or reduced to a simpler problem and then solved in real-time in a efficient manner. The following result is then evident from (Chmielewski and Manousiouthakis 1996):

Theorem 1 *(Nested HJB equation) Assume the minimum in the HJB equation (7) exists. With N sufficiently large and no restrictions on the active constraint set sequences allowed switching times ($S = N$), the HJB equation (7) is equivalent to*

$$0 = \min_{k\in\mathcal{C}} \left(\min_{\substack{\tilde{u}(t)\in R^{rN} \\ L_k\tilde{u}=M_kx(t)+M_k^g g+M_k^h h}} \left(V(x(t+N)) - V(x(t)) + \sum_{\tau=0}^{N-1} l_{QR}(x(t+\tau), E_{\tau+1}\tilde{u}(t)) \right) \right) \tag{12}$$

where the outer minimization is subject to the constraints

$$HE_\tau \tilde{u}_k^*(x(t)) \leq h \tag{13}$$

$$G(A^\tau x(t) + C_N\tilde{E}_\tau \tilde{u}_k^*(x(t))) \leq g \tag{14}$$

for all $1 \leq \tau \leq N$ *and* $\tilde{u}_k^*(x(t))$ *is the* $\tilde{u}(t)$ *solving the innermost optimization problem in (12).* □

Determining the optimal cost function V is in general a difficult problem, so similar to (Bemporad *et al.* 1999, Rantzer and Johansson 2000) we utilize a lower bound $\underline{V}$ as a suboptimal approximation in the control design. Any loss of performance resulting from this approximation as well as sub-optimality due to restrictions on the allowed active constraint set switching times may be analyzed using the tools given in (Johansson and Rantzer 1998, Rantzer and Johansson 2000).

Lemma 1 *A lower bound on the optimal cost function is given by* $\underline{V}(x) = x^T\underline{P}x \leq V(x)$ *where the matrix* $\underline{P} = \underline{P}^T$ *is the positive definite solution of the algebraic Riccati equation corresponding to the unconstrained LQR problem:*

$$A^T\underline{P}A - \underline{P} - A^T\underline{P}B(B^T\underline{P}B + R)^{-1}B^T\underline{P}A + Q = 0 \tag{15}$$

Proof. The result follows immediately from the observation that constraining the input will never decrease the value of the optimal cost function (Sznaier and Damborg 1990). □

Note that $\underline{V}(x) = V(x)$ for x in any compact set if N is sufficiently large (Chmielewski and Manousiouthakis 1996). For a given active constraint set sequence (with index $k \in \mathcal{C}$) this leads to the problem

$$\tilde{u}_k^*(x(t)) = \arg \min_{\substack{\tilde{u}(t)\in R^{rN} \\ L_k\tilde{u}(t)=M_kx(t)+M_k^g g+M_k^h h}} \underline{I}(\tilde{u}(t), x(t)) \tag{16}$$

where

$$\underline{I}(\tilde{u}(t), x(t)) = \underline{V}(x(t+N)) - \underline{V}(x(t)) + \sum_{\tau=0}^{N-1} l_{QR}(x(t+\tau), E_{\tau+1}\tilde{u}(t)) \tag{17}$$

and the outer finite discrete optimization problem of (12) is restated as

$$k^*(x) = \arg\min_{k\in\mathcal{C}} \varphi_k(x) \tag{18}$$
$$\varphi_k(x) = \underline{I}(\tilde{u}_k^*(x), x) \tag{19}$$

where the minimization is subject to

$$HE_\tau \tilde{u}_k^*(x) \le h \tag{20}$$
$$G(A^\tau x + C_N \tilde{E}_\tau \tilde{u}_k^*(x)) \le g \tag{21}$$

for all $1 \le \tau \le N$.

Note that the minimum $k^*(x)$ need not be unique. In this case $k^*(x)$ is selected according to some pre-ordering of the candidate minima. The optimization problem (18)-(21) is feasible if and only if $x \in X^F$, where

$$X^F = \bigcup_{k\in\mathcal{C}} X_k^F \tag{22}$$
$$X_k^F = \left\{ x \in R^n \mid HE_\tau \tilde{u}_k^*(x) \le h,\ G(A^\tau x + C_N \tilde{E}_\tau^T \tilde{u}_k^*(x)) \le g,\ \text{for } 1 \le \tau \le N \right\} \tag{23}$$

For $x \in X^F$, the suboptimal constrained LQR is given by $u^*(x) = E_1 \tilde{u}_{k^*(x)}^*(x)$. If $x \notin X^F$, we relax the problem by allowing minimum violation of some of the constraints according to some priority. Constraints that may be relaxed are called "soft" constraints (with indices in the constraint sets α_{soft} and β_{soft}), as opposed to "hard" constraints (with indices in the constraint sets α_{hard} and β_{hard}), which can not be relaxed under any circumstances. Hence, for $x \notin X^F$, we minimize the criterion

$$\begin{aligned}\nu_k(x) = &\sum_{\tau=1}^{N} \omega_{1,\beta_{soft}}^T \max(0, G_{\beta_{soft}}(A^\tau x + C_N \tilde{E}_\tau \tilde{u}_k^*(x)) - g_{\beta_{soft}}) \\ &+ \sum_{\tau=1}^{N} \omega_{2,\alpha_{soft}}^T \max(0, H_{\alpha_{soft}} E_\tau \tilde{u}_k^*(x) - h_{\alpha_{soft}})\end{aligned} \tag{24}$$

with respect to $k \in \mathcal{C}$ subject to "hard" constraints for $1 \le \tau \le N$

$$G_{\beta_{hard}} \left(A^\tau x + C_N \tilde{E}_\tau \tilde{u}_k^*(x) \right) \le g_{\beta_{hard}} \tag{25}$$
$$H_{\alpha_{hard}} E_\tau \tilde{u}_k^*(x) \le h_{\alpha_{hard}} \tag{26}$$

The constant positive vectors $\omega_{1,\alpha_{soft}}$ and $\omega_{2,\beta_{soft}}$ are weights that capture some prioritization among the soft constraints. The optimization problem (24)-(26)

is feasible when $x \in X^R$, where

$$X^R = \bigcup_{k \in \mathcal{C}} X_k^R \tag{27}$$

$$X_k^R = \left\{ x \in R^n - X^F \mid \text{such that } (25) - (26) \text{ holds} \right\} \tag{28}$$

If $x \notin X^F \cup X^R$, i.e. no active constraint set sequence in $\mathcal{C}$ gives a control input that is feasible with respect to the hard (non-relaxable) constraints on the horizon, the controller fails. Let the solution to (24)-(26) be denoted $k^*(x)$ and the associated control input $\tilde{u}^*_{k^*(x)}(x)$. Furthermore, let $X = X^F \cup X^R$ and define for $x \in X$ the control input of the suboptimal constrained LQR:

$$u^*(x) = E_1 \tilde{u}^*_{k^*(x)}(x) \tag{29}$$

The resulting PWL control structure may be summarized as follows. There is a number of affine feedbacks where each affine feedback is designed with the objective of minimizing the LQ cost function subject to the state and input trajectories moving on a specific active constraint set sequence. The affine state feedbacks are designed offline by solving (16) as described in section 3, so the real-time computations amount to selecting which affine state feedback to apply at a given state $x(t)$. This amounts to solving (18)-(21) (or (24)-(26) in case of infeasibility), which is addressed in section 4. Together, this provides a sub-optimal solution to the HJB equation.

3 Computing gain matrices

In this section we first present the solution to the optimization problem (16), for a fixed active constraint set sequence. Next, we present an example and introduce some modifications. The expression (17) for $\underline{I}$ can be formulated as follows:

$$\underline{I}(\tilde{u}, x) = x^T S_1 x + 2x^T S_2 \tilde{u} + \tilde{u}^T S_3 \tilde{u} \tag{30}$$

where

$$S_1 = Q + A^T Q A + (A^2)^T Q A^2 + ... + (A^{N-1})^T Q A^{N-1} + (A^N)^T \underline{P} A^N \tag{31}$$

$$S_2 = A^T Q C_N \tilde{E}_1 + ... + (A^{N-1})^T Q C_N \tilde{E}_{N-1} + (A^N)^T \underline{P} C_N \tag{32}$$

$$S_3 = \tilde{R} + \tilde{E}_1^T C_N^T Q C_N \tilde{E}_1 + ... + \tilde{E}_{N-1}^T C_N^T Q C_N \tilde{E}_{N-1} + C_N^T \underline{P} C_N \tag{33}$$

and the block diagonal $rN \times rN$-matrix $\tilde{R}$ is defined by $\tilde{R} = \text{diag}(R, R, ..., R)$.

Theorem 2 *(Gain matrices) Consider a fixed active constraint set sequence with index $k \in \mathcal{C}$. For any $x \in X^F$, the solution to the constrained quadratic optimization problem (16) is given by the affine state feedback*

$$\tilde{u}_k^*(x) = \begin{cases} K_{k,2}x, & \text{if } \ k = k_0 \\ K_{k,1}^g g + K_{k,1}^h h + K_{k,2}x, & \text{if } \ k \neq k_0 \end{cases} \tag{34}$$

where $K_{k,2} = -S_3^{-1}S_2^T$ for $k = k_0$, and for $k \neq k_0$

$$K_{k,1}^g = S_3^{-1}L_k^T \left(L_k S_3^{-1} L_k^T\right)^{-1} M_k^g$$
$$K_{k,1}^h = S_3^{-1}L_k^T \left(L_k S_3^{-1} L_k^T\right)^{-1} M_k^h$$
$$K_{k,2} = -S_3^{-1}\left(\left(I - L_k^T\left(L_k S_3^{-1} L_k^T\right)^{-1} L_k S_3^{-1}\right) S_2^T - L_k^T \left(L_k S_3^{-1} L_k^T\right)^{-1} M_k\right)$$

Proof. Note that (16) is strictly convex since $R > 0$, and it suffices to consider only first order optimality conditions, which is straightforward (Johansen *et al.* 2000*a*). □

Observe that in the case of no active constraints, then (34) takes the form of the well known unconstrained LQR solution, namely $u(x) = -(B^T(Q + \underline{P})B + R)^{-1}B^T(Q + \underline{P})Ax$. The affine state feedback (34) is parameterized such that individual constraints can be deactivated and the constraint limits may be changed on-line without changing the gain matrices.

Example 1: Double integrator with input and state constraints

Consider a double integrator with the discretized model

$$A = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix}, \ B = \begin{pmatrix} T_s^2 \\ T_s \end{pmatrix} \tag{35}$$

using a sampling-interval $T_s = 0.05$. The control objective is defined by the cost function $l_{QR}(x, u) = x_1^2 + u^2$ and the constraints $-0.5 \leq x_2 \leq 0.5$ and $-1 \leq u \leq 1$. Figure 1 shows a simulation when the initial state is $x(0) = (-2, 0)^T$. Observe that initially the input constraint $u = 1$ is active. After $t \approx 0.5$, the state constraint $x_2 \leq 0.5$ is active, until $t \approx 2.85$ when the controller switches strategy once more, since it appears to be no longer optimal to stay on the constraint $x_2 = 0.5$. After this point the unconstrained LQ controller is used and the state is controlled to the origin. The switching strategy chosen by the controller is intuitive if x_1 is interpreted as position,

x_2 as speed and u as acceleration: In order to reduce the position error the speed is first increased at a maximum rate (given by the input constraint). When the maximum speed allowed is reached, this speed is kept until the position error becomes so small that the speed must be reduced to stabilize the position at the setpoint. In this example we have chosen the smallest possible horizon, namely $N = S = 1$ since this is advantageous for computational reasons. The region of feasibility is seen to be $X^F = \{x \in R^2 \mid |x_2| \leq 0.55\}$ since the input constraints restricts x_2 to be changed by at most 0.05 units within one sample. Hence, the admissible region $|x_2| \leq 0.5$ can be reached in one sample from X^F. In order to efficiently handle cases when $|x_2| > 0.55$, we define the input constraints as hard (non-relaxable) constraints, and the state constraints as soft (relaxable) constraints. Furthermore, we define all the elements of the weight vector $\omega_{1,\beta_{soft}}$ to be equal to one. Since the hard constraints are associated with the input only, $X^R = R^2$. The piecewise linear feedback control law is shown in Figure 2. □

Effectively, the active state constraints $x_2 = \pm 0.5$ are enforced by a sliding-mode like strategy in the example above. This is mainly due to the choice of a very small N, and will lead to poor robustness (Johansen *et al.* 2000*a*). However, the problem can be resolved by modifying the state constraints (9) such that they do not require the active state constraints to be fulfilled in a dead-beat manner (at the first possible sample), but rather attract the state asymptotically towards the active constraints. This is achieved by replacing every instant of the active state constraint equation $G_\beta x(t+\tau) = g_\beta$ by its asymptotic version

$$D_{\beta,\tau} G_\beta x(t+\tau) + ... + D_{\beta,1} x(t+1) + D_{\beta,0} G_\beta x(t) = g_\beta \tag{36}$$

where $D_{\beta,i}$ are diagonal matrices defined by pole placement such that $G_\beta x(t) \to g_\beta$ at a desired rate. In order take full advantage of this modification, it is convenient to introduce another modification, namely an ε-boundary layer near each active state constraint, similar to what is common in sliding mode control (Slotine 1984). Within this boundary layer, the controller is only allowed to switch to feedbacks that either makes the associated state constraints asymptotically active, or makes the state move away from the state constraint in the direction of the admissible region of the state space. Formally, this is achieved by adding the following constraint to the optimization problem (18)-(21)

$$G_{\beta(x)}(A^\tau x(t) + C_N \tilde{E}_\tau \tilde{u}_k^*(x(t))) \leq G_{\beta(x)} x(t), \qquad \text{if } \beta_k \subset \beta(x(t)) \tag{37}$$

for $1 \leq \tau \leq N$. The symbol $\beta(x)$ denotes the set of currently ε-active state constraints at x, i.e. $\beta(x) = \{l \in \{1, 2, ..., q\} \mid |(G_{l1}, ..., G_{ln})x - g_l| \leq \varepsilon_l\}$, where $\varepsilon_l > 0$ defines the boundary layer. Eq. (37) excludes non-attractive feedbacks that tend to move the state towards violation of active state constraints.

Double integrator example, cont'd

In order to reduce the gain near the active state constraints, a boundary layer of $\varepsilon_1 = \varepsilon_2 = \pm 0.15$ is defined around the constraints $x_2 = 0.5$ and $x_2 = -0.5$. Hence, when $0.65 \geq x_2 \geq 0.35$, the control strategy is allowed to switch and the controller takes the objective of attracting the state towards the surface $x_2 = 0.5$ while minimizing the LQ objective. The speed of the motion towards the surface $x_2 = 0.5$ is defined by $D_1 = 1/0.9$ and $D_2 = 1$. Comparing the piecewise linear controller surface of the modified controller (right part of Figure 2) with the original controller (left part of Figure 2), it is seen that the gain has indeed been reduced in an ε-boundary layer near the active constraints $x_2 = \pm 0.5$. $\square$

4 State space partitioning

The purpose of this section is to discuss how to solve the outer optimization problem (18)-(21), and in particular to derive an algorithm for computing a state space partitioning that can be used to decide which active constraint set sequence is optimal at a given state. The discrete minimizations in (18) and (24) can either be avoided completely in the real-time computations if the set of candidate optima is reduced to single elements within subsets of the state space, or at least reduced to a small subset of $\mathcal{C}$ within subsets of the state space. This can be exploited in the real-time implementation to reduce the processing capacity and memory requirements and also for computational analysis as considered in section 5.

4.1 Activity region

The activity region $X_k \subset X$ is defined as the subset of the state space where the active constraint set sequence with index k is active, i.e.

$$X_k = \{x \in X \mid k = k^*(x)\} \tag{38}$$

Together with the affine functions (34), the activity regions X_k, $k \in \mathcal{C}$ completely describes the PWL structure of the controller.

Double integrator example, activity regions

For the double integrator example, there are five constituent affine feedbacks with corresponding activity regions. Region/Feedback 0: unconstrained case

($k = 0$), Region/Feedback 1: input constraint $u = -1$ active ($k = 1$), Region/Feedback 2: input constraint $u = 1$ active ($k = 2$), Region/Feedback 3: state constraint $x_2 = -0.5$ active ($k = 3$), Region/Feedback 4: state constraint $x_2 = 0.5$ active ($k = 4$). The activity regions for the suboptimal constrained LQ controller with boundary layer are shown in Figure 3. We observe that in this case the regions can be characterized as unions of polyhedra. □

In order to explicitly characterize the activity regions, it is natural to treat the feasible and relaxed feasible regions X^F and X^R separately, since the choice of optimal active constraint set sequence is based on different criteria in these cases. Thus, we define the activity regions contained in X^F as follows:

$$X_k^f = \{x \in X_k^F \mid k \text{ is optimal w.r.t. (18)} - \text{(21) and (37)}\} \tag{39}$$

For $x \in X^R$, the controller objective changes to minimize the constraint violation and we define

$$X_k^r = \left\{x \in X_k^R \mid k \text{ is optimal w.r.t. (24)} - \text{(26)}\right\} \tag{40}$$

Hence, the activity region X_k where the feedback with index k is active is now $X_k = X_k^f \cup X_k^r$. A slightly more explicit characterization of X_k^f than (39) is

$$X_k^f = \left\{x \in X_k^F \mid \varphi_k(x) \leq \varphi_j(x), \quad \text{for all } j \in \mathcal{F}(x) \cap \mathcal{A}(x)\right\} \tag{41}$$

where $\mathcal{F}(x) \subset \mathcal{C}$ is a set containing the active constraint set sequences that are feasible at x

$$\mathcal{F}(x) = \left\{k \in \mathcal{C} \mid (x) \in X_k^F\right\} \tag{42}$$

and $\mathcal{A}(x) \subset \mathcal{C}$ is a set containing the indices to the active constraint set sequences that are attractive or not currently active at x, cf. (37):

$$\mathcal{A}(x) = \left\{k \in \mathcal{C} \mid G_{\beta(x)}(A^\tau x + C_N \tilde{E}_\tau \tilde{u}_k^*(x)) \leq G_{\beta(x)} x \quad \text{or} \quad \beta_k \not\subset \beta(x)\right\} \tag{43}$$

Furthermore, it follows that

$$X^F = \bigcup_{k \in \mathcal{C}} X_k^f \;=\; \bigcup_{k \in \mathcal{C}} X_k^F \tag{44}$$

Likewise, a slightly more explicit characterization of X_k^r than (40) is given by

$$X_k^r = \left\{x \in X_k^R \mid \nu_k(x) \leq \nu_j(x), \text{ for all } j \in \mathcal{R}(x)\right\} \tag{45}$$

where $\mathcal{R}(x) \subset \mathcal{C}$ is defined as the set of active constraint set sequences that are feasible with respect to the non-relaxable constraints, but not feasible with respect to the relaxable constraints, at x:

$$\mathcal{R}(x) = \left\{ k \in \mathcal{C} \mid x \in X_k^R \right\} \tag{46}$$

and we also have

$$X^R = \bigcup_{k \in \mathcal{C}} X_k^r = \bigcup_{k \in \mathcal{C}} X_k^R \tag{47}$$

which is the set of states where there exists an active constraint set sequence that is feasible and optimal with respect to the non-relaxable constraints but not with respect to the relaxable constraints.

4.2 Outer Approximations to the Activity Regions

Since X_k^F and X_k^R are polyhedral, it is clear that X^F, X^R and $X = X^F \cup X^R$ are unions of polyhedra. However, because the optimality conditions in (41) are characterized by *quadratic* functions, the set $X_k \subset X$ may not be characterized only by the hyper-planes defined by feasibility, but possibly also by other hyper-planes or (convex or non-convex) quadratic surfaces due to the optimality conditions. Thus, X_k may in general not be a union of polyhedra and therefore difficult to characterize exactly in a more explicit manner than (41) and (45). Still, several explicit outer approximations of X_k can be computed in terms of sets that contain X_k. Here we develop an outer approximation $\overline{X}_k \supset X_k$ where $\overline{X}_k$ is a union of polyhedra. As the basic polyhedral building blocks in this characterization we consider the hyperplane partition $\mathcal{P}_X^{HP} = \{\mathcal{X}_l^{HP} \mid l \in \{1, 2, ..., N_P\}\}$ generated by all the hyper-planes involved in the characterization of X_k^F, X_k^R and X_k^A, for $k \in \mathcal{C}$:

$$HE_\tau K_{k,2} x = h - HE_\tau (K_{k,1}^g g + K_{1,k}^h h) \tag{48}$$

$$G(A^\tau + C_N \tilde{E}_\tau K_{k,2}) x = g - GC_N \tilde{E}_\tau (K_{k,1}^g g + K_{k,1}^h h) \tag{49}$$

$$G(A^\tau + C_N \tilde{E}_\tau K_{2,k} - I) x = GC_N \tilde{E}_\tau (K_{1,k}^h h + K_{1,k}^g g) \tag{50}$$

for $1 \leq \tau \leq N$, with obvious interpretation when h or g are zero-dimensional. Let (48)-(50) be written in compact notation $Yx = y$. The set of half-spaces $\mathcal{Y}_i^+ = \{x \in R^n \mid Y_i x \geq y_i\}$ and $\mathcal{Y}_i^- = \{z \in R^n \mid Y_i x < y_i\}$ now defines the hyperplane partition $\mathcal{P}_X^{HP}$ of X as the set of all possible non-empty intersections of half-spaces: $\mathcal{X}_l^{HP} = \mathcal{Y}_1^* \cap ... \cap \mathcal{Y}_{N_z}^*$ where $*$ symbolizes any combinations of $+/-$. Note that this hyperplane partition will contain unnecessarily many elements in many cases and is introduced here in order to develop a theoretical understanding.

Lemma 2 *The hyperplane partition $\mathcal{P}_X^{HP}$ has the following properties:*

(1) Each constituent region of the partition is uniquely associated with either the feasible region X^F or the relaxed feasible region X^R, i.e. $\mathcal{X}_l^{HP} \cap X^F = \emptyset$ and $\mathcal{X}_l^{HP} \cap X^R = \mathcal{X}_l^{HP}$, or vice versa $\mathcal{X}_l^{HP} \cap X^F = \mathcal{X}_l^{HP}$ and $\mathcal{X}_l^{HP} \cap X^R = \emptyset$, for all $l = 1, 2, ..., N_P$.

(2) For all $l \in \{1, 2, ..., N_p\}$ and $x \in \mathcal{X}_l^{HP}$ the sets $\mathcal{F}(x)$, $\mathcal{R}(x)$, $\mathcal{A}(x)$ and $\beta(x)$ are invariant in the sense that each of them contain the same elements for all $x \in \mathcal{X}_l^{HP}$.

Proof. Follows from the fact that the hyperplane partition $\mathcal{P}_X^{HP}$ of X is generated by all hyper-planes involved in the characterizations of $\mathcal{F}(x)$, $\mathcal{R}(x)$, $\mathcal{A}(x)$ and $\beta(x)$. □

From the first part of Lemma 2 it is evident that each $\mathcal{X}_l^{HP} \in \mathcal{P}_X^{HP}$ is fully contained in either X^R or X^F. Thus, we define disjoint index sets

$$\mathcal{L}^F = \left\{ l \in \{1, 2, ..., N_P\} \mid \mathcal{X}_l^{HP} \cap X^F \neq \emptyset \right\} \tag{51}$$

$$\mathcal{L}^R = \left\{ l \in \{1, 2, ..., N_P\} \mid \mathcal{X}_l^{HP} \cap X^R \neq \emptyset \right\} \tag{52}$$

Assume $l \in \mathcal{L}^F$, i.e. $\mathcal{X}_l^{HP} \subset X^F$. One may now define a set $\mathcal{F}_l^f \subset \mathcal{C}$ of feasible active constraint set sequences in the region $\mathcal{X}_l^{HP}$:

$$\mathcal{F}_l^f = \{ k \in \mathcal{C} \mid \mathcal{X}_l^{HP} \cap X_k^F \neq \emptyset \} \tag{53}$$

Hence, for any $x \in X^F$ there exists a unique $l(x) \in \mathcal{L}^F$ such that $x \in X_{l(x)}^F$ and at least one of the feasible active constraint set sequences in $\mathcal{F}_l^f$ is optimal for all $x \in \mathcal{X}_l^{HP}$. We continue by characterizing the subset of $\mathcal{F}_l^f$ that is optimal for some $x \in \mathcal{X}_l^F$, aiming towards a definition of $\overline{X}_k^f \supset X_k^f$.

Lemma 3 *Let $l \in \mathcal{L}^F$ and $j, k \in \mathcal{F}_l^f$ be arbitrary. Suppose the active constraint set sequences $((\alpha_1^k, \beta_1^k), (\alpha_2^k, \beta_2^k), ..., (\alpha_{N_S}^k, \beta_{N_S}^k))$ and $((\alpha_1^j, \beta_1^j), (\alpha_2^j, \beta_2^j), ..., (\alpha_{N_S}^j, \beta_{N_S}^j))$ are different. If $\alpha_i^k \subset \alpha_i^j$ and $\beta_i^k \subset \beta_i^j$, for all $i = 1, 2, ..., N_S$, then the active constraint set sequence with index j is suboptimal for all $x \in \mathcal{X}_l^{HP}$.*

Proof. Because the active constraint set sequence with index k is a subset of the active constraint set sequence with index j and both are feasible, it follows immediately that $\varphi_k(x) \leq \varphi_j(x)$ for all $x \in \mathcal{X}_l^{HP}$ since adding a constraint to some constraint set sequence will not reduce the cost. □

Lemma 4 *Let $l \in \mathcal{L}^F$ and $k \in \mathcal{F}_l^f$ be arbitrary, and define*

$$\gamma_{jk} = \max_{x \in \mathcal{X}_l^{HP}} (\varphi_k(x) - \varphi_j(x)) \tag{54}$$

$$\kappa_{jk} = \min_{x \in \mathcal{X}_l^{HP}} (\varphi_k(x) - \varphi_j(x)) \tag{55}$$

If $\gamma_{jk} \leq 0$ for all $j \in \mathcal{F}_l^f$, then the active constraint set sequence with index k is optimal for all $x \in \mathcal{X}_l^{HP}$. If $\kappa_{jk} \geq 0$ for all $j \in \mathcal{F}_l^f$, then the active constraint set sequence with index k is suboptimal for all $x \in \mathcal{X}_l^{HP}$.

Proof. Since $\gamma_{jk} \leq 0$ it follows that for all $x \in \mathcal{X}_l^{HP}$ and $j \in \mathcal{F}_l^f$, $\varphi_k(x) \leq \varphi_j(x)$. Note that due to Lemma 2, $\mathcal{F}_l^f = \mathcal{F}(x)$ for all $x \in \mathcal{X}_l^{HP}$, and the first part follows because $\mathcal{F}_l^f$ contains all feasible active constraint set sequences in $\mathcal{C}$. The second part of the lemma is analogous. □

Both (54) and (55) are quadratic programs, for a fixed $l \in \mathcal{L}^F$ and fixed active constraint set sequences $k, j \in \mathcal{F}_l^f$, since $\mathcal{X}_l^{HP}$ is polyhedral and φ_k and φ_j are quadratic. Using the optimality characterizations in Lemmas 3 and 4, one will typically be able to exclude a large set of candidate active constraint set sequences from the set of feasible active constraint set sequences $\mathcal{F}_l^f$ in the region $\mathcal{X}_l^{HP}$. We define $\mathcal{O}_l^f \subset \mathcal{F}_l^F$ as the indices of those active constraint set sequences that are consistent with the optimality conditions in Lemmas 3 and 4 in $\mathcal{X}_l^{HP}$:

$$\mathcal{O}_l^f = \left\{ k \in \mathcal{F}_l^f \mid k \text{ is optimal w.r.t. (18)-(21), (37) for some } x \in \mathcal{X}_l^{HP} \right\} \tag{56}$$

We define the outer approximation to the activity region X_k^f as follows:

$$\overline{X}_k^f = \bigcup_{l \in \mathcal{L}^F} \mathcal{X}_l^{HP} \tag{57}$$

Next, assume $l \in \mathcal{L}^R$, i.e. $\mathcal{X}_l^{HP} \subset X^R$. One may now define a set $\mathcal{F}_l^r \subset \mathcal{C}$ of relaxed feasible active constraint set sequences in the region $\mathcal{X}_l^{HP}$:

$$\mathcal{F}_l^r = \left\{ k \in \mathcal{C} \mid \mathcal{X}_l^{HP} \cap X_k^R \neq \emptyset \right\} \tag{58}$$

Unlike the characterization of X_k^f, we now have the following result:

Lemma 5 *For all $k \in \mathcal{C}$, the set X_k^r is a union of polyhedra.*

Proof. Let $k \in \mathcal{C}$ be arbitrary. The region of relaxed feasibility X_k^R is polyhedral, cf. (28). Since the function $\nu_k(x) - \nu_j(x)$ is piecewise linear in x, the sets

$\{x \in R^n \mid \nu_k(x) \leq \nu_j(x)\}$ that appear in the optimality condition in (45) are characterized using hyper-planes. Since all geometric objects characterizing X_k^r are hyper-planes, it is a union of polyhedral sets. □

According to Lemma 5 it is possible to obtain an exact and explicit characterization of X_k^r. However, for computational reasons it may be convenient with an outer approximation $\overline{X}_k^r \supset X_k^r$ in some cases. The following optimality lemma is useful in that respect:

Lemma 6 *Let $l \in \mathcal{L}^R$ and $k \in \mathcal{F}_l^r$ be arbitrary, and define*

$$\rho_{jk} = \max_{x \in \mathcal{X}_l^{HP}} (\nu_k(x) - \nu_j(x)) \tag{59}$$

$$\sigma_{jk} = \min_{(x) \in \mathcal{X}_l^{HP}} (\nu_k(x) - \nu_j(x)) \tag{60}$$

If $\rho_{jk} \leq 0$ for all $j \in \mathcal{F}_l^r$, then the active constraint set sequence with index k is optimal for all $x \in \mathcal{X}_l^{HP}$. If $\sigma_{jk} \geq 0$ for all $j \in \mathcal{F}_l^R$, then the active constraint set sequence with index k is suboptimal for all $x \in \mathcal{X}_l^{HP}$.

Proof. Analogous to Lemma 4. □

Note that (59) and (60) are piecewise linear programs. Using the optimality characterizations in Lemma 6, one will typically be able to exclude a large set of candidate active constraint set sequences from the set of feasible active constraint set sequences $\mathcal{F}_l^r$ in the region $\mathcal{X}_l^{HP}$. We define $\mathcal{O}_l^r \subset \mathcal{F}_l^R$ as the indices of those active constraint set sequences that are consistent with the optimality conditions in Lemma 6 in $\mathcal{X}_l^{HP}$:

$$\mathcal{O}_l^r = \left\{ k \in \mathcal{F}_l^r \mid k \text{ is optimal w.r.t. (24)-(26) for some } x \in \mathcal{X}_l^{HP} \right\} \tag{61}$$

Finally, we define the outer approximation to the activity region X_k^r as follows:

$$\overline{X}_k^r = \bigcup_{l \in \mathcal{L}^R} \mathcal{X}_l^{HP} \tag{62}$$

We are now in position to define $\overline{X}_k = \overline{X}_k^f \cup \overline{X}_k^r$ and

$$\mathcal{F}_l = \begin{cases} \mathcal{F}_l^f, & l \in \mathcal{L}^F \\ \mathcal{F}_l^r, & l \in \mathcal{L}^R \end{cases} \tag{63}$$

$$\mathcal{O}_l = \begin{cases} \mathcal{O}_l^f, & l \in \mathcal{L}^F \\ \mathcal{O}_l^r, & l \in \mathcal{L}^R \end{cases} \tag{64}$$

The above sets can in principle be computed directly by first determining the hyperplane partition $\mathcal{P}_X^{HP}$ and then using Lemmas 3-6 to compute the candidate optimal affine feedbacks within each region of the partition. However, this procedure may be too computationally intensive for large problems, and an alternative algorithm is required.

Algorithm 1 (Partitioning algorithm)

(1) Let $\mathcal{E} := \emptyset$, and $\mathcal{U} := \{X\}$.
(2) If $\mathcal{U} = \emptyset$, the partition generated by this algorithm is $\mathcal{P} = \mathcal{E}$ and the algorithm terminates.
(3) Let $\mathcal{X}_0 \in \mathcal{U}$ be arbitrary.
(4) Let $\mathcal{O}_0$ contain the candidate optimal active constraint set sequences in $\mathcal{X}_0$, computed according to Lemmas 3-6.
(5) If $\mathcal{O}_0$ contains a sufficiently small number of elements, add $\mathcal{X}_0$ to the set of explored subsets $\mathcal{E}$ and remove $\mathcal{X}_0$ from the set of unexplored subsets $\mathcal{U}$. Go to step 2.
(6) Select a hyperplane $Y_i x = y_i$ from $Yx = y$ and split $\mathcal{X}_0$ into non-empty $\mathcal{X}_0^+ = \mathcal{X}_0 \cap \mathcal{Y}_i^+$ and $\mathcal{X}_0^- = \mathcal{X}_0 \cap \mathcal{Y}_i^-$. If this is not possible for any hyperplane from $Yx = y$, add $\mathcal{X}_0$ to the set of explored subsets $\mathcal{E}$ and remove $\mathcal{X}_0$ from the set of unexplored subsets $\mathcal{U}$. Go to step 2.
(7) Add $\mathcal{X}_0^+$ and $\mathcal{X}_0^-$ to $\mathcal{U}$ and remove $\mathcal{X}_0$ from $\mathcal{U}$. Go to step 2.

□

The set $\mathcal{E}$ contains the set of explored subsets of X, while the set $\mathcal{U}$ contains the set of explored subsets of X. The algorithm will explore the candidate optimal active constraint sets associated with each element of $\mathcal{E}$ sequentially. The regions of X will be split using the hyper-planes from $Yx = y$ and explored individually until either a sufficiently small number of candidate optimal active constraint set remains in each region, or the region can not be split any further using hyper-planes from $Yx = y$. The following theorem summarizes the properties of the result of Algorithm 1.

Theorem 3 *(Partitioning) Algorithm 1 terminates with a partition $\mathcal{P}_X$ and sets $\mathcal{O}_l$, $\overline{X}_k$ that satisfies $k^*(x) \in \mathcal{O}_l$, for all $x \in X$, and $X_k \subset \overline{X}_k$, and $\overline{X}_k$ is a union of polyhedra.* □

In order to reduce the computational complexity of Algorithm 1 one should implement heuristics in step 6 in order to select a "promising" hyperplane for splitting the region $\mathcal{X}_0$ such that unnecessary splitting is avoided. Note that the partition $\mathcal{P}_X$ generated by Algorithm 1 may be unnecessarily fine since at each step it is not known a priori if one can reduce the number of elements in

$\mathcal{O}_0$ by further partitioning of $\mathcal{X}_0$. Hence, after the algorithm terminates, the number of constituent polyhedra in the partition of X can often be reduced considerably by aggregating pairs of neighboring polyhedra whenever their union remains polyhedral, see also (Bemporad *et al.* 2001).

Double integrator example, cont'd

The partition computed using Algorithm 1 with a successive aggregation of neighboring regions is shown in Figure 4. We observe that the number of regions is 11, which is the smallest possible number of polyhedral regions capable of characterizing the activity sets for this problem. Also, we observe that within each region, there is a single candidate optimal active constraint set sequence. Hence, the PWL feedback law is explicitly characterized by this partition. Feedback 0 (unconstrained case) is associated with R1, R3 and R4 in this partition. Feedback 1 ($u = -1$) is associated with R5 and R11. Feedback 2 ($u = 1$) is associated with R7 and R10. Feedback 3 ($x_2 = -0.5$) is associated with R2 and R6, while feedback 4 ($x_2 = 0.5$) is associated with R8 and R9. □

So far the partitioning has been restricted to utilize only the hyper-planes derived from the linear feasibility (and attraction) constraints. Consequently, there need not always be a single candidate optimal constraint set sequence within each region of the partition. Indeed, if the remaining number of candidate optima in $\mathcal{O}_l$ is unacceptably large for some region $\mathcal{X}_l \in \mathcal{P}_X$, one still has the option to proceed by partitioning the polyhedral region $\mathcal{X}_l$ further, either utilizing the (possibly non-linear) surfaces derived from the optimality conditions or some approximating hyper-planes. An exception is in the case of the optimal constrained LQR (Bemporad *et al.* 1999). When there are no restrictions on the allowed switching between active constraint set sequences on the horizon, the exact partition is a union of polyhedra.

5 Optimality, complexity and real-time implementation

5.1 Upper and lower bounds on cost function

Define the closed loop performance of the suboptimal constrained LQR as follows:

$$\hat{V}(x(0)) = \sum_{t=0}^{\infty} \left(x^T(t)Qx(t) + (u^*(t))^T Ru^*(t) \right) \tag{65}$$

For example 1, upper and lower bound on cost $V(x(0))$ are illustrated in Figure 5. These bounds are computed by solving LMIs with a continuous piecewise quadratic parameterization of the functions as described in (Johansson and Rantzer 1998, Rantzer and Johansson 2000), see (Johansen *et al.* 2000*a*) for details. Note that a continuous-time approximation is utilized due to restrictions in the available software implementation (Hedlund and Johansson 1999), and that the bounds have no direct meaning for $x \notin X^F$, except that the upper bound defines a Lyapunov function (under a detectability assumption).

5.2 Complexity reduction by sub-optimality

It was claimed initially that we expect that the restrictions introduced on the allowed active constraint set sequence switching times will reduce the computational complexity of the controller, i.e. lead to a partition of the state space with less regions. We illustrate this by an example.

Example 2, Double integrator (Bemporad *et al.* 1999)

Consider the double integrator

$$A = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} T_s^2 \\ T_s \end{pmatrix} \tag{66}$$

with sampling-interval $T_s = 0.05$. The control objective is defined by the cost function $l_{QR}(x, u) = x_1^2 + 0.1u^2$ and the constraints $-1 \leq u \leq 1$. We consider two cases

(1) $N = 8$, with no restrictions on the number of active constraint set switches on the horizon, same as (Bemporad *et al.* 1999).
(2) $N = 8,\ S = 2,\ N_2 = 3$, i.e. only one active constraint set switch allowed on the horizon

The second case leads to the following nine candidate active constraint set sequences that enumerates the set $\mathcal{C}$:

First 3 samples	Last 5 samples
$u = Kx$	$u = Kx$
$u = Kx$	$u = -1$
$u = Kx$	$u = 1$
$u = -1$	$u = Kx$
$u = -1$	$u = -1$
$u = -1$	$u = 1$
$u = 1$	$u = Kx$
$u = 1$	$u = -1$
$u = 1$	$u = 1$

The suboptimal strategy gives a reduction from 93 to 33 regions, cf. Figure 6, which allows a significant reduction of the real-time processing and memory requirements. From Figure 7 we observe that the differences in the closed loop trajectories for $x(0) = (-3, 3)^T$ are not very significant. □

5.3 Real-time Implementation

The suboptimal constrained LQR is a PWL function of the state. However, efficient evaluation of this PWL function in the real-time control system requires that one is able to efficiently compute in real time which affine feedback to associate with each vector x. The affine state feedbacks are computed offline and stored in real-time computer memory. Whether it is desirable to also compute offline an explicit characterization of the subsets of X where each affine feedback is active depends on several factors: Acceptable offline processing time, available real-time computer memory and real-time computer processing capacity. There exist at least two real-time implementation strategies that can be employed in order to address the above mentioned tradeoffs:

(1) The discrete optimization problems (18)-(21) and (24)-(26) are solved in real time. Discrete search techniques such as branch-and-bound and A^* can be applied for this purpose (Korf 1990).
(2) A partitioning of X such that within each constituent region of the partition there are at most a given small number of affine feedbacks that may be optimal. A search among the small number of remaining candidates (if more than one) is then carried out in real time.

Example 2, continued

By early termination of the partitioning algorithm one can achieve for example the partitions shown in Figure 6. In the first case there are 9 regions, each with a list of up to 3 affine feedbacks that are optimal at various states within each region, see Table 1. In the second case there are 3 regions, with a list of up to 5 affine feedbacks that are optimal at various states in each region, see also Table 1. Hence, one can reduce the complexity of the partition by comparing the values of a user-specified number of quadratic functions and linear constraints in real time. Obviously, this gives the user additional flexibility for the real-time implementation. In a sense, one has a method for partially solving the real-time quadratic program offline. □

Example 3, laboratory model helicopter

A laboratory model helicopter (Quanser 3-DOF Helicopter) with two DC-motor driven rotors is sampled with $T = 0.01s$, and the following state-space representation is obtained

$$A = \begin{pmatrix} 1 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0.01 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 0 & 0 \\ 0.0001 & -0.0001 \\ 0.0019 & 0.0019 \\ 0.0132 & -0.0132 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

The states of the system are x_1 - elevation, x_2 - pitch angle, x_3 - elevation rate, x_4 - pitch angle rate, x_5 - integral of elevation error, and x_6 - integral of pitch angle error. The inputs are u_1 and u_2, the front and rear rotor voltages. Assume the system is to be regulated to some setpoint with the following constraints on the inputs, pitch and elevation rates $-1 \leq u_1 \leq 3$, $-1 \leq u_2 \leq 3$, $-0.25 \leq x_3 \leq 0.25$, and $-0.6 \leq x_4 \leq 0.6$. The LQ cost function is given by $Q = \text{diag}(100, 20, 40, 8, 1, 0.5)$ and $R = \text{diag}(1, 1)$. With $N = 1$ this leads to 33 active constraint sets. Comparing their quadratic cost function and evaluating the linear constraints requires in the worst 320 microseconds on a 450 MHz Pentium II with our implementation. If necessary, this can be reduced by state space partitioning using Algorithm 1. The experimental results in Figure 8 compares the performance along the elevation axis with unconstrained LQR. □

Another experimental case study utilizing this approach in an automotive application is reported in (Petersen *et al.* 2001).

Due to the exponential growth of the number of candidate active constraint set sequences as the number of states, horizon and constraints increases, the approach is restricted to problems of low and moderate complexity. As the problem complexity increases, the use of prior knowledge and simulation are the keys to restricting the number of candidate active constraint set sequences and the (offline and online) computational complexity.

6 Conclusions

A suboptimal strategy for explicit offline design of LQ controllers subject to state and input constraints is derived. It is demonstrated that allowing suboptimality in terms of restrictions on the number of allowed active constraint set changes on the horizon leads to significant reduction in the complexity of the state space partitioning. The method gives the user flexibility to address the tradeoff between real-time computer memory and processing capacity. The approach provides a practical framework for design, analysis and efficient real-time implementation of LQ controllers that are explicitly designed to satisfy constraints on the states and inputs.

References

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

Bemporad, A., M. Morari, V. Dua and E. N. Pistikopoulos (1999). The explicit linear quadratic regulator for constrained systems. Technical Report AUT99-16. Automatic Control Laboratory, ETH Zurich, Switzerland.

Bemporad, A., M. Morari, V. Dua and E. N. Pistikopoulos (2000). The explicit solution of model predictive control via multiparametric quadratic programming. In: *Proc. American Control Conference, Chicago*. pp. 872–876.

Chisci, L. (1999). Fast algorithm for constrained infinite horizon LQ problem. *Int. J. Control* **72**, 1020–1026.

Chmielewski, D. and V. Manousiouthakis (1996). On constrained infinite-time linear quadratic optimal control. *Systems and Control Letters* **29**, 121–129.

Hedlund, S. and M. Johansson (1999). Pwltool - A MATLAB toolbox for analysis of piecewise linear systems. Technical Report LUTFD2/TFRT-7582-SE. Automatic Control Department, Lund Institute of Technology, Sweden.

Johansen, T. A., I. Petersen and O. Slupphaug (2000*a*). Explicit suboptimal linear quadratic regulation with input and state constraints. Technical Report STF72-A00303. SINTEF.

Johansen, T. A., I. Petersen and O. Slupphaug (2000*b*). On explicit suboptimal LQR with state and input constraints. In: *Proc. IEEE Conf. Decision and Control, Sydney.* pp. TuM05–6.

Johansson, M. and A. Rantzer (1998). Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Automatic Control* **43**, 555–559.

Keerthi, S. S. and E. G. Gilbert (1988). Optimal infinite horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations. *J. Optimization Theory and Applications* **57**, 265–293.

Korf, R. E. (1990). Real-time heuristic search. *Artificial Intelligence* pp. 189–211.

Petersen, I., T. A. Johansen, J. Kalkkuhl and J. Lüdemann (2001). Wheel slip control in ABS brakes using gain scheduled constrained LQR. In: *Proc. European Control Conference, Porto, Portugal.* pp. 606–611.

Rantzer, A. and M. Johansson (2000). Piecewise linear quadratic optimal control. *IEEE Trans. Automatic Control* **45**, 629–637.

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

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

Seron, M., J. A. De Dona and G. C. Goodwin (2000). Global analytical model predictive control with input constraints. In: *Proc. IEEE Conf. Decision and Control, Sydney.* pp. TuA05–2.

Slotine, J.-J. E. (1984). Sliding controller design for non-linear systems. *Int. J. Control* **40**, 421–434.

Sznaier, M. and M. J. Damborg (1987). Suboptimal control of linear systems with state and control inequality constraints. In: *Proc. 26th Conf. Decision and Control.* pp. 761–762.

Sznaier, M. and M. J. Damborg (1990). Heuristically enhanced feedback control of constrained discrete-time linear systems. *Automatica* **26**, 521–532.

Wredenhagen, G. F. and P. R. Belanger (1994). Piecewise-linear LQ control for systems with input constraints. *Automatica* **30**, 403–416.

Zheng, A. and M. Morari (1995). Stability of model predictive control with mixed constraints. *IEEE Trans. Automatic Control* **60**, 1818–1823.

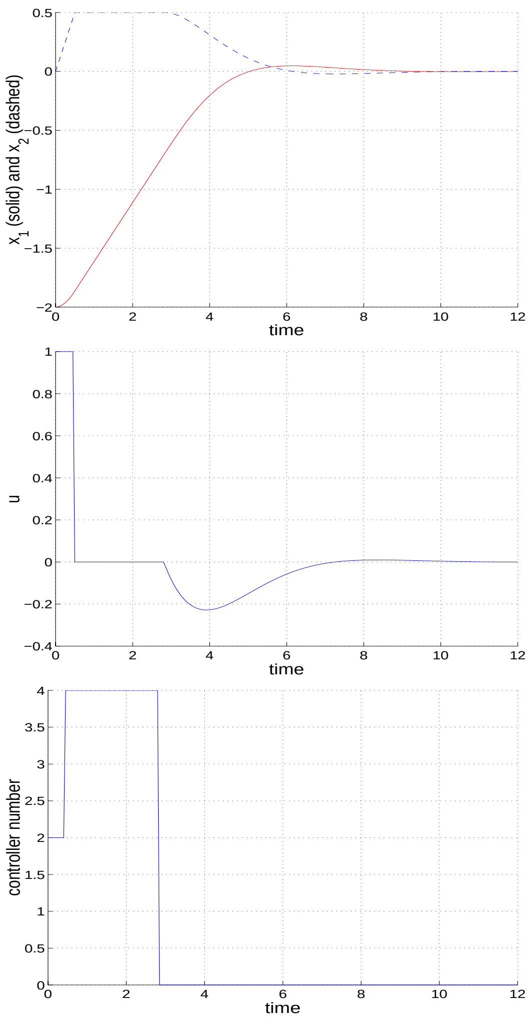

Fig. 1. Constrained control of a double integrator from initial state $x(0) = (-2, 0)^T$.

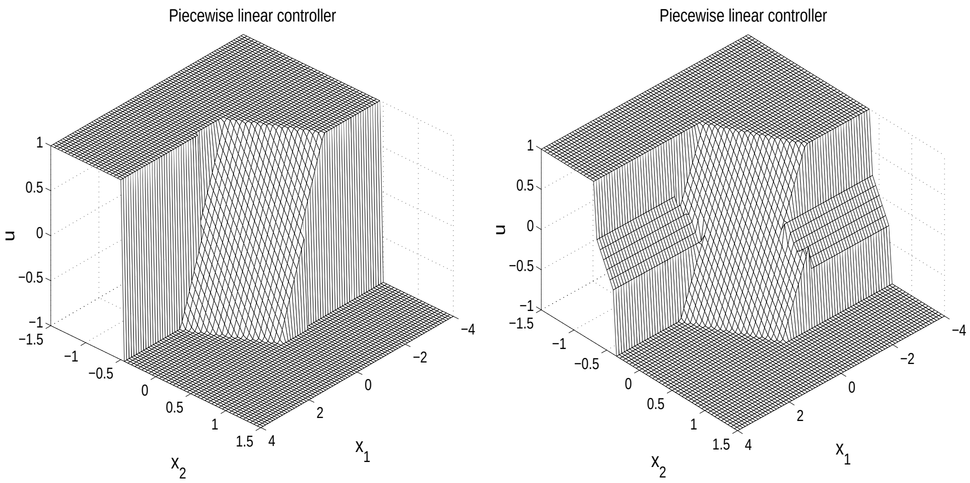

Fig. 2. PWL constrained LQ feedback controller for the double integrator (left) and with boundary layer around active state constraints (right).

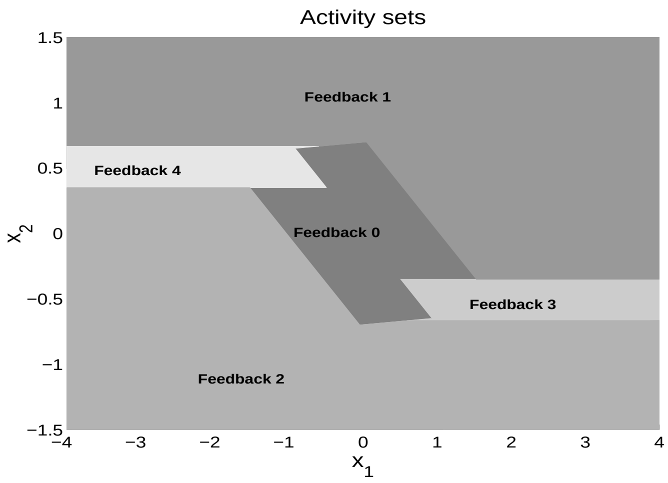

Fig. 3. Activity regions for the five constituent affine feedbacks in the constrained LQR for the double integrator with boundary layers.

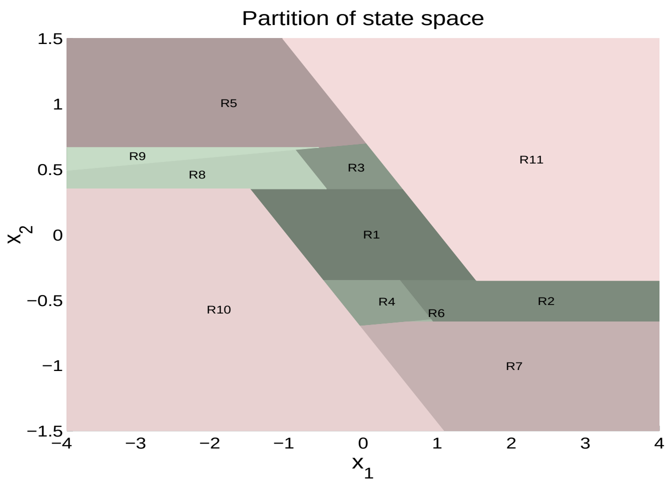

Fig. 4. Partition for the constrained LQR for the double integrator with boundary layers.

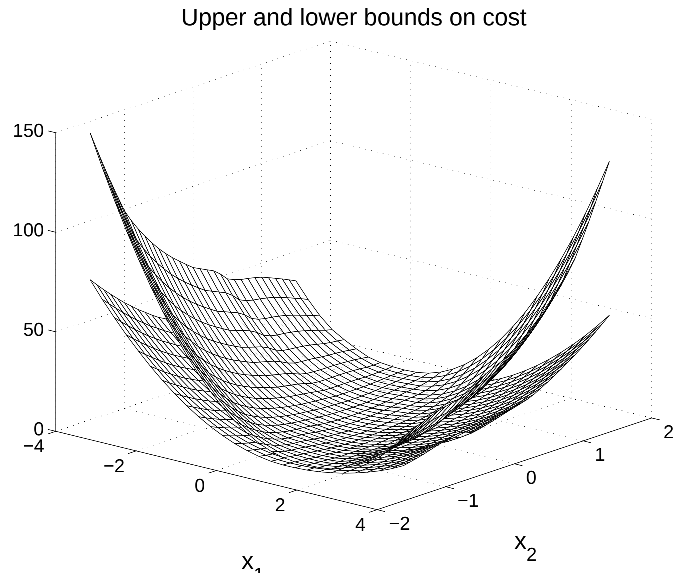

Fig. 5. Upper and lower bounds on the cost for the constrained LQR for the double integrator with boundary layers.

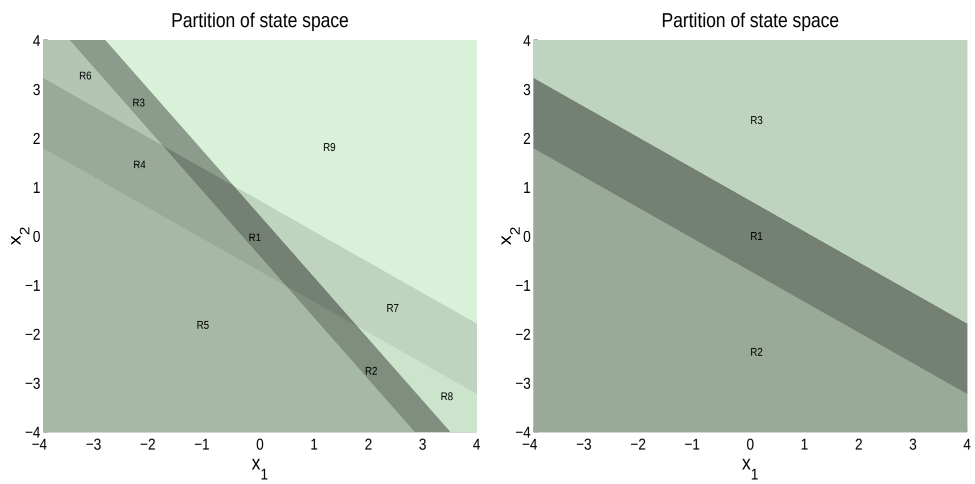

Fig. 6. Double integrator with input constraints and $N = 8$. Left: Simple partition where the maximum number of candidate state feedbacks in each region is 3. Right: Simple partition where the maximum number of candidate state feedbacks in each region is 5.

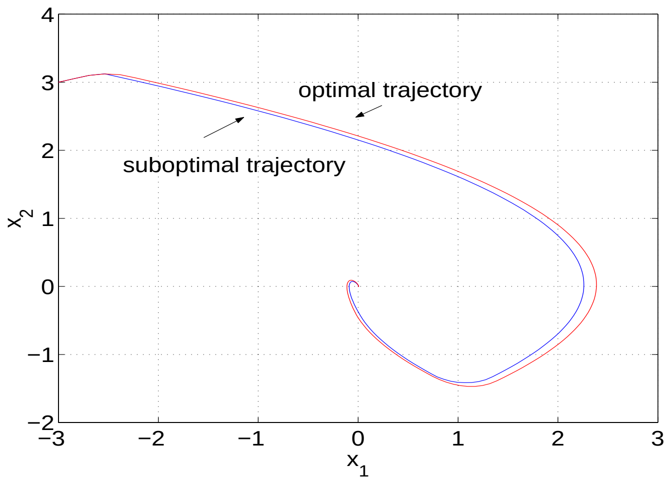

Fig. 7. Example of trajectories with and without active constraint set change restrictions.

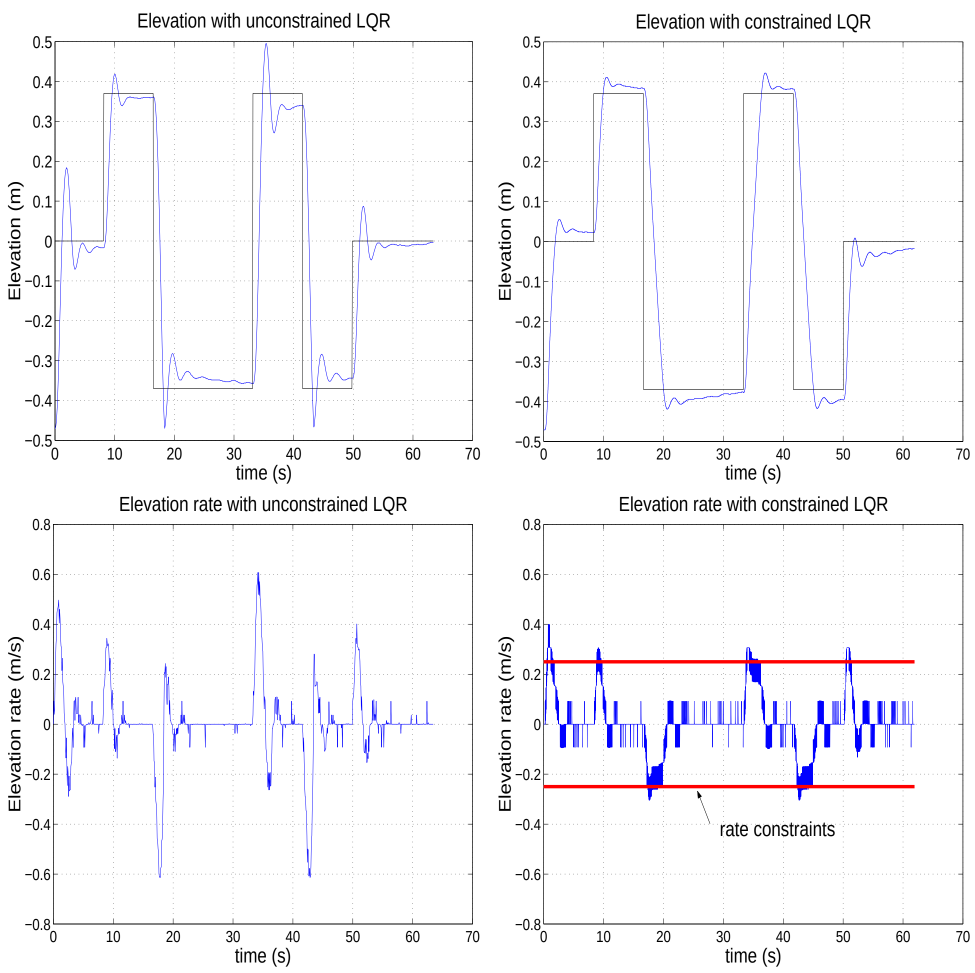

Fig. 8. Experimental results with 3-DOF laboratory model helicopter.

	Region	Candidate optimal feedbacks
9 regions	R1	$\{1\}$
	R2	$\{3,6,9\}$
	R3	$\{2,5,8\}$
	R4	$\{7,8,9\}$
	R5	$\{9\}$
	R6	$\{2,8,9\}$
	R7	$\{4,5,6\}$
	R8	$\{3,5,6\}$
	R9	$\{5\}$
3 regions	R1	$\{1,4,5,7,9\}$
	R2	$\{3,5,6,9\}$
	R3	$\{2,5,8,9\}$

Table 1
List of candidate optimal feedbacks for the simplified partitions of example 2.