On Explicit Suboptimal LQR with State and Input Constraints

Tor A. Johansen*,**[1], Idar Petersen**, and Olav Slupphaug*

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

** SINTEF Electronics and Cybernetics, N-7465 Trondheim, 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. 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. It is shown that the resulting feedback controller is piecewise linear, and the piecewise linear structure is exploited for computational analysis of stability and performance as well as efficient real-time implementation.

1 Introduction

Consider the continuous-time linear time-invariant system

$$\dot{x} = A_c x + B_c u \quad (1)$$

and its sampled discrete-time version

$$x(t+1) = Ax(t) + Bu(t) \quad (2)$$

where $x \in R^n$, and $u \in R^r$. The discrete-time version will be used for controller design, while the continuous-time version is used for analysis. The optimal constrained LQ feedback controller minimizes the infinite horizon quadratic cost

$$J(u(t), u(t+1), ...; x(t)) = \sum_{\tau=t}^{\infty} l_{QR}(x(\tau), u(\tau)) \quad (3)$$

$$l_{QR}(x, u) = x^T Q x + u^T R u \quad (4)$$

subject to the linear constraints

$$Gx(\tau+1) \leq g \quad (5)$$

$$Hu(\tau) \leq h \quad (6)$$

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 $g, h > 0$ to ensure that the origin is an interior point in the admissible region. The *optimal cost function* is defined as

$$V(x(t)) = \min_{u(t), u(t+1), ...} J(u(t), u(t+1), ...; x(t)) \quad (7)$$

where the minimization is subject to the dynamics of the system (2), and the constraints (5)-(6) 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 $V(x(t))$. The Hamilton-Jacobi-Bellman (HJB) equation characterizes the optimal cost function and optimal control action for the problem:

$$0 = \min_{\substack{u(\tau) \in R^r, Gx(\tau+1) \leq g, Hu(\tau) \leq h \\ \tau \in \{t, t+1, t+2, ..., t+N-1\}}} \Bigg(V(x(t+N)) - V(x(t)) + \sum_{\tau=t}^{t+N-1} l_{QR}(x(\tau), u(\tau)) \Bigg) \quad (8)$$

where $N \geq 1$ is some horizon, and $V(0) = 0$. Under the assumptions of feasibility, non-explicit optimal solutions to the HJB (8) are known for the case when N is so large that there are no active or violated constraints beyond this horizon and the unconstrained LQ solution is optimal beyond the horizon [1, 2, 3]. These solutions are based on 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 Ricatti equation associated with the unconstrained LQR. An algorithm to compute the sufficiently large N can be found in [2]. In all the above mentioned approaches the controller is implicit and computation of the control input $u(t)$ relies on real-time quadratic programming at each sample. This imposes severe limitations on the achievable sample rate that may discourage the application of this approach in many cases. Related approaches include [4, 5, 6].

Recently, Bemporad et al. [7, 8] derived an *explicit* optimal solution to the constrained LQR problem using offline multi-parametric quadratic programming. The constrained LQR problem is viewed as a quadratic program parameterized by the state x, and a multi-parametric quadratic programming algorithm essentially finds an explicit solution for all x within an arbitrary subset of the state space. Again, the optimal solution requires that N is sufficiently large. The resulting optimal controller was proved to be a piecewise linear function, see also [9], defined on a polyhedral partitioning of the state-space.

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 constraints imposed by the real-time application on both computer memory and processing capacity, a (possibly) suboptimal strategy is developed. Hence, one has a mechanism to trade performance

[1]Email: Tor.Arne.Johansen@itk.ntnu.no. This work was in part sponsored by the European Commission under the ESPRIT Long Term Research project 28104 $\mathcal{H}^2\mathcal{C}$.

for computational advantages. Moreover, the problem formulation is extended from the basic formulation (2)-(6) by introducing two additional specifications: First, we allow $g \in \mathcal{G}$ and $h \in \mathcal{H}$, where $\mathcal{G}$ and $\mathcal{H}$ are polyhedral sets. Second, if no feasible control input that prevents the state from moving into the non-admissible region on the prediction horizon can be computed, the LQ objective is subsumed by the more important objective of minimizing the constraint violations according to some criterion that is defined using weighting and prioritization among the violated state constraints. A full version of this paper is available [10].

2 Controller Design

2.1 Main ideas

Let $D^T = (d_1, d_2, ..., d_m)$ and $e^T = (e_1, e_2, ..., e_n)$. 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. 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. Note that the active constraint set may be empty, meaning that no constraints are active.

At each sample one may impose a number of equality constraints on the states and inputs that at most is 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 explicit LQR problem is now simply to evaluate all feasible active constraint set sequences on a sufficiently large horizon N in order to determine their regions of optimality. Here we suggest to use a smaller horizon N than required by the optimal solution and in addition to reduce the flexibility in the active constraint set sequence by allowing only a smaller number of changes in the active constraint set on the horizon N. Furthermore, one may require that the active constraint set changes are made at predetermined (fixed) samples. For example, one may subdivide the horizon $N = 10$ into $S = 3$ subintervals, and only allow the active constraint set to change at the beginning of each subinterval. Suppose the set of indices α is associated with the active input constraints in (6) at some sample, and the set of indices β is associated with the active state constraints (5) at the same sample. Then (α, β) is an active constraint set. Next, suppose we define allowed switching times as follows

$$0 = N_1 < N_2 < \cdots < N_S < N \tag{9}$$

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:

$$\left.\begin{array}{c} G_{\beta_1}\left(Ax(t) + C_N\tilde{E}_1\tilde{u}(t)\right) = g_{\beta_1} \\ G_{\beta_1}\left(A^2x(t) + C_N\tilde{E}_2\tilde{u}(t)\right) = g_{\beta_1} \\ \vdots \\ G_{\beta_1}\left(A^{N_2-1}x(t) + C_N\tilde{E}_{N_2-1}\tilde{u}(t)\right) = g_{\beta_1} \\ G_{\beta_2}\left(A^{N_2}x(t) + C_N\tilde{E}_{N_2}\tilde{u}(t)\right) = g_{\beta_2} \\ G_{\beta_2}\left(A^{N_2+1}x(t) + C_N\tilde{E}_{N_2+1}\tilde{u}(t)\right) = g_{\beta_2} \\ \vdots \\ G_{\beta_{N_S}}\left(A^{N}x(t) + C_N\tilde{E}_{N}\tilde{u}(t)\right) = g_{\beta_{N_S}} \end{array}\right\} \tag{10}$$

and

$$\left.\begin{array}{c} H_{\alpha_1}u(t) = h_{\alpha_1} \\ H_{\alpha_1}u(t+1) = h_{\alpha_1} \\ \vdots \\ H_{\alpha_1}u(t+N_2-2) = h_{\alpha_1} \\ H_{\alpha_2}u(t+N_2-1) = h_{\alpha_2} \\ H_{\alpha_2}u(t+N_2) = h_{\alpha_2} \\ \vdots \\ H_{\alpha_{N_S}}u(t+N-1) = h_{\alpha_{N_S}} \end{array}\right\} \tag{11}$$

where $C_\tau = (A^{\tau-1}B|A^{\tau-2}B|\cdots|B)$ and the $rN \times rN$-matrix $\tilde{E}_\tau$ is defined by

$$\tilde{E}_\tau = \begin{pmatrix} 0 & 0 \\ I_{r\tau\times r\tau} & 0 \end{pmatrix} \tag{12}$$

In (10) we have used the well known formula

$$x(t+\tau) = A^\tau x(t) + C_N\tilde{E}_\tau\tilde{u}(t) \tag{13}$$

where $\tilde{u}(t) = (u^T(t),\ u^T(t+1), ...,\ u^T(t+N-1))^T$. Removing from (11) and (10) equations that are a priori known to be infeasible and duplicated equations, (11) and (10) may be stacked into the following set of equations

$$L_k\tilde{u}(t) = M_k x(t) + M_k^g g + M_k^h h \tag{14}$$

where k is an index in the index set $\mathcal{C} = \{0, 1, ..., N_K - 1\}$ enumerating the finite set of all active constraint set sequences generated by the constraints (5), (6) and the division into subintervals (9). Let $k_0 \in \mathcal{C}$ be the index to the active constraint set sequence with no active constraints.

2.2 Decomposition of the HJB equation

The minimization problem (8) with the stated constraints is a 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 reformulation of the minimization in (8) into two nested parts where one part only involves equality contraints and the other part is a finite discrete optimization problem over all allowed active constraint set sequences. The part that involves equality constraints can then be solved explicitly offline using Lagrange multipliers, while the discrete optimization problem can in many cases also be solved offline or at least reduced to a simpler problem and then solved in real-time in a efficient manner. Next, define the $r \times rN$ matrix

$$E_\tau = (0_{r\times r}, ..., 0_{r\times r}, I_{r\times r}, 0_{r\times r}, ...0_{r\times r}) \tag{15}$$

where the $I_{r\times r}$ is at the τ-th $r \times r$ block. The following result is then evident from [1, 2]:

Theorem 1 *(Nested HJB equation) Assume the minimum in the HJB equation (8) exists. With N sufficiently large and no restrictions on the active constraint set sequence's allowed switching times on the horizon ($S = N$), the HJB equation (8) is equivalent to*

$$0 = \min_{k\in\mathcal{C}} \left(\min_{\substack{\tilde{u}(t)\in R^{rN} \\ L_k\tilde{u}=M_k x(t)+M_k^g g+M_k^h h}} (V(x(t+N)) - V(x(t)) + \sum_{\tau=0}^{N-1} l_{QR}(x(t+\tau), E_{\tau+1}\tilde{u}(t)) \right) \quad (16)$$

where the outer minimization is subject to the constraints

$$HE_\tau \tilde{u}_k^*(x(t), g, h) \le h \quad (17)$$

$$G(A^\tau x(t) + C_N \tilde{E}_\tau \tilde{u}_k^*(x(t), g, h)) \le g \quad (18)$$

for all $1 \le \tau \le N$ and $\tilde{u}_k^*(x(t), g, h)$ is the $\tilde{u}(t)$ solving the innermost optimization problem in (16).

Determining the optimal cost function V is in general an extremely difficult problem, so similar to [11] we utilize a lower bound $\underline{V}(x) = x^T \underline{P} x$, where $\underline{P}$ solves the algebraic Ricatti equation of the unconstrained problem, as a suboptimal approximation in the control design in Section 2.3, and analyse the loss of performance resulting from this approximation later in Section 3.

2.3 Feedback design

Consider the innermost optimization on the RHS of (16) with the lower bound $\underline{V}$ substituted for V. This problem can be stated as follows: For a given active constraint set sequence (with index $k \in \mathcal{C}$) solve the problem

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

where

$$\underline{J}(\tilde{u}(t), x(t)) = \underline{V}(x(t+N)) - \underline{V}(x(t)) + \sum_{\tau=0}^{N-1} l_{QR}(x(t+\tau), E_{\tau+1}\tilde{u}(t)) \quad (19)$$

$$= x^T S_1 x + 2x^T S_2 \tilde{u} + \tilde{u}^T S_3 \tilde{u} \quad (20)$$

and

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

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

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

with the $rN \times rN$-matrix $\tilde{R} = \mathrm{diag}(R, R, ..., R)$. The outer finite discrete optimization problem of (16) is restated as

$$k^*(x, g, h) = \arg\min_{k\in\mathcal{C}} \underline{J}(\tilde{u}_k^*(x, g, h), x) \quad (21)$$

where the minimization is subject to

$$HE_\tau \tilde{u}_k^*(x, g, h) \le h \quad (22)$$

$$G(A^\tau x + C_N \tilde{E}_\tau \tilde{u}_k^*(x, g, h)) \le g \quad (23)$$

for all $1 \le \tau \le N$. The optimization problem (21)-(23) is feasible if and only if $(x, g, h) \in Z^F$:

$$Z^F = \bigcup_{k\in\mathcal{C}} Z_k^F$$

$$Z_k^F = \{(x, g, h) \in R^n \times \mathcal{G} \times \mathcal{H} \mid HE_\tau \tilde{u}_k^*(x, g, h) \le h, \; G(A^\tau x + C_N \tilde{E}_\tau^T \tilde{u}_k^*(x, g, h)) \le g, \text{ for } 1 \le \tau \le N\}$$

For $(x, g, h) \in Z^F$, the suboptimal constrained LQR is now

$$u^*(x, g, h) = E_1 \tilde{u}_{k^*(x,g,h)}^*(x, g, h) \quad (24)$$

Theorem 2 *(Piecewise linear state feedback design) Consider a fixed active constraint set sequence with index $k \in \mathcal{C}$. The solution to the constrained quadratic optimization problem (19) is given by the affine state feedback*

$$\tilde{u}_k^*(x, g, h) = \begin{cases} K_{k,2} x, & \text{if } k = k_0 \\ K_{k,1}^g g + K_{k,1}^h h + K_{k,2} x, & \text{if } k \ne k_0 \end{cases} \quad (25)$$

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

$$K_{k,1}^g = S_3^{-1} L_k^T \left(L_k S_3^{-1} L_k^T \right)^{-1} M_k^g \quad (26)$$

$$K_{k,1}^h = S_3^{-1} L_k^T \left(L_k S_3^{-1} L_k^T \right)^{-1} M_k^h \quad (27)$$

$$K_{k,2} = -S_3^{-1} \left(\left(I - L_k^T \left(L_k S_3^{-1} L_k^T \right)^{-1} L_k S_3^{-1} \right) S_2^T - L_k^T \left(L_k S_3^{-1} L_k^T \right)^{-1} M_k \right) \quad (28)$$

for $k \ne k_0$ and $(x, g, h) \in Z^F$.

A proof is given in [10]. The affine state feedback (25) is parameterized such that individual constraints can be deactivated and the constraint limits may be changed on-line without changing the parameters of the controller. The reason for this is that the constraint limits appear only in g, h, and that the precomputed feedback matrices $K_{k,1}^g, K_{k,1}^h$ and $K_{k,2}$ do not depend on these limits.

2.4 Recovery from infeasibility

The optimization problem (21)-(23) will not have solutions that are feasible when $(x, g, h) \notin Z^F$. Then one may relax the problem by allowing minimum violation of some of the constraints according to some prioritizations. 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 $(x, g, h) \notin Z^F$, one might minimize the criterion

$$\nu_k(x, g, h) = \quad (29)$$

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

subject to "hard" constraints for $1 \le \tau \le N$

$$G_{\beta_{hard}} \left(A^\tau x + C_N \tilde{E}_\tau \tilde{u}_k^*(x, g, h) \right) \le g_{\beta_{hard}} \quad (30)$$

$$H_{\alpha_{hard}} E_\tau \tilde{u}_k^*(x, g, h) \le h_{\alpha_{hard}} \quad (31)$$

with respect to $k \in \mathcal{C}$. 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 (29)-(31) is feasible when $(x,g,h) \in Z^R$, where

$$\begin{aligned} Z^R &= \bigcup_{k\in\mathcal{C}} Z_k^R \\ Z_k^R &= \left\{(x,g,h) \in R^n \times \mathcal{G} \times \mathcal{H} - Z^F \mid \text{s.t. } (30)-(31)\right\} \end{aligned}$$

If $(x,g,h) \notin Z^F \cup Z^R$, 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 (29)-(31) be denoted $k^*(x,g,h)$ and the associated control input $\tilde{u}^*_{k^*(x,g,h)}(x,g,h)$. Furthermore, let $Z = Z^F \cup Z^R$ and define for $(x,g,h) \in Z$ the control input of the suboptimal constrained LQR with infeasibility handler:

$$u^*(x,g,h) = E_1 \tilde{u}^*_{k^*(x,g,h)}(x,g,h) \qquad (32)$$

2.5 Avoiding high-gain feedback, chattering and sliding modes

Effectively, the active state constraint is enforced by a sliding-mode like strategy. This is partly due to the choice of a very small N, but the problem can be resolved by modifying the state constraints (10) such that they do not require the active state constraint sets 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 $G_\beta x(t+\tau) = g_\beta$ in (10) by its asymptotic version

$$\begin{aligned} G_\beta(x(t+\tau) + C_1 x(t+\tau-1) + ... + C_\tau x(t)) &= \\ (1 + C_1 + ... + C_\tau) g_\beta & \qquad (33) \end{aligned}$$

such that $1 + C_1 q^{-1} + ... + C_\tau q^{-\tau}$ has prescribed roots. This implies convergence of the state towards the subspace defined by the active state constraint set. The convergence rate can be tuned by the choice of $C_1, ..., C_\tau$. 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 [12]. 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 (21)-(23)

$$G_{\beta(x,g)}(A^\tau x(t) + C_N \tilde{E}_\tau \tilde{u}^*_k(x(t),g,h)) \leq G_{\beta(x,g)} x(t) \qquad (34)$$

if $\beta_k \subset \beta(x,g)$ for $1 \leq \tau \leq N$. The symbol $\beta(x,g)$ denotes the set of currently ε-active state constraints at (x,g), i.e. $\beta(x,g) = \{l \in \{1,2,...,q\} \mid |(G_{l1}, ..., G_{ln})x - g_l| \leq \varepsilon_l\}$, where $\varepsilon_l > 0$ defines the boundary layer. Eq. (34) excludes non-attractive feedbacks that tend to move the state towards violation of active state constraints.

2.6 Real-time algorithms

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,g,h). 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 $X \times \mathcal{G} \times \mathcal{H}$ 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. Certainly, in a high-dimensional problem with a large number of regions, an explicit offline characterization of the regions where each affine feedback is active is unlikely to be computationally attractive. Alternatively, 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 (21)-(23) and (29)-(31) are solved in real time. Discrete search techniques such as branch-and-bound can be applied for this purpose.
2. A partitioning of $X \times \mathcal{G} \times \mathcal{H}$ 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.

2.7 Activity regions and partition

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

$$Z_k = \{(x,g,h) \in Z \mid k = k^*(x,g,h)\} \qquad (35)$$

Together with the affine functions (25), the activity regions $\{Z_k,\ k \in \mathcal{C}\}$ completely describes the PWL structure of the controller. As mentioned in section 2.6, in order to have flexibility to address tradeoffs in terms of real-time computer memory and processing capacity, it is of interest to compute outer approximations to these activity sets. More specifically, we compute a partition of the state and constraint limit space such that within each region a (given) small number of active constraint set sequences may be optimal. A computational procedure is given below, see [10] for details:

1. Let $\mathcal{E} := \emptyset$, and $\mathcal{U} := \{Z\}$.
2. If $\mathcal{U} = \emptyset$, the partition generated by this algorithm is $\mathcal{P} = \mathcal{E}$ and the algorithm terminates.
3. Let $\mathcal{Z}_0 \in \mathcal{U}$ be arbitrary. Let $\mathcal{O}_0$ contain the candidate optimal active constraint set seqeuences in $\mathcal{Z}_0$. These can be computed by solving a number of LP/QP problems, see [10].
4. If $\mathcal{O}_0$ contains a sufficiently small number of elements, add $\mathcal{Z}_0$ to the set of explored subsets $\mathcal{E}$ and remove $\mathcal{Z}_0$ from the set of unexplored subsets $\mathcal{U}$. Go to step 2.
5. Select a hyperplane and split $\mathcal{Z}_0$ into two non-empty parts $\mathcal{Z}_0^+$ and $\mathcal{Z}_0^-$. Add $\mathcal{Z}_0^+$ and $\mathcal{Z}_0^-$ to $\mathcal{U}$ and remove $\mathcal{Z}_0$ from $\mathcal{U}$. Go to step 2.

The set $\mathcal{E}$ contains the set of explored subsets of Z, while the set $\mathcal{U}$ contains the set of unexplored subsets of Z. The algorithm will explore the candidate optimal active constraint sets associated with each element of $\mathcal{E}$ sequentially. The candidate hyperplanes for splitting may be problem-independent (for example ones that allow efficient real-time computations) or chosen from the state, input and admissibility constraints of the problem.

2.8 Example

Consider the continuous-time double integrator

$$A_c = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad B_c = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{36}$$

with the discretized model

$$A = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} T_s^2 \\ T_s \end{pmatrix} \tag{37}$$

using a sampling-interval $T_s = 0.05$. The control objective is defined by the cost function

$$l_{QR}(x,u) \quad = \quad x_1^2 + u^2 \quad = \quad x^T \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} x + u^2 \tag{38}$$

and the constraints

$$-0.5 \quad \leq \quad x_2 \quad \leq \quad 0.5 \tag{39}$$

$$-1 \quad \leq \quad u \quad \leq \quad 1 \tag{40}$$

If x_1 is interpreted as position, x_2 as speed and u as force, the objective is to control the position under constraints on the speed and force. The state constraint will be of the form (5) by defining

$$G = \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix}, \qquad g = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \tag{41}$$

and the input constraints of the form

$$H = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \qquad h = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{42}$$

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 double poles at 0.9. 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: In order to reduce the position error the speed is first increased at a maxmimum 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 choosen the smallest possible horizon, namely $N = S = 1$ since this is advantageous for computational reasons. However, this extreme choice does not appear to be significantly suboptimal. Now, the region of feasibility is $X^F = \{x \in R^2 \mid |x_2| \leq 0.55\}$ 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 (40) as hard (non-relaxable) constraints, and the state constraints (39) 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 constaints are assoicated with the input only, $X^R = R^2$. The piecewise linear feedback control law is shown in Figure 2. 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. Within each region, the feedback is linear, cf. Figure 2. We observe that in this case the regions can be characterized as unions of polyhedra.

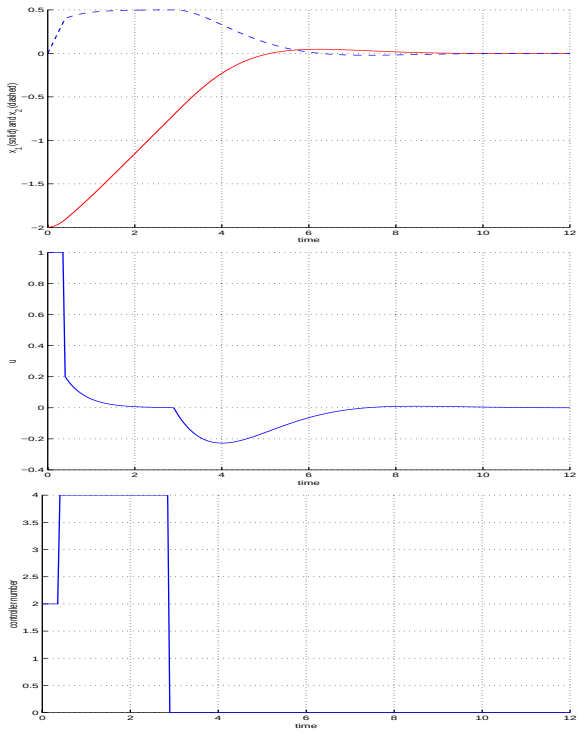

Figure 1: Constrained control of a double integrator from initial state $x(0) = (-2, 0)^T$, with a boundary layer around the active state constraints.

3 Stability and Performance Analysis

In this section we analyse the stability and performance of the PWL suboptimal constrained LQR. Due to the suboptimality these properties are not known a priori, and should be computed for each particular control design. Exploiting the PWL nature of the closed loop system, we utilize recently developed computational tools for PWL system analysis based on LMIs (linear matrix inequalities) [13, 11, 14] allowing us to compute quadratic and piecewise quadratic (PWQ) Lyapunov functions and upper/lower bounds on the cost function of the controller, see [10] for details.

3.1 Example

A PWQ Lyapunov function for the constrained LQR for the double integrator with boundary layers is illustrated in Figure 4. Upper and lower bounds on the cost are illustrated in Figure 5.

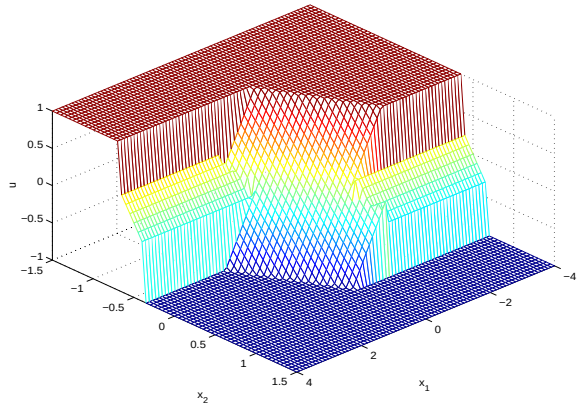

Figure 2: PWL constrained LQ controller for the double integator, with a boundary layer around the active state constraints.

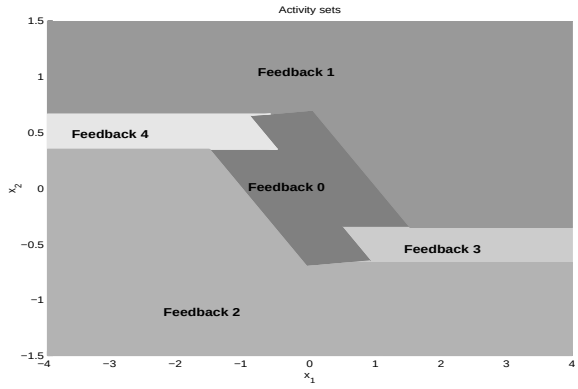

Figure 3: Activity regions for the five constituent affine feedbacks in the constrained LQR for the double integrator with boundary layers.

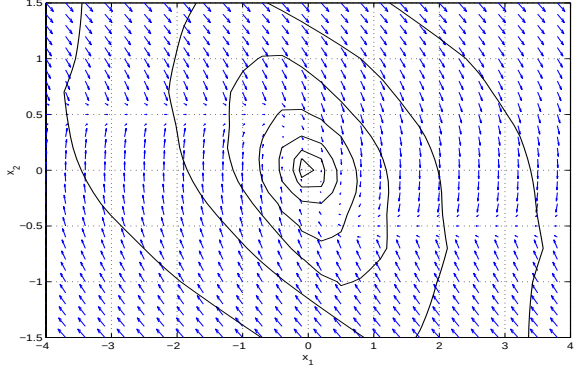

Figure 4: Level curves of a PWQ Lyapunov function.

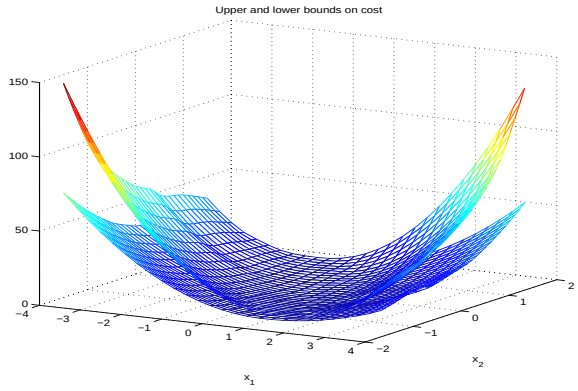

Figure 5: Upper and lower bounds on the optimal cost function $V(x)$.

References

[1] M. Sznaier and M. J. Damborg, "Suboptimal control of linear systems with state and control inequality constraints", in *Proc. 26th Conf. Decision and Control*, 1987, pp. 761–762.

[2] D. Chmielewski and V. Manousiouthakis, "On constrained infinite-time linear quadratic optimal control", *Systems and Control Letters*, vol. 29, pp. 121–129, 1996.

[3] P. O. M. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation", *IEEE Trans. Automatic Control*, vol. 43, pp. 1163–1169, 1998.

[4] L. Chisci, "Fast algorithm for constrained infinite horizon LQ problem", *Int. J. Control*, vol. 72, pp. 1020–1026, 1999.

[5] M. Sznaier and M. J. Damborg, "Heuristically enhanced feedback control of constrained discrete-time linear systems", *Automatica*, vol. 26, pp. 521–532, 1990.

[6] G. F. Wredenhagen and P. R. Belanger, "Piecewise-linear LQ control for systems with input constraints", *Automatica*, vol. 30, pp. 403–416, 1994.

[7] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems", Tech. Rep. AUT99-16, Automatic Control Laboratory, ETH Zurich, Switzerland, September 1999, http://www.control.ethz.ch/pub/publications/AUTAr-99-33.pdf.

[8] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit solution of model predictive control via multiparametric quadratic programming", in *Proc. American Control Conference, Chicago*, 2000, pp. 872–876.

[9] E. Zafiriou, "Robust model predictive control of processes with hard constraints", *Computers and Chemical Engineering*, vol. 14, pp. 359–371, 1990.

[10] T. A. Johansen, I. Petersen, and O. Slupphaug, "Explicit suboptimal linear quadratic regulation with input and state constraints", Preprint, submitted for publication, www.itk.ntnu.no/ansatte/Johansen_Tor.Arne/lqrc_report.pdf, 2000.

[11] A. Rantzer and M. Johansson, "Piecewise linear quadratic optimal control", *IEEE Trans. Automatic Control*, vol. 45, pp. 629–637, 2000.

[12] J.-J. E. Slotine, "Sliding controller design for nonlinear systems", *Int. J. Control*, vol. 40, pp. 421–434, 1984.

[13] M. Johansson and A. Rantzer, "Computation of piecewise quadratic Lyapunov functions for hybrid systems", *IEEE Trans. Automatic Control*, vol. 43, pp. 555–559, 1998.

[14] S. Hedlund and M. Johansson, "A toolbox for computational analysis of piecewise linear systems", in *Proceedings of European Control Conference, Karlsruhe*, 1999.