15th Triennial World Congress, Barcelona, Spain

APPROXIMATE EXPLICIT MODEL PREDICTIVE CONTROL IMPLEMENTED VIA ORTHOGONAL SEARCH TREE PARTITIONING

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

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

Abstract: Solutions to constrained linear model predictive control (MPC) problems can be pre-computed off-line in an explicit form as a piecewise linear (PWL) state feedback defined on a polyhedral partitioning of the state space. Even though real-time optimization is avoided, implementation of the PWL state feedback may still require a significant amount of computations. We suggest an algorithm that will determine an approximate explicit PWL state feedback solution by imposing an orthogonal search tree structure on the partition. This leads to efficient real-time computations and admits implementation at high sampling frequencies in embedded systems with inexpensive processors and low software complexity. The algorithm yields guarantees on the cost function error and constraint violations.

1. INTRODUCTION

The main motivation behind explicit model predictive control (MPC) is that an explicit state feedback law avoids the need for real-time optimization, and is therefore potentially useful for applications with fast sampling where MPC has not traditionally been used. In (Bemporad *et al.* 2002, Bemporad *et al.* 2000) 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. They show that the solution (the control input) has an explicit representation as a piecewise linear (PWL) function on a polyhedral partition of the state space and develop an mp-QP algorithm to compute this function. The solution can also be constructed by the algorithms (Johansen *et al.* 2000, Seron *et al.* 2000). In (Tøndel *et al.* 2001) a significantly more efficient mp-QP solver is developed by inferring additional information about neighboring regions during the iterative solution. The approach of (Johansen *et al.* 2000), starting with the Hamilton-Jacobi-Bellman equation for the optimal control problem, allows sub-optimality to be introduced by predetermining a small number of sampling instants when the active set is allowed to change on the horizon. The advantage of this is less regions in the polyhedral partition. A similar approach was suggested in (Tøndel and Johansen 2002) for mp-QP based explicit MPC. An alternative sub-optimal approach was introduced in (Bemporad and Filippi 2001) where small slacks are introduced on the optimality conditions and the mp-QP algorithm of Bemporad *et al.* (2000) is modified for the relaxed problem. This leads to reduced computational complexity and reduced complexity of the solution (in terms of less regions in the partition). Another method for reduction of computational complexity are proposed in (Borrelli *et al.* 2001).

Here we suggest a different approach to compute sub-optimal explicit MPC solutions. The idea is to require that the state space partition is represented as a search tree, i.e. to consist of orthogonal hypercubes organized in a hierarchical data-structure that allows extremely fast real-time search. The computational complexity with the suggested approach is logarithmic with respect to the number of regions, while a general polyhedral partitioning leads to a computational complexity that is linear with respect to the number of regions, if no additional data structures are built, cf. (Tøndel and Johansen 2002). The optimal solution is computed explicitly using quadratic programming (QP) only at the vertices of these hypercubes, and an approximate solution valid in the whole hypercube is computed based on this data. A hypercube is partitioned into two or more

⋆ This work was sponsored by the European Commission through the Research Training Network **MAC** (Multi Agent Control)
[1] Email: `Tor.Arne.Johansen@itk.ntnu.no`
[2] On leave from Institute of Control and System Research, Bulgarian Academy of Sciences, Acad. G. Bonchev str., Bl.2, P.O.Box 79, Sofia 1113, Bulgaria.

smaller hypercubes only if this is necessary to achieve the desired local accuracy of the solution. This makes the idea similar to storing the precomputed QP solutions at the various states in a multi-resolution lookup table.

Unlike any other method mentioned above, that all relies on the linearity of the problem to build polyhedral regions and a PWL solution, the suggested method is straightforward to extended to convex nonlinear constrained MPC problems by replacing the QPs with convex nonlinear programs. Other function approximation methods for optimal control are described in (Parisini and Zoppoli 1995, Parisini and Zoppoli 1996, Bertsekas and Tsitsiklis 1998).

2. EXPLICIT MPC AND EXACT MP-QP

Formulating a linear MPC problem as an mp-QP is briefly described below, see (Bemporad *et al.* 2002) for further details. Consider the linear system

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

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 pair. For the current $x(t)$, MPC solves the optimization problem

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

subject to $x_{t|t} = x(t)$ and

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

with the cost function given by

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

and symmetric $R > 0$, $Q \ge 0$, $P > 0$. 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 $k \ge N$ this corresponds to an infinite horizon LQ criterion (Chmielewski and Manousiouthakis 1996). This and related problems can by completing squares be reformulated as

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

$$\text{subject to} \quad Gz \le W + Sx \tag{6}$$

where $z \triangleq U + H^{-1}F^T x$. Note 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 are included, (Bemporad *et al.* 2002). 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 $z^* = z^*(x)$, Bemporad *et al.* (2000):

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 parameter space, and $V_z(x)$ is a convex (and therefore continuous) piecewise quadratic function.

□

As shown in (Bemporad *et al.* 2000), the mp-QP problem (5) - (6) can be solved by applying the Karush-Kuhn-Tucker (KKT) conditions

$$Hz + G^T\lambda = 0 \tag{7}$$

$$\text{diag}(\lambda)\,(Gz - W - Sx) = 0 \tag{8}$$

$$\lambda \ge 0 \tag{9}$$

$$Gz - W - Sx \le 0 \tag{10}$$

with $\lambda \in \mathbb{R}^q$. Since H has full rank, (7) gives

$$z = -H^{-1}G^T\lambda \tag{11}$$

Assume for the moment that we know which constraints are active at the optimum for a given x. Let $\tilde{\lambda}$ be the Lagrange multipliers of the active constraints, $\tilde{\lambda} \ge 0$. We can now form matrices $\tilde{G}, \tilde{W}$ and $\tilde{S}$ which contains the rows of G, W and S corresponding to the active constraints. Assume that $\tilde{G}$ has full row rank, such that the rows of $\tilde{G}$ are linearly independent (see (Tøndel *et al.* 2001) and (Bemporad *et al.* 2002) for details on how to handle degenerate situations when this is violated). For the active constraints, (8) and (11) gives $-\tilde{G}H^{-1}\tilde{G}^T\tilde{\lambda} - \tilde{W} - \tilde{S}x = 0$, which leads to

$$\tilde{\lambda} = -(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x). \tag{12}$$

Eq. (12) can now be substituted into (11) to obtain

$$z = H^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x). \tag{13}$$

We have now characterized the solution to (5)-(6) for a given optimal active set, and a fixed x. However, as long as the active set remains optimal (in a neighborhood of x), the solution (13) remains optimal, when z is viewed as a function of x. Next, we characterize the region where this active set remains optimal. First, z must remain feasible (10)

$$GH^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \le W + Sx. \tag{14}$$

Second, the Lagrange multipliers λ must remain non-negative (9)

$$-(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \ge 0. \tag{15}$$

The inequalities (14) and (15) 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. This region is a

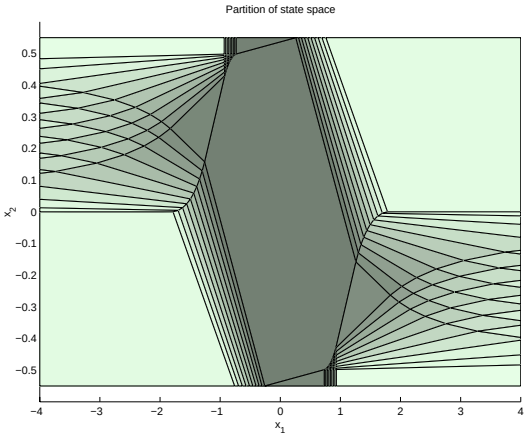

Fig. 1. Polyhedral partition of state space, $N = 10$.

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 are given by (Bemporad *et al.* 2000, Tøndel *et al.* 2001).

Example. Consider the double integrator (Johansen *et al.* 2000)

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

where the sampling interval $T_s = 0.05$, and consider the MPC problem with cost matrices $Q = \mathrm{diag}(1, 0)$, $R = 1$, and the matrix $P > 0$ given as the solution of the algebraic Riccati equation. The constraints are $-0.5 \leq x_2 \leq 0.5$ and $-1 \leq u \leq 1$. Figure 1 shows the partition for horizon $N = 10$ corresponding to the exact solution provided by the algorithm (Tøndel *et al.* 2001). We observe that the exact solution is fairly complex, containing 191 polyhedral critical regions, many of them of very small volume.

3. ERROR BOUNDS

When constructing approximate solutions it is useful to be able to compute bounds on the approximation error. We consider any approximate solution $\hat{z}_0(x)$ defined on an arbitrary polyhedron $X_0 \subset \mathbb{R}^n$. The corresponding cost is given by

$$\hat{V}_z(x) = \frac{1}{2}\hat{z}_0^T(x) H \hat{z}_0(x) \tag{16}$$

Assume we know the exact solution only at the vertices of X_0, and need to compute uniform bounds on the error $\hat{V}_z(x) - V_z^*(x)$ (for all $x \in X_0$). Since no assumption has so far been made on the feasibility of $\hat{z}_0(x)$, we cannot conclude that $\hat{V}_z(x)$ itself is an upper bound on $V_z^*(x)$. Instead, we proceed similar to Fiacco (1983) and take advantage of the convexity of V_z^* to derive upper and lower bounds on V^*. Let the polyhedron X_0 be represented by its vertices $\mathcal{V} = \{v_1, v_2, ..., v_M\}$. Define the affine function $\overline{L}(x) = \overline{L}_0 x + \overline{l}_0$ as the solution to the following LP:

$$\min_{\overline{L}_0, \overline{l}_0} \left(\overline{L}_0 v + \overline{l}_0 \right) \tag{17}$$

subject to

$$\overline{L}_0 v_i + \overline{l}_0 \geq V_z^*(v_i), \text{ for all } i \in \{1, 2, ..., M\} \tag{18}$$

Likewise, define the affine function

$$\underline{L}(x) = V_z^*(v) + \nabla^T V_z^*(v)(x - v) \;=\; \underline{L}_0 x + \underline{l}_0$$

where $v \in X_0$ is arbitrary. If V_z^* is not differentiable at v, $\nabla V_z^*(v)$ is taken as any sub-gradient. We observe that both $\overline{L}$ and $\underline{L}$ can be defined using only information computed from the solutions of the QPs at the points in $\mathcal{V} \cup \{v\}$. Typically, we choose v near the center of X_0 to minimize conservativeness.

Theorem 2. For all $x \in X_0$ the following inequalities hold

$$\underline{V}(x) \triangleq \underline{L}(x) + \frac{1}{2}x^T P x \leq V^*(x) \leq \overline{L}(x) + \frac{1}{2}x^T P x \triangleq \overline{V}(x) \tag{19}$$

Proof. The upper bound is a consequence of the convexity of V_z^*, cf. Theorem 1, combined with the fact $V^*(x) = V_z^*(x) + \frac{1}{2}x^T P x$. To see this, let $x \in X_0$ be arbitrary, and consider the convex combination $x = \sum_i \alpha_i v_i$ where $\alpha_i \geq 0$ satisfies $\sum_i \alpha_i = 1$:

$$\begin{aligned} V_z^*(x) &\leq \sum_{i=1}^{M} \alpha_i V_z^*(v_i) \\ &\leq \sum_{i=1}^{M} \alpha_i \left(\overline{L}_0 v_i + \overline{l}_0 \right) \\ &= \overline{L}_0 x + \overline{l}_0 \end{aligned}$$

The lower bound is derived as follows:

$$V_z^*(x) \geq V_z^*(v) + \nabla^T V_z^*(v)(x - v) \tag{20}$$

as a consequence of the convexity of V_z^*.

□

It follows that $-\varepsilon_1 \leq V^*(x) - \hat{V}(x) \leq \varepsilon_2$ where

$$\varepsilon_2 = \max_{x \in X_0} \left(\overline{V}(x) - \hat{V}(x) \right) \tag{21}$$

$$\varepsilon_1 = \max_{x \in X_0} \left(\hat{V}(x) - \underline{V}(x) \right) \tag{22}$$

Hence, ε_1 and ε_2 can be computed by solving two QPs.

4. APPROXIMATE MP-QP ALGORITHM

We restrict our attention to a hypercube $X \subset \mathbb{R}^n$ 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 search tree (generalized quad-tree or oct-tree, (de Berg *et al.* 2000)). Then the search complexity is logarithmic with respect to the number of regions. The orthogonal search tree is a hierarchical data structure where a hypercube can be sub-divided into smaller hybercubes allowing the local resolution to be adapted, cf. Figure 2. When searching the tree, only n scalar comparisons are required at each level.

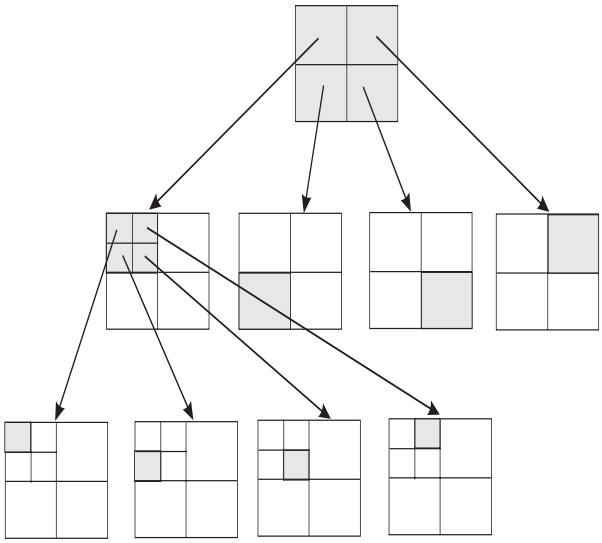

Fig. 2. Quadtree partition of a rectangular region in a 2-dimensional state-space.

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 hypercube X_0, by solving up to 2^n QPs. Based on these solutions (13), written in the compact notation

$$z_i^0(x) = K_i^0 x + g_i^0 \tag{23}$$

we compute a local approximation $\hat{z}_0(x)$ to the PWL optimal solution $z^*(x)$, restricted to the hypercube X_0. As described above, the linear solution (23) is exact in some polyhedral critical region containing the vertex v_i. A simple approximation is the linear approximation

$$\hat{z}_0(x) = K_0 x + g_0 \tag{24}$$

using averaged gain matrices

$$K_0 = \frac{1}{2^n}\sum_{i=1}^{2^n} K_i^0, \quad g_0 = \frac{1}{2^n}\sum_{i=1}^{2^n} g_i^0$$

Based on this information we are able to compute the error bound

$$\varepsilon = \max(\varepsilon_1, \varepsilon_2) \tag{25}$$

from (21) and (22). In addition we need to consider violation of the constraints as measured by the vector $\tilde{\delta}(x)$ defined by

$$\tilde{\delta}(x) = D(G(K_0 x + g_0) - W - Sx) \tag{26}$$

where D is a diagonal scaling matrix with positive elements. The maximum violation of each constraint within a polyhedron X_0 is computed by solving the set of LPs

$$\delta_j = \max_{x \in X_0} \tilde{\delta}_j(x) \tag{27}$$

for $j = 1, 2, ..., q$. Now, if the cost function error ε is smaller than some prescribed tolerance $\overline{\varepsilon}$ and the constraint violations δ_j are smaller than some prescribed tolerance $\overline{\delta}_j$, no further refinement of the region X_0 is needed. Otherwise, we partition X_0 into 2^n equal-sized hypercubes, and repeat the procedure described above for each of these. This algorithm can be summarized as follows.

Algorithm 1 (approximate mp-QP)

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

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

3. Compute the solution to the QP (5)-(6) for x fixed to each of the 2^n vertices of the hypercube X_0 and its center point (some of these QPs may have been solved in earlier steps). From the optimal active set at each solution, compute the local linear optimal solution (13).

4. From the local linear optimal solutions at the vertices of the hypercube, compute a continuous local linear state feedback (24) as an approximation to be used in the hypercube X_0.

5. Determine if the hypercube needs to be split in order to reduce the cost function approximation error bound ε or the constraint violations bound δ. If so, go to step 6. Otherwise, mark X_0 explored and go to step 2.

6. Split the hypercube X_0 into hypercubes X_1, X_2, ..., X_{2^n}. Mark them all unexplored, remove X_0 from $\mathcal{P}$, add X_1, X_2, ..., X_{2^n} to $\mathcal{P}$ and go to step 2.

□

This algorithm will terminate with a piecewise continuous and PWL function that is an approximation to the continuous PWL exact solution.

Theorem 3. Consider the mp-QP problem (5)-(6) defined on a hybercube X. Suppose $D > 0$ and $\overline{\varepsilon}, \overline{\delta} > 0$ are given and we require in Step 5 of Algorithm 1 that $\varepsilon \leq \overline{\varepsilon}$ and $\delta \leq \overline{\delta}$ (elementwise). Then Algorithm 1 terminates after a finite number of steps with an approximate solution $\hat{z}(x)$ and associated cost $\hat{V}(x)$ that satisfies

$$\sup_{x \in X} |\hat{V}(x) - V^*(x)| \leq \overline{\varepsilon} \tag{28}$$

and constraints satisfying

$$\sup_{x \in X} D\left(G\hat{z}(x) - W - Sx\right) \leq \overline{\delta} \tag{29}$$

Proof. The bounds (28) and (29) follows from Theorem 2 due to step 5 of the algorithm that ensures that the algorithm will not terminate before the cost and constraint errors respect their bounds in all hypercubes of the partition.

The algorithm terminates after a finite number of steps because the optimal cost V^* is continuous and can be uniformly approximated to arbitrary accuracy by a sufficiently large finite number of regions. Due to this regularity, the bound on the error is reduced by some fraction at each step due to the quad-tree splitting into equal-sized hypercubes. A similar argument can be used also on the constraint violations since $\overline{\delta}$ is strictly positive.

□

It might be useful if we construct the local linear approximation of the control law such that feasibility is guaranteed (only optimality is relaxed) rather than using the averaging (24). As shown in (Bemporad and Filippi 2001), one may solve a QP to compute feasible approximate local linear state feedbacks. Then $\hat{V}_z$ is an upper bound on V_z^*.

An advantage of the present method, compared to (Bemporad and Filippi 2001), is that a posteriori analysis of the approximation error is in general not needed.

Improved accuracy might be achieved if some interpolation scheme is used instead of the averaging (24), at the cost of increased real-time computational complexity and local non-linearity.

The method can be easily adapted to other spatial data structures such as k-d-trees and others (de Berg *et al.* 2000). This would be expected to lead to less regions, especially for larger n.

A variant of the algorithm may include partitions where a small number of candidate active sets must be compared in real time for each given hypercube. This allows the real-time computer memory requirements to be reduced at the cost of increased real-time computations.

When there is noise or uncertainty on the state, this imposes a limitation on how small regions are necessary. In such cases one may impose a tolerance that prevents the algorithm from generating such small regions.

Recognition of the same solution in neighboring hypercubes that can be combined are easily done, as such hypercubes would be leaf-nodes with the same parent node in the tree. We recommend this is implemented as a post-processing step in order to take into account that only the first m elements of the solution z^* are required for the MPC implementation. Likewise, a useful heuristic is to only require approximate feasibility at the first sample.

5. EXAMPLE

Consider the double integrator example introduced above. With $\overline{\varepsilon} = 0.4$ and $\overline{\delta}_u = 0.2$ for input constraints and $\overline{\delta}_x = 0.005$ for state constraints, Algorithm 1 gives the quad-tree partition in Figure 3 with 412 regions for $N = 10$. Figure 4 shows a typical trajectory with the exact and approximate approach, both starting from the same initial state. We observe that the discrepancy is small. Table 1 summarizes the properties of the approximate approach compared to the exact approach, as a function of the horizon N, while Table 2 illustrates how the approximate solution depends on the tolerance parameters $\overline{\epsilon}, \overline{\delta}_u, \overline{\delta}_x$. We observe that with the exact approach the number of regions grow rapidly with the horizon N, while with the approximate approach the number of regions is fairly independent of the horizon N. This is to be expected since the regularity (which essentially determines the difficulty of approximation) of the controller mapping $x \mapsto u$ is fairly independent of N. For most N there are 7 levels in the quad-tree. With two scalar comparisons required at each level, a total of 14 scalar arithmetic operations are required in the worst case to determine which region the state belongs to, which is impossible to achieve with the exact approach.

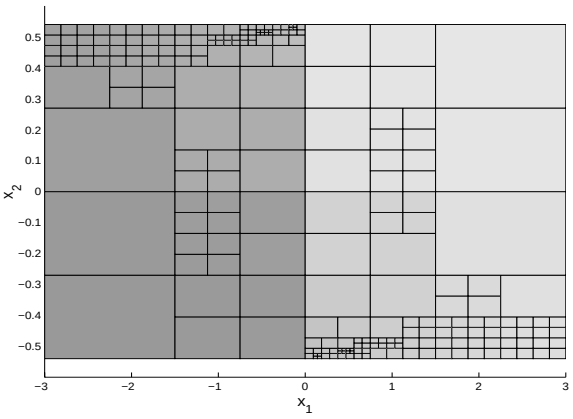

Fig. 3. Partition for double integrator, $N = 10$.

Table 1. Characteristics of approximate and exact explicit MPC solutions for the double integrator example as a function of the horizon N.

N	Exact regions	Approx. regions	Ave. u error	Max u error
1	5	-	-	-
2	13	232	0.029	0.67
3	23	280	0.032	0.72
4	35	322	0.017	0.35
5	51	334	0.017	0.35
6	71	394	0.013	0.29
7	95	400	0.013	0.30
8	123	394	0.014	0.31
9	155	406	0.012	0.31
10	191	412	0.014	0.37
11	231	394	0.013	0.31
12	277	394	0.013	0.33
13	325	388	0.012	0.34
14	379	394	0.012	0.34
15	437	394	0.012	0.31

Table 2. Characteristics of approximate explicit solutions for the double integrator example as a function of the tolerance parameters.

$(\overline{\epsilon}, \overline{\delta}_u, \overline{\delta}_x)$	Num. regions	Ave. error	Max error
(0.8,0.3,0.01)	232	0.024	0.67
(0.4,0.2,0.005)	412	0.014	0.37
(0.2,0.1,0.0025)	898	0.008	0.34
(0.1,0.05,0.001)	1570	0.004	0.35

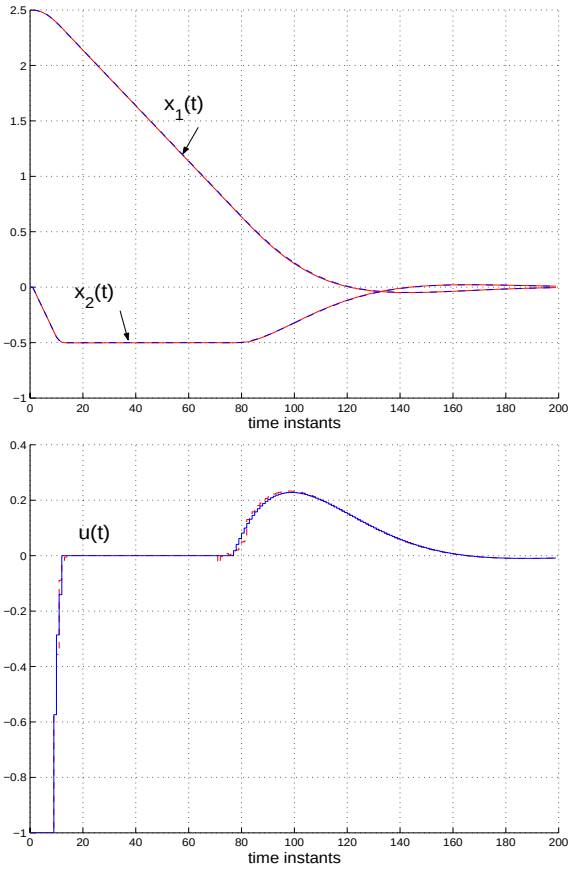

Fig. 4. The solid and dashed curves show an exact and approximate trajectory, respectively, for the double integrator example.

The real-time computer memory requirements are similar for the two cases, while the off-line computation time may be larger in the approximate case depending on the required accuracy. Thus, the main advantage of the approximate approach is that is admits a highly efficient real-time implementation based on a search tree. We note that although the specification allowed some input and state constraint violations, significant violations are only present in very small parts of the state space.

Like the exact solution, the complexity of the approximate solution typically increases exponentially with n (the order to the system). The complexity of the approximate solution also typically increases exponentially with respect to the required accuracy. However, while the complexity of the exact solution increases exponentially with the horizon N, the present approximate solution seems to be fairly independent of N.

6. CONCLUSIONS

An algorithm for off-line computation of approximate explicit solutions to linear constrained MPC problems is described. The algorithm allows tolerances on the approximation error and constraint violations to be specified, and guarantees that these tolerances are not violated. The resulting explicit piecewise linear state feedback is defined on an orthogonal partition of the state space that allows very efficient real-time computations through a search tree.

7. REFERENCES

Bemporad, A. and C. Filippi (2001). Suboptimal explicit MPC via approximate quadratic programming. In: *Proc. IEEE Conf. Decision and Control, Orlando*. pp. FrP08–5.

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.

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

Bertsekas, D. P. and J. N. Tsitsiklis (1998). *Neuro-dynamic Programming*. Athena Scientific, Belmont.

Borrelli, F., M. Baotic, A. Bemporad and M. Morari (2001). Efficient on-line computation of explicit model predictive control. In: *Proc. IEEE Conf. Decision and Control, Orlando*. pp. TuP11–2.

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

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

Fiacco, A. V. (1983). *Introduction to sensitivity and stability analysis in nonlinear programming*. Orlando, Fl: Academic Press.

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

Parisini, T. and R. Zoppoli (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica* **31**, 1443–1451.

Parisini, T. and R. Zoppoli (1996). Neural approximations for multistage optimal control of nonlinear stochastic systems. *IEEE Trans. on Automatic Control* **41**, 889–895.

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.

Tøndel, P. and T. A. Johansen (2002). Complexity reduction in explicit model predictive control. In: *Preprints, IFAC World Congress, Barcelona*.

Tøndel, P., T. A. Johansen and A. Bemporad (2001). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In: *Proc. IEEE Conf. Decision and Control, Orlando*. pp. TuP11–4.