Approximate Explicit Model Predictive Control Incorporating Heuristics

Alexandra Grancharova[1 2 3] **and Tor A. Johansen**[4]

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

Abstract

Explicit piecewise linear state feedback solutions to the constrained linear model predictive control problem has recently been characterized and computed numerically using multi-parametric quadratic programming. The piecewise linear state feedback is defined on a polyhedral partitioning of the state space, which may be quite complex. Here we suggest an *approximate* multi-parametric quadratic programming approach, which has the advantages that the partition is structured as a binary search tree. This leads to real-time computation of the piecewise linear state feedback with a computational complexity that is logarithmic with respect to the number of regions in the partition. The algorithm is based on heuristic rules that are used to partition the state space and estimate the approximation error.

1 Introduction

The main motivation behind explicit model predictive control (MPC) is that an explicit state feedback solution avoids the need for real-time optimization, and is therefore potentially useful for embedded control applications with fast sampling where MPC has not traditionally been used. In [1, 2] it was recognized that the constrained linear MPC problem is a multi-parametric quadratic program (mp-QP), when the state is viewed as a parameter to the problem. It was shown that the solution (the control input) has an explicit representation as a piecewise linear (PWL) state feedback on a polyhedral partition of the state space, see also [3, 4, 5], and they develop an mp-QP algorithm to compute this function. In [6] an alternative efficient mp-QP algorithm is developed.

Even though an explicit PWL state feedback representation of the MPC is found, it may be quite complex. It is of interest to study approximate solutions which may offer computational advantages. A sub-optimal approach was developed in [3, 7], by introducing constraints on the allowed number of active set switches on the horizon. An alternative sub-optimal approach was introduced in [8] where small slacks are introduced on the optimality conditions and the mp-QP algorithm [1] is modified for the relaxed problem. An algorithm for approximate mp-QP was developed in [9], based on the idea that significant real-time computational advantages are achieved by restricting the state space partition to be orthogonal and represented as a search tree. The real-time computational complexity can also be reduced by exploiting the convexity of the cost function [10], and it should be mentioned that recent work has also focused on reducing the computational complexity of MPC implemented by real-time optimization, e.g. [11, 12].

The present paper extends the authors work [9], where the approximate mp-QP algorithm is guaranteed to terminate with an approximate solution that satisfies a specified maximum allowed error in the cost function and constraint violations. It is of interest to study alternative approaches since sometimes it may be more natural to specify the approximation tolerance in terms of the error in the control input rather than the cost. Moreover, it is desirable to fully exploit that one is only interested in the first sample of the optimal control trajectory for receding horizon MPC implementation. Hence, we study a modified algorithm that allows an approximation tolerance on the control input at the first sample to be specified. Such an algorithm is developed in this paper, and we suggest the use of a $k-d$ tree [13, 14] as a more flexible and powerful alternative to the generalized quad/oct-tree used in [9]. In [15] it was shown how also in the exact case a binary search tree can be computed a posteriori to reduce the computational complexity.

2 Explicit MPC and exact mp-QP

Formulating a linear MPC problem as an mp-QP is briefly described below, see [1] for further details. Consider the discrete-time linear system

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

where $x(t) \in \mathbb{R}^n$ is the state variable, $u(t) \in \mathbb{R}^m$ is the input variable, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and (A, B) is a controllable system. For the current $x(t)$, MPC solves the optimization problem

$$V^*(x(t)) \quad = \quad \min_{U \triangleq \{u_t, \ldots, u_{t+N-1}\}} J(U, x(t)) \tag{2}$$

subject to

$$\begin{aligned}
&y_{\min} \le y_{t+k|t} \le y_{\max},\ k = 1, ..., N \\
&u_{\min} \le u_{t+k} \le u_{\max},\ k = 0, 1, ..., N-1, \\
&x_{t|t} = x(t) \\
&x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k},\ k \ge 0 \\
&y_{t+k|t} = Cx_{t+k|t},\ k \ge 0
\end{aligned} \tag{3}$$

[1] `Alexandra.Grancharova@itk.ntnu.no`

[2] This work was sponsored by the European Commission through the Research Training Network MAC

[3] On leave from Institute of Control and System Research, Bulgarian Academy of Sciences, Sofia, Bulgaria.

[4] `Tor.Arne.Johansen@itk.ntnu.no`

with the cost function given by

$$J(U, x(t)) = \sum_{k=0}^{N-1} \left(x_{t+k|t}^T Q x_{t+k|t} + u_{t+k}^T R u_{t+k} \right) + x_{t+N|t}^T P x_{t+N|t} \quad (4)$$

and symmetric $R > 0$, $Q \geq 0$, $P > 0$. The final cost matrix P may be taken as the solution of the algebraic Riccati equation. With the assumption that no constraints are active for $k \geq N$ this corresponds to an infinite horizon LQ criterion [16]. This and related problems can by some algebraic manipulations be reformulated as

$$V_z^*(x) = \min_z \frac{1}{2} z^T H z \quad (5)$$

$$\text{subject to} \quad Gz \leq W + Sx \quad (6)$$

where $z = U + H^{-1}F^T x$. Notice that $H > 0$ since $R > 0$. The vector x is the current state, which can be treated as a vector of parameters. A similar reformulation can also be found for the tracking problem or when infeasibility relaxations or other variations of the MPC problem are included, [1]. For ease of notation we write x instead of $x(t)$. The number of inequalities is denoted q and the number of free variables is $n_z = m \cdot N$. Then $z \in \mathbb{R}^{n_z}$, $H \in \mathbb{R}^{n_z \times n_z}$, $G \in \mathbb{R}^{q \times n_z}$, $W \in \mathbb{R}^{q \times 1}$, $S \in \mathbb{R}^{q \times n}$. The solution of the optimization problem (5)-(6) can be found in an explicit form $z^* = z^*(x)$, [1]:

Theorem 1 *Consider the mp-QP (5)-(6) and suppose $H > 0$. The solution $z^*(x)$ (and $U^*(x)$) is a continuous PWL function of x defined over a polyhedral partition of the state space, and $V_z^*(x)$ is a convex (and therefore continuous) piecewise quadratic function.*

As shown in [1], the mp-QP problem (5) - (6) can be solved by applying the Karush-Kuhn-Tucker conditions which leads to the following explicit solution for a given active set:

$$z = H^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \quad (7)$$

The matrices $\tilde{G}$, $\tilde{W}$ and $\tilde{S}$ contain the rows of G, W and S corresponding to the given active set. As long as this active set remains optimal, the solution (7) remains optimal, when z is viewed as a function of x. First, z must remain feasible

$$GH^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \leq W + Sx. \quad (8)$$

Second, the Lagrange multipliers λ must remain nonnegative

$$\lambda(x) = -(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \geq 0. \quad (9)$$

The inequalities (8) and (9) describe a polyhedron in the state space. This region is denoted as the critical region CR_0 corresponding to the given set of active constraints. It is a polyhedral set and represents the largest set of parameters x such that the combination of active constraints at the minimizer remains optimal. Algorithms for iteratively constructing a polyhedral partition of the state space into critical regions and computing the PWL solution function are given in [2, 6].

3 Approximate mp-QP algorithm

We will in this section describe an approximate mp-QP algorithm. We restrict our attention to a hyper-rectangle $X \subset \mathbb{R}^n$ where we seek to approximate the optimal PWL solution $z^*(x)$ to the mp-QP (5)-(6). In order to minimize the real-time computational complexity we require that the state space partition is orthogonal and can be represented as a $k-d$ tree, [13, 14], such that the search complexity is logarithmic with respect to the number of regions. The $k-d$ tree is a hierarchical data structure where a hyper-rectangle can be sub-divided into smaller hyper-rectangles allowing the local resolution to be adapted, cf. Figure 1. When searching the tree, only one scalar comparison is required at each level, leading to extremely fast real-time MPC computations.

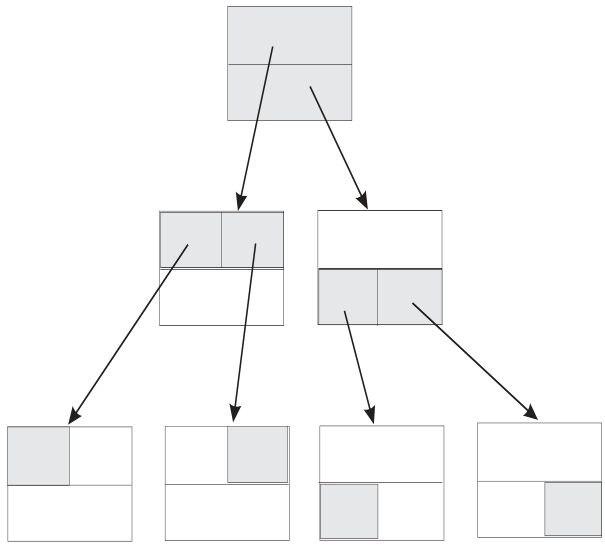

Figure 1: $k-d$-tree partition of a rectangular region in $\mathbb{R}^2$.

Initially the algorithm will consider the whole region $X_0 = X$. The main idea of the approximate mp-QP algorithm is to compute the solution of the problem (5)-(6) at the 2^n vertices of the hyper-rectangle X_0, by solving up to 2^n QPs. Based on these solutions we compute a local approximation $\hat{z}_0(x)$ to the PWL optimal solution $z^*(x)$, restricted to the hyper-rectangle X_0. Assume for a moment there exist feasible solutions at all vertices of X_0. As described above, the linear solution (7) is exact in some polyhedral critical region. A linear approximation may be computed as a least-squares fit under the constraint of feasibility of the solution, [8]:

Lemma 1 *Consider the polyhedral region X_0 with vertices $\{v_1, v_2, ..., v_M\}$. Let K_0 and g_0 solve the least squares problem for some weighting matrix $\Sigma_0 > 0$*

$$\min_{K_0, g_0} \sum_{i=1}^{M} (z^*(v_i) - K_0 v_i - g_0)^T \Sigma_0 (z^*(v_i) - K_0 v_i - g_0) \quad (10)$$

subject to

$$G(K_0 v_i + g_0) \leq S v_i + W, \qquad i \in \{1, 2, ..., M\} \quad (11)$$

Then the least squares approximation $\hat{z}_0(x) = K_0 x + g_0$ is feasible for all $x \in X_0$.

This follows directly from the convexity. We define the error in the solution

$$e(x) = (z^*(x) - \hat{z}_0(x))^T \Sigma (z^*(x) - \hat{z}_0(x)) \qquad (12)$$

where $\Sigma \geq 0$ is a weighting matrix which typically has non-zero weight only on the components of the solution corresponding to the first sample of the trajectory. Ideally, we would like to find an approximate PWL solution which respects a pre-specified tolerance $\bar{\varepsilon} > 0$ on the uniform error

$$\varepsilon \;=\; \max_{x \in X_0} e(x) \qquad (13)$$

Consider an arbitrary polyhedral set $X_0 \subset X \subset \mathbb{R}^n$, and suppose the exact optimal solution $z_i^0(x)$ (defined by (7)) at the vertices $\{v_1, v_2, ..., v_M\}$ of X_0 and an approximate solution $\hat{z}_0(x)$ in X_0 are known.

Lemma 2 *Consider a polyhedron $X_0 \subset \mathbb{R}^n$ with vertices $\{v_1, v_2, ..., v_M\}$ and any affine approximate solution $\hat{z}_0$. If $X_0 \subset \cup_i CR_i$, then*

$$\varepsilon \leq \max_{i,j} (z_i^0(v_j) - \hat{z}_0(v_j))^T \Sigma (z_i^0(v_j) - \hat{z}_0(v_j)) \qquad (14)$$

Proof. Let $x \in X_0$ be arbitrary. Since $X_0 \subset \cup_i CR_i$ there exists a $k \in \{1, 2, ..., M\}$ such that $z^*(x) = z_k^0(x)$. Assume without loss of generality that $z^*(x) - \hat{z}_0(x) \geq 0$. Because z_k^0 and $\hat{z}_0$ are linear functions it follows that there exists a vertex v_j such that $z_k^0(v_j) \geq z_k^0(x) = z^*(x) \geq \hat{z}_0(x)$. It is evident that the error $z^*(x) - \hat{z}_0(x)$ is largest for x at some vertex, and it suffices to consider only the vertices on X_0 when computing the error bound (13).

□

Observe that the error bound (14) can be computed using information of the solutions at the vertices only if $X_0 \subset \cup_i CR_i$. Unfortunately, this condition is not straightforward to verify without knowing the exact solution in X_0. In special cases, for example when the active set is the same at all vertices, the condition in Lemma 2 is easily verified to hold. Better estimates than (14) can in general be found by utilizing additional information, such as sampling the exact solution also at the interior of X_0 or compute the exact solution using an mp-QP solver. In this paper we use the following estimate

$$\hat{\varepsilon} \;=\; \max_{x \in X_0^d} e(x) \qquad (15)$$

where $X_0^d \subset X_0$ contains a finite number of representative points in X_0, typically the vertices of one or more hyper-rectangles contained in the interior of X_0.

Algorithm 1 (approximate mp-QP)

Step 1. Initialize the partition to the whole hyper-rectangle, i.e. $\mathcal{P} = \{X\}$. Mark the hyper-rectangle X as unexplored.

Step 2. Select any unexplored hyper-rectangle $X_0 \in \mathcal{P}$. If no such hyper-rectangle exists, the algorithm terminates with the partition $\mathcal{P}$.

Step 3. Compute the solution to the QP (5)-(6) for x fixed to each of the 2^n vertices of the hyper-rectangle X_0 and its center point (some of these QPs may have been solved in earlier steps). If one or more solutions are infeasible, go to step 6. Otherwise, go to step 4.

Step 4. From the optimal solutions at the vertices of the hyper-rectangle, compute a local linear state feedback as an approximation to be used in the hyper-rectangle X_0, using Lemma 1. If the least-squares problem in Lemma 1 is feasible, go to step 5, otherwise go to step 6.

Step 5. Compute the estimate $\hat{\varepsilon}$ of ε from (15). If $\hat{\varepsilon} \leq \bar{\varepsilon}$, mark X_0 as explored and go to step 2. Otherwise, go to step 6.

Step 6. Split the hyper-rectangle X_0 into two hyper-rectangles X_1 and X_2. Mark them unexplored, remove X_0 from $\mathcal{P}$, add X_1 and X_2 to $\mathcal{P}$, and go to step 2.

□

Step 6 needs further specification of how the hyper-rectangle is split. Assume first that the solutions at all vertices are feasible. We suggest a heuristic rule for how to split a hyper-rectangle X_0 into two hyper-rectangles such that the error $e(x)$ in each hyper-rectangle is significantly reduced. The rule attempts to split the hyper-rectangle at the axis along which the change of error $e(x)$ is maximal (before splitting), because it is reasonable to hope this is how the largest reduction of the error $e(x)$ can be made.

Heuristic splitting rule. Split the hyper-rectangle X_0 by a hyperplane through its center and orthogonal to the axis x_j where the total absolute change κ_j of the approximation error measured both at the facet centers of X_0 and the vertices of X_0^d is maximal.

□

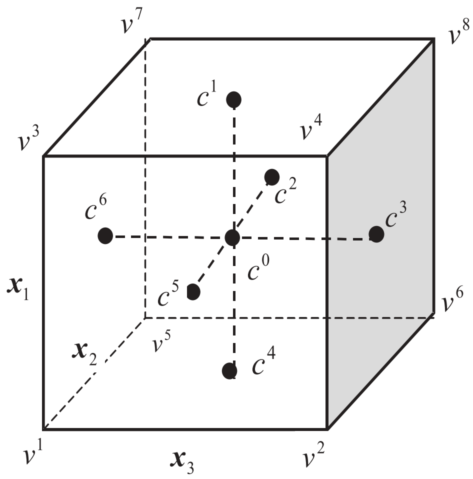

Figure 2: Illustration of the heuristic splitting rule

Consider Figure 2, where $c^1, c^2, ..., c^6$ are the facet centers of X_0, c^0 is the center point of X_0 and $v^1, v^2, ..., v^8$ are the vertices of $X_0^d \subset X_0$. Then, the total absolute changes κ_1, κ_2 and κ_3 of the approximation error along the axes x_1, x_2 and

x_3 are estimated as

$$\begin{aligned}\kappa_1 &= \left|e(v^3)-e(v^1)\right| + \left|e(v^4)-e(v^2)\right| \\ &\quad + \left|e(v^7)-e(v^5)\right| + \left|e(v^8)-e(v^6)\right| \\ &\quad + \left|e(c^1)-e(c^0)\right| + \left|e(c^4)-e(c^0)\right| \quad (16)\\ \kappa_2 &= \left|e(v^5)-e(v^1)\right| + \left|e(v^6)-e(v^2)\right| \\ &\quad + \left|e(v^7)-e(v^3)\right| + \left|e(v^8)-e(v^4)\right| \\ &\quad + \left|e(c^2)-e(c^0)\right| + \left|e(c^5)-e(c^0)\right| \quad (17)\\ \kappa_3 &= \left|e(v^2)-e(v^1)\right| + \left|e(v^4)-e(v^3)\right| \\ &\quad + \left|e(v^6)-e(v^5)\right| + \left|e(v^8)-e(v^7)\right| \\ &\quad + \left|e(c^3)-e(c^0)\right| + \left|e(c^6)-e(c^0)\right| \quad (18)\end{aligned}$$

The splitting hyperplane is then selected to be orthogonal to the axis x_j for which κ_j is largest.

If the solution is infeasible at one or more vertices, or the least-squares problem in Lemma 1 is infeasible, the above rule is not used. Instead we apply the following rules

Infeasibility heuristic rules.

1. If all vertices are infeasible, no splitting is necessary, since by convexity arguments every point in the hyper-rectangle has an infeasible solution. An exception is when the state space region to be partitioned is chosen too large so that all vertices of the initial hyper-rectangle are infeasible and the feasible part is inside the hyper-rectangle. If this occur, the initial hyper-rectangle is split on arbitrary axes until one or more feasible points appear.

2. Split the hyper-rectangle on all state space axes where the solution changes from feasible to infeasible.

3. If the least-squares problem in Lemma 1 is infeasible, split the hyper-rectangle on an arbitrary axis.

□

The state $x(t)$ used for feedback is usually uncertain due to measurement or estimation errors. Hence, no overall accuracy is gained by allowing extremely small regions in the partition, since they cannot be distinguished due to state uncertainty. Hence, we allow a minimum size on each region in the partition to be specified.

The tolerance $\overline{\varepsilon} > 0$ is typically chosen to depend on X_0 such that

$$\overline{\varepsilon} = \max\left(\overline{\varepsilon}_a, \overline{\varepsilon}_r \max_{x\in X_0} ||\hat{z}_0(x)||_2^2\right) \quad (19)$$

where $\overline{\varepsilon}_a > 0$ and $\overline{\varepsilon}_r > 0$ can be interpreted as absolute and relative tolerances, respectively.

4 Example

Consider the double integrator [3]

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

$N;\sqrt{\overline{\varepsilon}_r}$	N_r	err^{avg}	err^{max}
2; 0.5	97	0.073	0.45
2; 0.4	105	0.054	0.35
2; 0.3	111	0.040	0.28
2; 0.2	121	0.020	0.18
2; 0.1	157	0.011	0.13
10; 0.5	97	0.086	0.45
10; 0.4	105	0.067	0.35
10; 0.3	109	0.058	0.28
10; 0.2	121	0.031	0.18
10; 0.1	156	0.017	0.13

Table 1: Complexity of partition using the heuristic splitting rule with different horizon N, relative tolerance $\overline{\varepsilon}_r$. The columns err^{avg} and err^{max} contain the average and maximum error in the solution (control input) for each partition.

$N;\sqrt{\overline{\varepsilon}_r}$	N_r	err^{avg}	err^{max}
2; 0.5	364	0.023	0.35
2; 0.4	376	0.023	0.29
2; 0.3	376	0.023	0.29
2; 0.2	412	0.019	0.20
2; 0.1	484	0.007	0.09
10; 0.5	352	0.035	0.35
10; 0.4	364	0.034	0.25
10; 0.3	364	0.034	0.25
10; 0.2	400	0.028	0.20
10; 0.1	460	0.014	0.10

Table 2: Complexity of partition, with no heuristic splitting rule used, with different horizon N, relative tolerance $\overline{\varepsilon}_r$. The columns err^{avg} and err^{max} contain the average and maximum error in the solution (control input) for each partition.

with the sampling interval $T_s = 0.05$. Consider the MPC problem with cost matrices $Q = \text{diag}(1, 0)$, $R = 1$, and the matrix $P > 0$ given as the solution of the algebraic Riccati equation. The constraints are $-0.5 \le x_2 \le 0.5$ and $-1 \le u \le 1$.

Table 1 shows how the number of regions in the state space partition, generated by the approximate mp-QP algorithm, depends on the horizon N, and the relative tolerance $\overline{\varepsilon}_r$. N_r is the number of regions in the partition. In all cases $\sqrt{\overline{\varepsilon}_a} = 0.001$ and we have restricted the size of the regions to be larger than $\Delta_{x_1} = 0.06$ and $\Delta_{x_2} = 0.01$. The matrix Σ in the criterion is chosen such that only the first sample on the input trajectory is given weight. Table 1 also shows how the achieved approximation error is closely related to the specified tolerance. Although no guarantees are given with the algorithm used, in most cases the tolerances are respected without severe conservativeness. Table 2 shows the complexity of the partition and accuracy of approximation when no heuristic is used (the hyper-rectangles are split on both axes into four hyper-rectangles rather than just two hyper-rectangles), for comparison. We observe that the use of heuristics reduce the complexity of the partition significantly. Figure 3 shows the partitions for the case $N = 10$ and relative tolerance $\sqrt{\bar{\varepsilon}_r} = 0.5$, with and without the use of heuristics.

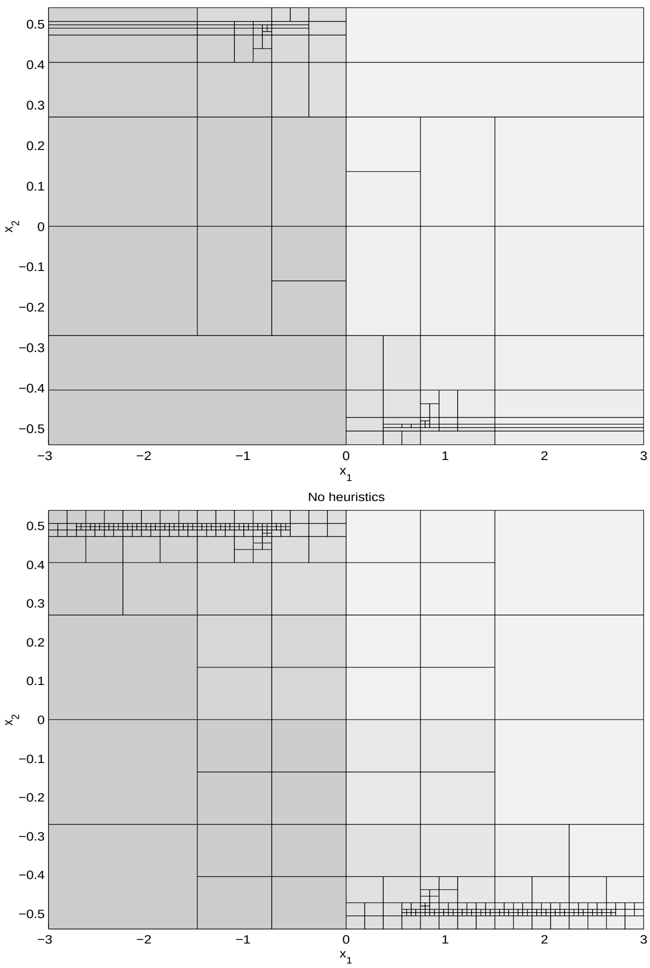

Figure 3: State space partition generated by the algorithm on the double integrator, with $N = 10$ and $\sqrt{\bar{\varepsilon}_r} = 0.5$. The upper partition is with the use of heuristics, and the lower partition is without the use of heuristics.

It is interesting to compare the structure of the partitions of the approximate PWL explicit MPC feedback laws with the partitions of the exact PWL explicit MPC feedback law, as shown in Figure 4 for the case of horizon $N = 10$. In parts of the state space where the exact partition contains several smaller regions while the approximate partition contains only a few large regions, the explanation is that the approximate approach only considers the first sample of the control input and is able to reduce complexity. In parts of the state space where the opposite is true, i.e. the approximate partition is more complex, this is due to a structural mismatch because the orthogonality of the hyperplanes of the approximate partition is enforced.

The exact partition in Figure 4 contains 191 polyhedral regions and is thus of comparable complexity to the approximate partitions. Still, it is clear that there will be significantly higher demand for real-time processing capacity and computer memory, since all hyperplanes in the partition are different and they are not orthogonal. This also holds if a

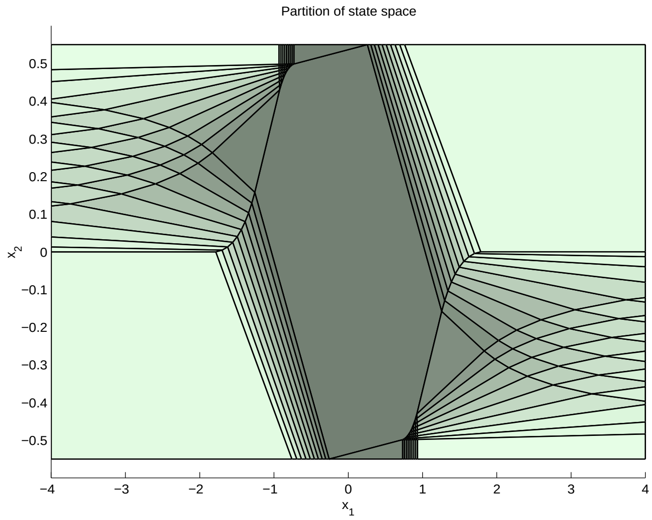

Figure 4: Exact polyhedral partition of the state space, $N = 10$.

search tree is constructed from the exact partition as proposed in [15]. In this case there will be 9 levels in the tree and 60 arithmetic operations are required to compute the exact solution, while about 3150 numbers must be stored in real-time computer memory. With the suggested approach, 18 arithmetic operations are sufficient, and only about 700 numbers must be stored for the partition with 97 regions ($N = 10$, $\sqrt{\bar{\varepsilon}_r} = 0.5$). Of course, the price to be paid for this complexity reduction is an approximation error.

As in [9] we remark that there is a significant difference between the exact and approximate approaches when the complexity of the partition is viewed as a function of the horizon N. While the number of regions with the exact approach seems to give a very rapid growth with N, [6], the approximate approach gives a partition complexity that is almost independent of the horizon N. One reason for this is that in the approximate approach it is taken into account that we only need the first sample of the input trajectory in order to implement the MPC.

Figure 5 shows a simulation comparing the approximate and exact solutions. We notice that the control inputs computed by exact and approximate controller are almost indistinguishable.

5 Conclusions

An algorithm for approximate implementation of linear MPC in the form of an explicit PWL state feedback is described. The approximate implementation has the advantage (compared to the exact PWL state feedback, [1, 6]) that it imposes a search tree structure on an orthogonal partition. This leads to reduced real-time computational complexity compared to the exact approach.

References

[1] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.

[2] 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.

[3] T. A. Johansen, I. Petersen, and O. Slupphaug, "On explicit suboptimal LQR with state and input constraints," in *Proc. IEEE Conf. Decision and Control, Sydney*, 2000, pp. TuM05–6.

[4] M. Seron, J. A. De Dona, and G. C. Goodwin, "Global analytical model predictive control with input constraints," in *Proc. IEEE Conf. Decision and Control, Sydney*, 2000, pp. TuA05–2.

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

[6] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," in *Proc. IEEE Conf. Decision and Control, Orlando*, 2001, pp. TuP11–4.

[7] T. A. Johansen, I. Petersen, and O. Slupphaug, "Explicit suboptimal linear quadratic regulation with input and state constraints," *Automatica*, vol. 38, July, accepted 2002.

[8] A. Bemporad and C. Filippi, "Suboptimal explicit MPC via approximate quadratic programming," in *Proc. IEEE Conf. Decision and Control, Orlando*, 2001, pp. FrP08–5.

[9] T. A. Johansen and A. Grancharova, "Approximate explicit model predictive control implemented via orthogonal search tree partitioning," in *Preprints, IFAC World Congress, Barcelona*, 2002, http://www.itk.ntnu.no/ansatte/Johansen_Tor.Arne/mpqp-tree-ifac.pdf.

[10] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, "Efficient on-line computation of explicit model predictive control," in *Proc. IEEE Conf. Decision and Control, Orlando*, 2001, pp. TuP11–2.

[11] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *IEEE Trans. Automatic Control*, vol. 45, pp. 1545–1549, 2000.

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

[13] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, pp. 509–517, 1975.

[14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry, 2nd edition*, Springer-Verlag, Berlin, 2000.

[15] P. Tøndel and T. A. Johansen, "Complexity reduction in explicit model predictive control," in *Preprints, IFAC World Congress, Barcelona*, 2002, http://www.itk.ntnu.no/ansatte/Johansen_Tor.Arne/mpqp-ifac-2002.pdf.

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

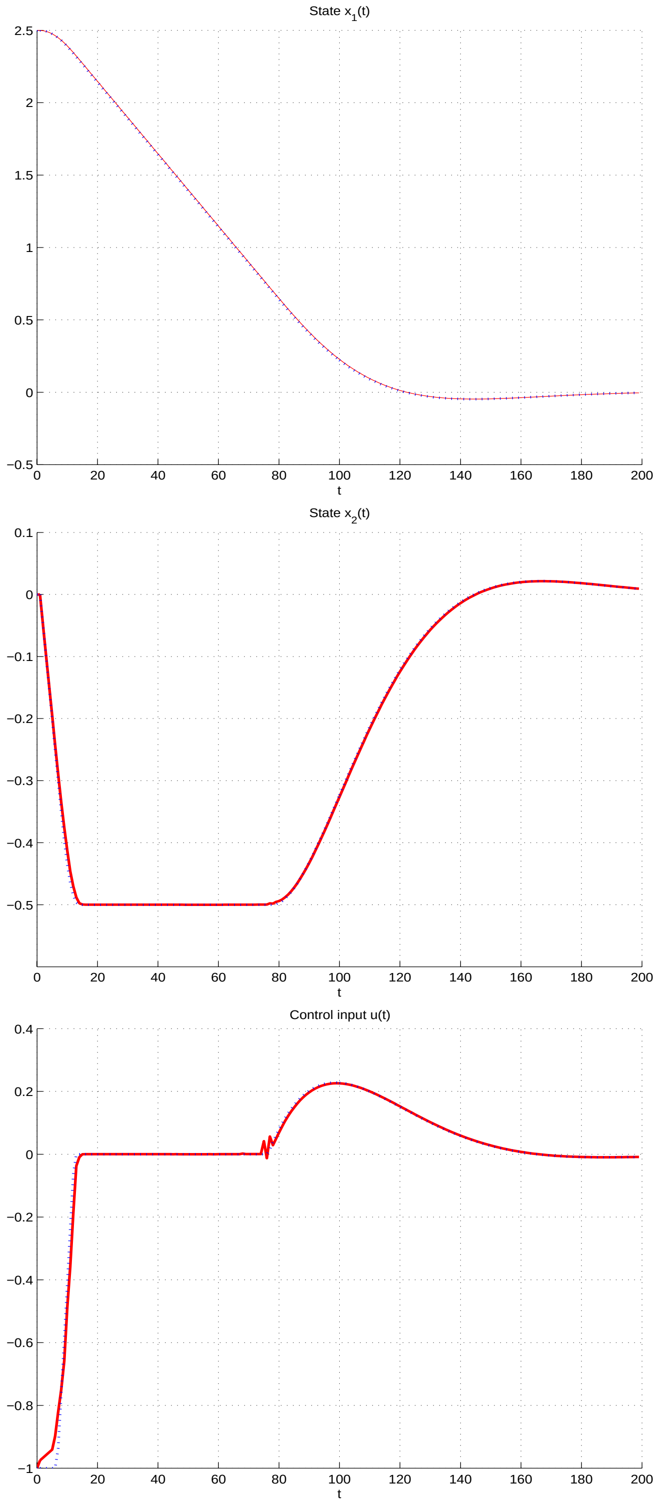

Figure 5: Simulation where the dashed curves are with the exact controller, and the solid curves are with the approximate controller, with $N = 10$ and $\sqrt{\bar{\varepsilon}_r} = 0.5$.