Approximate explicit constrained linear model predictive control via orthogonal search tree

Tor A. Johansen and Alexandra Grancharova

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

Abstract

Solutions to constrained linear model predictive control problems can be pre-computed off-line in an explicit form as a piecewise linear state feedback on a polyhedral partition of the state space, avoiding real-time optimization. We suggest an algorithm that will determine an approximate explicit piecewise linear state feedback by imposing an orthogonal search tree structure on the partition. This leads to a real-time computational complexity that is logarithmic in the number of regions in the partition, and the algorithm yields guarantees on the sub-optimality, asymptotic stability and constraint fulfillment.

I. 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 applications with fast sampling where MPC has traditionally not 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. They show that the solution (the control input) has an explicit representation as a piecewise linear (PWL) function and develop an mp-QP algorithm to compute this function, see also the algorithms [3], [4]. The approaches of [5], [6], [7], allows sub-optimality to be introduced by pre-determining a small number of sampling instants when the active set or input is allowed to change on the horizon, leading to less regions in the polyhedral partition. An alternative sub-optimal approach was introduced in [8] where small slacks are introduced on the optimality conditions and the mp-QP algorithm in [2] is modified for the relaxed problem. This leads to reduced computational complexity and reduced complexity of the solution, in terms

This work was sponsored by the European Commission through the Research Training Network **MAC** (Multi Agent Control)
Corresponding author: `Tor.Arne.Johansen@itk.ntnu.no`

of less regions in the state space partition.

Here we suggest an entirely different approach to compute sub-optimal explicit MPC solutions. The idea is to require that the state space partition is represented by a search tree. Hence, the partition consists of orthogonal hypercubes organized in a hierarchical data-structure that allows extremely fast real-time search. 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 smaller hypercubes only if this is necessary to achieve the desired accuracy. The real-time 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 [7]. It must be stressed that the advantage of this approach is the efficient real-time computations rather than the off-line computations. Unlike any other method mentioned above, that all rely 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 [9], [10], [11].

II. Explicit MPC and exact mp-QP

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, $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$. For the current $x(t)$, a typical MPC algorithm, see [12] for an overview, 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 $x_{t|t} = x(t)$ and

$$\begin{aligned}
& y_{\min} \leq y_{t+k|t} \leq y_{\max}, \quad k = 1, ..., N \\
& u_{\min} \leq u_{t+k} \leq u_{\max}, \quad k = 0, 1, ..., N-1, \\
& x_{t+N|t} \in \Omega \\
& x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k}, \quad k = 0, 1, ..., N-1 \\
& y_{t+k|t} = Cx_{t+k|t}, \quad k = 1, 2, ..., N
\end{aligned} \tag{3}$$

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 $R \succ 0$ (positive definite), $Q \succeq 0$ (positive semi-definite). We assume (A, B) is controllable, $(A, \sqrt{Q})$ is observable, Ω is a polyhedral terminal set, and the final cost matrix $P \succ 0$ is the solution of the associated algebraic Riccati equation. With the assumption that no constraints are active for $k \geq N$, (4) corresponds to an infinite horizon LQ criterion [13]. It is also assumed $u_{max} > 0$, $u_{min} < 0$, $y_{max} > 0$, and $y_{min} < 0$ such that the origin is an interior point in the feasible set $X_f \subseteq \mathbb{R}^n$. The optimal solution is denoted $U^* = (u_t^{*T}, u_{t+1}^{*\ T}, ..., u_{t+N-1}^{*\ T})^T$, and the control input is chosen according to the receding horizon policy $u(t) = u_t^*$. Problem (2)-(3) and similar problems can by algebraic manipulations be reformulated as

$$V_z^*(x) = \min_z \frac{1}{2} z^T H z, \quad \text{subject to} \ \ Gz \leq W + Sx \tag{5}$$

where $z = U + H^{-1}F^T x$, and the matrices are defined in [1]. Notice that $H \succ 0$ since $R \succ 0$, [1]. The vector x is the current state, which can be treated as a vector of parameters. 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 $n_z = mN$. The problem (5) defines an mp-QP, since it is a QP in z parameterized by x. In parametric programming problems one seek the solution z^* as an explicit function of the parameters x. For the mp-QP (5), the solution $z^*(x)$ has the following properties, [14], [1]:

Theorem 1. Consider the mp-QP (5) with $H \succ 0$. The solution $z^*(x)$ (and $U^*(x) = z^*(x) - H^{-1}F^T x$) is a continuous PWL function of x, and $V_z(x)$ is a convex and continuous piecewise quadratic function. □

The concept of active constraints is instrumental to characterize the PWL solution. An inequality constraint is said to be active for some x if it holds with equality at the optimum. An explicit representation of the optimal PWL state feedback is given in the following theorem [1]:

Theorem 2. Consider the mp-QP (5) with $H \succ 0$, and an arbitrary fixed set of active constraints, where the sub-matrices $\tilde{G}, \tilde{W}$ and $\tilde{S}$ contain the corresponding rows of G, W and S. If the rows of $\tilde{G}$ are linearly independent, the optimal solution and associated Lagrange multipliers are given by the affine functions

$$z_0^*(x) = H^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \tag{6}$$

$$\tilde{\lambda}_0(x) = -(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x) \tag{7}$$

Moreover, the critical region $CR_0 \subseteq \mathbb{R}^n$ where this solution is optimal is given by the polyhedron

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

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

□

General mp-QP algorithms based on Theorem 2 are given in [4] and [1], where it is also discussed how to handle situations when the linear independence condition is violated.

Example. Consider a discrete-time double integrator

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

where the sampling interval is $T_s = 0.3$, and consider the MPC problem with cost matrices $Q = \mathrm{diag}(1,0)$ and $R = 1$. The constraints in the system are $-0.5 \leq x_2 \leq 0.5$, and $-1 \leq u \leq 1$, and we restrict our attention to the set $X = [-2.8, 2.8] \times [-0.8, 0.8]$. Figure 1 shows polyhedral partition of the optimal state feedback for horizon $N = 20$ corresponding to the exact solution provided by the algorithm [4]. We observe that the exact solution is fairly complex, containing 205 polyhedral critical regions.

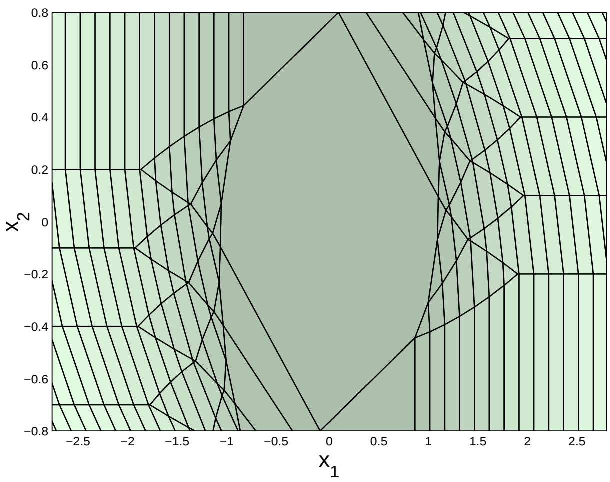

Fig. 1. Polyhedral partition of the state space for the optimal PWL state feedback controller of the double integrator with horizon $N = 20$.

III. Error bounds

When constructing approximate solutions it is useful to compute bounds on the approximation error. We consider any approximate affine solution $\hat{z}_0(x)$ defined on an arbitrary bounded polyhedron $X_0 \subset \mathbb{R}^n$. The corresponding sub-optimal cost is given by $\hat{V}_z(x) = \frac{1}{2}\hat{z}_0^T(x)H\hat{z}_0(x)$ for $x \in X_0$. Assume it is needed to compute a bound on the error $\hat{V}_z(x) - V_z^*(x)$, uniformly for all $x \in X_0$. We develop a method based on similar ideas as in [14]. Let the polyhedron X_0 be represented by its vertices $\mathcal{V} = \{v_1, v_2, ..., v_M\}$, and define the affine function $\overline{L}(x) = \overline{L}_0 x + \overline{l}_0$ as the solution to the following LP:

$$\min_{\overline{L}_0, \overline{l}_0} \left(\overline{L}_0 v + \overline{l}_0\right) \quad \text{subject to} \quad \overline{L}_0 v_i + \overline{l}_0 \geq V_z^*(v_i), \text{ for all } i \in \{1, 2, ..., M\} \tag{10}$$

Likewise, define the PWL function

$$\underline{L}(x) \quad = \quad \max_{i=1,2,...,M} V_z^*(v_i) + \nabla^T V_z^*(v_i)(x - v_i) \tag{11}$$

where $\nabla V_z^*(v)$ is taken as any sub-gradient if V_z^* is not differentiable at v. Furthermore, define $\underline{V}(x) = \underline{L}(x) + x^T P x$ and $\overline{V}(x) = \overline{L}(x) + x^T P x$. We observe that both $\overline{V}$ and $\underline{V}$ can be defined using only information computed from the solutions of the QP at the vertices $\mathcal{V}$.

Theorem 3. Let $X_0 \subset \mathbb{R}^n$ be a bounded polyhedron. Then $\underline{V}(x) \leq V^*(x) \leq \overline{V}(x)$ for all $x \in X_0$.

Proof. Notice that $V^*(x) = V_z^*(x) + x^T P x$, since $V_z^*(x) = 0$ is the optimal cost for x in the critical region where the unconstrained LQR feedback is optimal. The upper bound is a consequence of the convexity of V_z^*, cf. Theorem 1. 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 $\sum_i \alpha_i = 1$:

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

The lower bound is also derived as a direct consequence of the convexity of V_z^*, since for any $v \in X_0$ the following sub-gradient inequality holds, [15]: $V_z^*(x) \geq V_z^*(v) + \nabla^T V_z^*(v)(x - v)$. □

It follows that $-\varepsilon_1 \leq V^*(x) - \hat{V}(x) \leq \varepsilon_2$, where ε_1 and ε_2 can be computed by

$$\varepsilon_1 \quad = \quad \max_{x \in X_0} \left(\hat{V}(x) - \underline{V}(x)\right) \tag{12}$$

$$\varepsilon_2 \quad = \quad \max_{x \in X_0} \left(\overline{V}(x) - \hat{V}(x)\right) \tag{13}$$

The PWL lower bound can be replaced by a simpler affine lower bound $\underline{L}(x) = V_z^*(v) + \nabla^T V_z^*(v)(x - v)$, where $v \in X_0$ is arbitrary. Since X_0 is a hypercube, the solution of the optimization problems (12) and (13) then becomes particularly simple.

IV. Approximate mp-QP algorithm

Consider a hypercube $X \subset \mathbb{R}^n$ where we seek to approximate the optimal PWL solution $z^*(x)$ to the mp-QP (5). In order to keep the real-time computational complexity at a minimum, we require that the state space

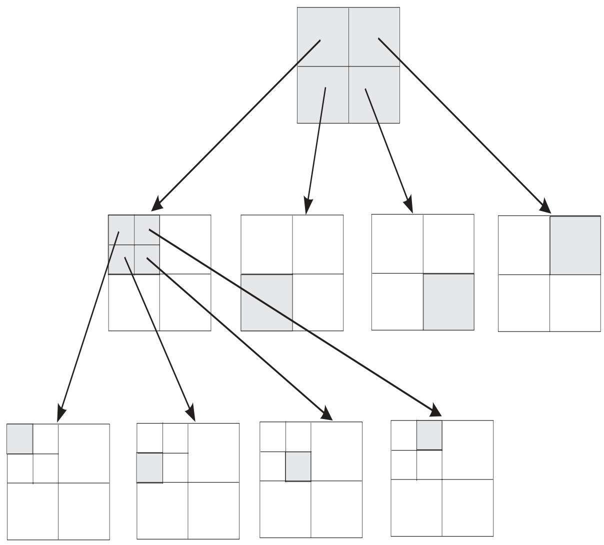

Fig. 2. Quadtree partition of a rectangular region in a 2-dimensional space.

partition is orthogonal and can be represented by a search tree (generalized quad-tree or oct-tree, [16]), such that the real-time 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. 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) at the 2^n vertices of the hypercube X_0, by solving up to 2^n QPs. Based on these solutions, we compute a feasible local approximation to the PWL optimal solution $z^*(x)$, restricted to the hypercube X_0, using the following result [8]:

Lemma 1. Consider the bounded polyhedron X_0 with vertices $\{v_1, v_2, ..., v_M\}$. If K_0 and g_0 solve the QP

$$\begin{aligned} \min_{K_0, g_0} \quad & \sum_{i=1}^{M} (z^*(v_i) - K_0 v_i - g_0)^T H (z^*(v_i) - K_0 v_i - g_0) \\ \text{subject to} \quad & G(K_0 v_i + g_0) \leq S v_i + W, \quad i \in \{1, 2, ..., M\} \end{aligned} \tag{14}$$

then the least squares approximation $\hat{z}_0(x) = K_0 x + g_0$ is feasible for the mp-QP (5) for all $x \in X_0$. □

Since $\hat{z}_0$ is feasible, $\hat{V}(x) = \hat{V}_z(x) + x^T P x$ is itself an upper bound on $V^*(x)$ such that for all $x \in X_0$

$$0 \;\leq\; \hat{V}(x) - V^*(x) \;\leq\; \varepsilon_1 \tag{15}$$

If the cost function error ε_1 is smaller than some prescribed tolerance $\overline{\varepsilon} > 0$, 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 procedure 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, go to step 7.

3. Solve the QP (5) for x fixed to each of the 2^n vertices of the hypercube X_0 (some of these QPs may have been solved in earlier steps).

4. Compute an affine state feedback $\hat{z}_0$ using Lemma 1, as an approximation to be used in X_0.

5. Compute the error bound ε_1, using Theorem 3 and (12). If $\varepsilon_1 \leq \bar{\varepsilon}$, mark X_0 as 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.

7. If necessary, split the hypercubes containing the origin such that $u^*(x) = Kx$ is optimal everywhere in these hypercubes, where K is the unconstrained LQR gain matrix.

□

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

Theorem 4. Algorithm 1 terminates after a finite number of steps with a feasible approximate solution $\hat{z}(x)$ and associated cost $\hat{V}(x)$ that satisfies $0 \leq \hat{V}(x) - V^*(x) \leq \bar{\varepsilon}$ for all $x \in X$.

Proof. The error bound follows from (15) due to step 5 of the algorithm that ensures that the algorithm will not terminate before the cost error is smaller than the tolerance $\bar{\varepsilon}$ 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 some $\hat{V}$ with a sufficiently large finite number of regions, such that the bound on the error is reduced by some minimum fraction at each step due to the quad-tree splitting into equal-sized hypercubes. □

An advantage of the present method, compared to [8], is that a posteriori analysis of the approximation error is not needed. Step 7 is mainly required to ensure that the solution is exact in a neighborhood of the origin, which proves useful when studying stability properties. It is worthwhile remarking that recognition of the same solution in neighboring hypercubes that can be combined into a larger hypercube is easily done, as such hypercubes would be all the 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 mN-dimensional solution $z^*(x)$ are required for the MPC implementation, as in the exact case [1].

V. Stability

Under some assumptions on the MPC tuning, the MPC solving (5) will make the origin asymptotically stable [12]. Based on a similar analysis as [17] we show below that these properties are inherited by the approximate MPC under some assumptions on the terminal set Ω and tolerance $\bar{\varepsilon}$. Suppose $X_f \subseteq X$, which is a polyhedral set [1]. Let $\Gamma \subseteq X_f$ be a hypercube where the solution computed by the approximate explicit MPC is $u^*(x) = Kx$, i.e. exactly the unconstrained LQR feedback. It is straightforward to show that Algorithm 1 leads to a non-empty Γ containing the origin in its interior, due to step 7. Let the terminal set Ω be the maximal output admissible set [18] for the linear system $x(t+1) = (A + BK)x(t)$ contained in the polyhedral set

$$\mathcal{F} \quad = \quad \{x \in \Gamma \mid u_{min} \leq Kx \leq u_{max},\ y_{min} \leq Cx \leq y_{max}\} \tag{16}$$

Ω is a polyhedron with a finite number of facets and can be easily computed, since $A + BK$ is Hurwitz and Γ is bounded because X is bounded [18].

Theorem 5. Consider the mp-QP problem (5) with $H \succ 0$ defined on a hybercube X such that $X_f \subseteq X$. Define $\Sigma = Q + K^T RK$, assume $\Sigma \succ 0$, and let γ be the largest positive number for which the ellipsoid $E = \{x \in X_f \mid x^T \Sigma x \leq \gamma\}$ is contained in Ω. Moreover, assume the tolerance $\bar{\varepsilon}$ satisfies

$$0 \ < \ \bar{\varepsilon} \ \ \leq \ \ \frac{\gamma + x_0^T \Sigma x_0}{2} \tag{17}$$

where $x_0 = \arg\min_{x \in X_0} x^T \Sigma x$. Then the approximate explicit MPC computed by Algorithm 1 in closed loop with the system (1) makes the origin asymptotically stable for all $x(0) \in X_f$, and the state and input trajectories are feasible.

Proof. Let $x(t) \in X_f$ be arbitrary. At time $t+1$ consider $\tilde{U} = ({u^*_{t+1}}^T, {u^*_{t+2}}^T, ..., {u^*_{t+N-1}}^T, (Kx^*_{t+N})^T)^T$, where $x^*_{t+k|t}$ is the state at time $t + k$ associated with U^*. Since U^* is feasible, $x^*_{t+N|t} \in \Omega$ and due to the way Ω is constructed it follows that $x^*_{t+N+1|t} = (A + BK)x^*_{t+N|t} \in \Omega \subseteq X_f$. Hence, $\tilde{U}$ is feasible and the trajectories remain feasible since Ω is a positively invariant set [18]. It follows that X_f is a positively invariant set. Since $\hat{V}(x)$ is an upper bound on $V^*(x)$, standard arguments of dynamic programming, as in [17], show that along the trajectories of the sub-optimal closed loop dynamics

$$V^*(x(t+1)) - V^*(x(t)) \ \leq \ \hat{V}(x(t+1)) - V^*(x(t)) \tag{18}$$

$$= \ \hat{V}(x(t)) - x^T(t)Qx(t) - {u^*}^T(t)Ru^*(t) - V^*(x(t)) \tag{19}$$

For $x(t) \in \Omega$ it is clear that $\hat{V}(x(t)) = V^*(x(t))$ and

$$V^*(x(t+1)) - V^*(x(t)) \leq -x^T(t)\Sigma x(t) \tag{20}$$

For $x(t) \notin \Omega$ we have $x^T(t)\Sigma x(t) > \gamma$ such that (15) gives

$$V^*(x(t+1)) - V^*(x(t)) \leq \bar{\varepsilon} - x^T(t)\Sigma x(t) < 0 \tag{21}$$

Since V^* is positive definite with $V^*(0) = 0$, and radially unbounded, it is suited as a Lyapunov-function candidate. From LaSalle's invariance principle, $x(t) \to \Omega$ as $t \to \infty$, and the origin is asymptotically stable with region of attraction equal to the positively invariant set X_f. □

Notice that the tolerance $\bar{\varepsilon}$ that is sufficient for stability can be computed a priori from (17) and the control specification. It is clear that it makes sense to let $\bar{\varepsilon}$ depend on X_0 through x_0, since this bound is used in Algorithm 1 only in the context of a fixed given X_0.

VI. Example

TABLE I

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

Horizon N	Exact regions	Approx. regions	Error Δu_{ave}	Error Δu_{max}	Exact offline CPU time (s)	Approx. offline CPU time (s)
2	13	82	0.018	0.18	0.2	1.5
4	41	118	0.011	0.17	0.6	2.9
6	75	124	0.011	0.17	1.3	4.1
8	111	124	0.010	0.17	2.3	6.5
10	139	136	0.010	0.17	3.4	9.8
12	161	142	0.010	0.16	4.5	12.3
14	181	148	0.010	0.15	5.8	19.0
16	195	158	0.011	0.17	7.1	23.4
18	203	160	0.010	0.16	8.4	27.2
20	205	148	0.011	0.17	9.7	30.8

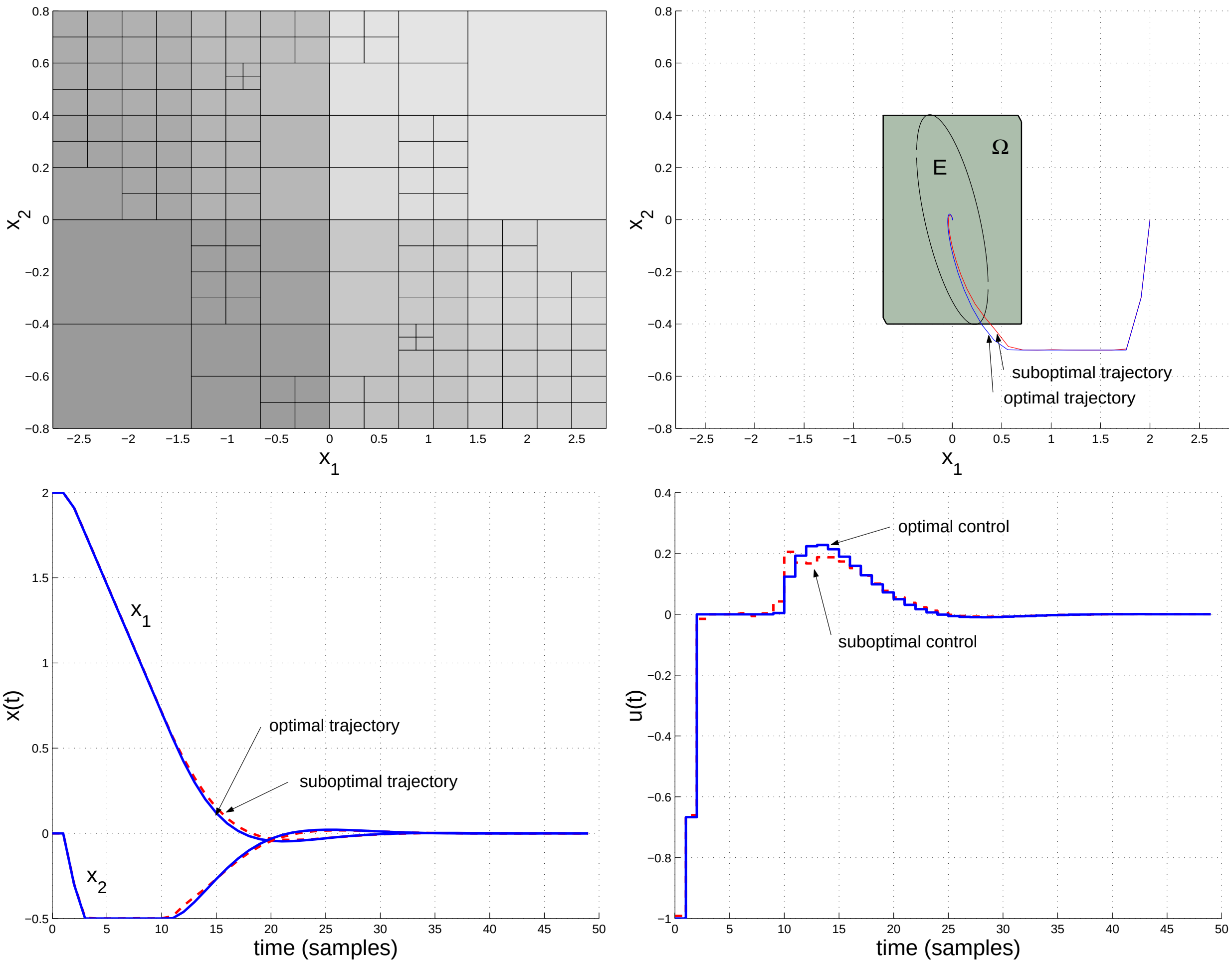

Fig. 3. Partition for double integrator with $N = 20$ (upper left) and the associated sets E and Ω (upper right). The solid and dashed curves show an exact and approximate trajectories, respectively (lower left), and the input (lower right).

Consider the double integrator example introduced above. The tolerance on the approximation error is chosen according to (17). Algorithm 1 gives the quad-tree partition in Figure 3 with 148 regions for $N = 20$. $\Gamma = [-0.7, 0.7] \times [-0.4, 0.4]$, and the sets Ω and E are also shown in Figure 3, as well as a typical trajectory with the exact and approximate approaches, both starting from the same initial state $x(0) = (2, 0)^T$. We observe that the discrepancy is fairly small. Table I summarizes the properties of the approximate approach compared to the exact approach, as a function of the horizon N. Δu_{ave} and Δu_{max} are the average and maximum values of the error $||u^*(x) - \hat{u}(x)||_2$ in X. We observe that with the exact approach the number of regions grows more rapidly with the horizon N than with the approximate approach. This is to be expected since the difficulty of approximation of the exact controller mapping $u^*(x)$ is fairly independent of N. For most N there are 5 levels in the quad-tree. With two scalar comparisons required at each level, a total of 10

TABLE II

Characteristics of approximate explicit solutions for the double integrator example as a function of the relative tolerance.

Tolerance	Regions	Δu_{ave}	Δu_{max}
100 %	148	0.0108	0.124
30 %	364	0.0072	0.088
10 %	780	0.0045	0.051
3 %	1722	0.0029	0.039

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. The real-time computer memory requirements are similar for the two cases, while the off-line computation time is typically larger in the approximate algorithm depending on the required accuracy (the offline CPU times in Table I are with the algorithm in [4] which is significantly more efficient than [1]). Thus, the main advantage of the approximate approach is that is admits a highly efficient real-time implementation based on a search tree. Table II illustrates how the approximate solution depends on the tolerance $\bar{\varepsilon}$. The tolerance in the leftmost column is relative to the tolerance (17).

VII. Conclusions

An algorithm for off-line computation of approximate explicit solutions to linear constrained MPC problems is described and analyzed. The algorithm allows a tolerance on the cost function approximation error to be specified, and guarantees no loss of stability with this tolerance chosen properly. The resulting explicit PWL state feedback is defined on an orthogonal partition of the state space that allows very efficient real-time computations through a search tree. Notice, however, that the offline computational complexity and real-time computer memory requirements typically increase exponentially with the number of states. The present results allow model predictive control to be implemented without real-time optimization for systems with only a few states at high sampling frequencies in embedded systems with inexpensive processors and low software complexity.

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] 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.

[4] 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.

[5] 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.

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

[7] P. Tøndel and T. A. Johansen, "Complexity reduction in explicit model predictive control," in *Preprints, IFAC World Congress, Barcelona*, 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. Parisini and R. Zoppoli, "A receding-horizon regulator for nonlinear systems and a neural approximation," *Automatica*, vol. 31, pp. 1443–1451, 1995.

[10] T. Parisini and R. Zoppoli, "Neural approximations for multistage optimal control of nonlinear stochastic systems," *IEEE Trans. on Automatic Control*, vol. 41, pp. 889–895, 1996.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic Programming*, Athena Scientific, Belmont, 1998.

[12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, 2000.

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

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

[15] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, New Jersey, 1970.

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

[17] A. Bemporad and C. Filippi, "Suboptimal explicit RHC via approximate quadratic programming," Preprint, submitted for publication, 2001.

[18] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Trans. Automatic Control*, vol. 36, pp. 1008–1020, 1991.