43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

A Dynamic Programming Approach for Determining the Explicit Solution of Linear MPC Controllers

D. Muñoz de la Peña†, T. Alamo†, A. Bemporad* and E.F. Camacho†

***Abstract*— Recently multi-parametric methods have been applied with success to Model Predictive Control (MPC) schemes. In this paper we propose a novel method for linear systems to obtain the explicit description of the control law that is based on dynamic programming and exploits the structure of the MPC formulation.**

Keywords: Predictive control of linear systems; Multi-parametric programming; Optimal control

I. INTRODUCTION

In practice, all processes are subject to constraints. Actuators have a limited range and output process variables are bounded by constructive, safety or quality reasons. Control systems often operate close to the limits and constraint violation is likely to occur. Model Predictive Control (MPC) is one of the control strategies that is able to cope with constraints in an explicit way [5].

All MPC implementations require the solution of a constrained optimization problem which in general has to be solved by numerical methods. Recent works have proved the piecewise affine nature of MPC controllers for linear systems. Algorithms based on multi-parametric programming for determining the explicit solution are given for both linear and quadratic problems. The explicit solution for the linear quadratic regulator was first introduced by Bemporad et. al in [3].

Algorithms for determining explicit solutions fall into two categories: state space exploration and reverse transformation methods. State space exploration methods (see [3], [12], [13]) build the solution exploring the state space until the whole space is characterized by a set of regions.

Reverse transformation methods (see [11], [7]) are based on evaluating all the possible solution conditions and keeping only the ones which take place in some nonempty region of the state space. In general, the combinatorial explosion of possible solution conditions make these algorithms very inefficient.

In this paper a novel algorithm based on dynamic programming is presented. This algorithm explores all the possible solution conditions (as reverse transformation methods) in such a way that the combinatorial explosion is avoided. The number of conditions to be explored is bounded by the total number of solution regions. The proposed algorithm gives the explicit description of the optimal control input as a function of the current state not only for a prediction horizon n, but for all predictions horizons from 1 to n.

The paper is organized as follows: Section II describes the MPC control strategy and introduces the optimization problem to solve. Section III presents well known results on multi-parametric quadratic programming. Section IV presents the main results of the paper. Section V shows some examples. Section VI presents the main conclusions of the paper.

II. PROBLEM FORMULATION

Consider the problem of regulating the discrete invariant time linear system:

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$

to the origin while fulfilling linear constraints on the state, $x_k \in X$, and on the input, $u_k \in U$, where $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$.

Model Predictive Control (MPC) determines the control input u at each time step by means of the solution of the following optimization problem, denoted $P_n(x)$, for a given prediction horizon n:

$$\begin{aligned} J_n(x) = \quad & \min_{u_0 \cdots u_{n-1}} x_n^T P x_n + \sum_{j=0}^{n-1} x_j^T Q x_j + u_j^T R u_j \\ & x_{k+1} = Ax_k + Bu_k \quad k = 0 \ldots n-1, \\ & x_k \in X \quad k = 1 \ldots n-1, \\ & u_k \in U \quad k = 0 \ldots n-1, \\ & x_n \in \Omega, \\ & x_0 = x, \end{aligned} \tag{2}$$

with $Q = Q^T \geq 0$, $P = P^T \geq 0$ and $R > 0$. This is a general MPC formulation. The terminal region Ω and cost function $x^T P x$ can be designed to assure stability (see [8] for a complete review in MPC stability issues). Other formulations can also be treated by the results of this paper.

The free variable vector is made up of the future input trajectory $\mathbf{u}_n = \{u_0, \ldots, u_{n-1}\}$. The optimization provides a solution which is a function of the current state vector x and is denoted $\mathbf{u}_n^*(x)$. The controller implements the control law by applying only the first control input:

$$u^*(x) = u_0^*,$$

and repeating the optimization at the next time step with the new initial state and the finite horizon shifted by one (receding horizon scheme, see [5]).

*Dip. di Ingegneria dell'Informazione, Università di Siena.
†Dep. de Ingeniería de Sistemas y Automática, Universidad de Sevilla.
E-mail: davidmps@cartuja.us.es (D. Muñoz de la Peña), alamo@cartuja.us.es (T. Alamo), bemporad@unisi.it (A. Bemporad), eduardo@cartuja.us.es (E.F. Camacho)

A. MPC Computation

Taking into account that

$$x_k = A^k x + \sum_{j=0}^{k-1} A^j B u_{k-1-j},$$

$P_n(x)$ can be rewritten in the form

$$\begin{aligned} J_n(x) = x^T Y_n x + \quad & \min_{\mathbf{u}_n} \mathbf{u}_n^T H_n \mathbf{u}_n + 2x^T F_n^T \mathbf{u}_n & (3) \\ & \text{s.t.} \quad G_n \mathbf{u}_n \leq W_n + E_n x, & (4) \end{aligned}$$

where $Y_n, H_n, F_n, G_n, W_n, E_n$ are obtained from the problem parameters (see [5], [3]). By definition, $H_n > 0$.

Definition 1: The feasible region χ_n of $P_n(x)$ is defined by all state vectors for which there exists a future input vector $\mathbf{u}_n$ that satisfies the constraints:

$$\chi_n = \{x \in \mathbb{R}^{n_x} | \ \exists \mathbf{u}_n \text{ such that } G_n \mathbf{u}_n \leq W_n + E_n x\}.$$

Note that χ_n is the projection of a convex polyhedron, and therefore is a convex polyhedral set.

B. Dynamic Programming Approach

The full horizon optimization problem can be solved by means of a sequence of one step optimal control problems. This approach has been successfully applied for determining the explicit solution of feedback min max MPC controllers based on $\|.\|_\infty$ and $\|.\|_1$ norm (see [2]) and of discrete time hybrid systems (see [4]). To the best knowledge of the authors, there is no equivalent result in the literature for constrained MPC controllers with quadratic cost functions.

Property 1: Problem $P_n(x)$ is equivalent to the following optimization problem:

$$\begin{aligned} J_n(x) = \quad & \min_u \ x^T Q x + u^T R u + J_{n-1}(Ax + Bu) \\ & x \in X, \\ & u \in U, \\ & Ax + Bu \in \chi_{n-1}, \end{aligned} \tag{5}$$

with:

$$J_0(x) = x^T P x, \quad \chi_0 = \Omega.$$

Proof: This property is proved using Bellman's optimality principle. ■

III. Multi-parametric Quadratic Programming

In this section well known results are presented (see [3]).

Definition 2: A function $f(x) : \chi \mapsto \mathbb{R}^s$, where χ is a polyhedral set in $\mathbb{R}^{n_x}$, is piecewise affine if it is possible to partition χ into convex polyhedral regions, R^i, and

$$f(x) = f^i(x) = H^i x + h^i, \ \forall x \in R^i.$$

Definition 3: A function $f(x) : \chi \mapsto \mathbb{R}$, where χ is a polyhedral set in $\mathbb{R}^{n_x}$, is piecewise quadratic if it is possible to partition χ into convex polyhedral regions, R^i, and

$$f(x) = f^i(x) = x^T P^i x + H^i x + h^i, \ \forall x \in R^i.$$

Theorem 1 (cf. [3], Theorem 4): Consider the following mpQp problem

$$\begin{aligned} V(x) \ &= \ \min_{\mathbf{z}} \ \frac{1}{2}\mathbf{z}^T H \mathbf{z} & (6) \\ s.t. \quad & G\mathbf{z} \leq W + Sx, & (7) \end{aligned}$$

and let $H > 0$. Then the optimizer $\mathbf{z}^*(x)$ is continuous piecewise affine and the optimal solution $V(x)$ is continuous, convex and piecewise quadratic.

Any optimization problem equivalent to (6) has an explicit solution that can be determined using a multi-parametric quadratic programming solver.

Each critical region R^i of an explicit solution, is defined by a set of active constraints $\tilde{G}z \leq \tilde{W} + \tilde{S}x$ out of (7). For this set of constraints the following equality holds:

$$\tilde{G}z^i(x) = \tilde{W} + \tilde{S}x. \tag{8}$$

Theorem 2 (cf. [3], Theorem 2, Section 4.1.1): Consider a set of constraints, $\tilde{G}, \tilde{W}, \tilde{S}$, out of (7). The set of state vectors in which such a combination of constraints (set A^i) is active at the optimum and the corresponding optimal feedback control law, denoted R^i and $z^i(x)$ respectively, are uniquely defined as follows:

If $\tilde{G}$ is full rank, denote $(\tilde{G}H^{-1}\tilde{G}^T)^{-1} = M$, then

$$z^i(x) = H^{-1}\tilde{G}^T M(\tilde{W} + \tilde{S}x),$$
$$R^i = \left\{ x \mid \begin{array}{l} -M(\tilde{W} + \tilde{S}x) \geq 0 \\ GH^{-1}\tilde{G}^T M(\tilde{W} + \tilde{S}x) \leq W + Sx \end{array} \right\},$$

else a degenerate situation is encountered that can be dealt with as in [3] (Section 4.1.1).

Given a set of constraints A and using Theorem 2 its critical region can be defined. If it is empty, that set is never the active set for any given state vector. The explicit solution of a mpQp problem is defined by all possible critical regions, and so, by the collection of constraints sets that defines them.

State space exploration algorithms are based on finding the active sets from known solutions of the optimization problem. In these methods the optimization problem is solved with numerical methods for different state vectors until the whole state space is explored and characterized. See [3], [12], [13].

Reverse transformation methods are based on exploring all possible subsets that can be built from the whole set of constraints. In this way all critical regions are found. The general principle of this method is formulating a condition that is potentially satisfied by the optimal solution in some state vector (i.e. the active constrain set A). This condition, if known a priori, enables a simple solution to the problem (Theorem 2). Finally, the set of state vectors for which this condition is satisfied is determined (the critical region R^i). It may be empty. By repeating this procedure for all possible conditions the explicit solution is defined as the union of all non empty non redundant regions. The amount of possible active sets grows combinatorially explosion. The number of possible subsets of constraints is 2^q, where q is the number of constraints (see [11], [7]).

In this work a reverse transformation algorithm (Algorithm 1) that explores a given collection of constraints sets Γ is used. The result of the algorithm is the collection $\Phi \subseteq \Gamma$ of constraint sets which are active at some state vector x.

Algorithm 1:

- $\Phi = \emptyset$
- for each A^i in Γ
  - Apply Theorem 2 and define CR^i
  - If CR^i is not empty then $\Phi = \Phi \bigcup A^i$
- end

Definition 4: We define $\Phi = \text{Rev}(\Gamma)$ as the collection of constraints sets obtained from applying Algorithm 1 to the collection of constraints sets Γ.

A. MPC as mpQp

Property 2: The optimization problem $P_n(x)$ has an optimizer $\mathbf{u}_n^*(x)$ continuous piecewise affine and the optimal solution $J_n(x)$ is continuous, convex and piecewise quadratic. The critical regions are denoted CR_n^i, $i = 1, \ldots, S(n)$, and the corresponding affine control law and quadratic cost function are denoted $\mathbf{u}_n^i(x)$ and $J_n^i(x)$. The active constraints set of each critical region is denoted A_n^i.

Proof: $P_n(x)$ can be posed as Problem (6) using the following variable change:

$$\mathbf{z} = \mathbf{u}_n + H_n^{-1} F_n x.$$

■

It is important to note that matrices H, G, W are equal to H_n, G_n, W_n respectively, and that $S = E_n + G_n H_n^{-1} F_n$.

Definition 5: Given any affine control law, $\mathbf{u}(x) = Kx + q$, for $P_n(x)$, we define the constraint set A of $\mathbf{u}(x)$, the maximum subset of constraints defined by $\tilde{G}_n$, $\tilde{W}_n$ and $\tilde{E}_n$ out of (4) that satisfy

$$\begin{aligned} \tilde{G}_n K &= \tilde{E}_n, \\ \tilde{G}_n q &= \tilde{W}_n. \end{aligned}$$

Property 3: The set of constraints given by Definition 5 for a given optimal control law $\mathbf{u}_n^i(x)$, is equal to the active set, A_n^i, that defines the critical region in the equivalent Problem (6) by Theorem 2.

Proof: By definition, (8) holds for $z^i = \mathbf{u}_n^i(x) + H_n^{-1} F_n x$ and the set of constraints $\tilde{G}, \tilde{W}, \tilde{S}$ defined by A_n^i. As $\tilde{S} = \tilde{E}_n + \tilde{G}_n H_n^{-1} F_n$ and (8) holds for all possible state vectors x, is easy to see that

$$\begin{aligned} \tilde{G}_n K &= \tilde{E}_n, \\ \tilde{G}_n q &= \tilde{W}_n. \end{aligned}$$

■

In this paper we present an algorithm that builds a collection of hypothesis sets of constraints that contains all the active constraints sets for a given MPC problem. A reverse transformation method over this collection of hypothesis sets characterizes the explicit solution. This hypothesis set is built in a way such that the combinatorial explosion is avoided, this is, the number of constraints sets to be explored depends only on the actual complexity of the controller.

Definition 6: Given the collection of all active constraints sets for $P_n(x)$, denoted the solution collection Φ_n, a collection Γ_n of constraints sets is an hypothesis collection for $P_n(x)$ if

$$\Phi_n \subseteq \Gamma_n.$$

By definition, $\Phi_n = \bigcup_{i \in S(n)} A_n^i$.

IV. Algorithm for Determining an Hypothesis Set

In this algorithm, an hypothesis collection of $P_n(x)$ is built from the explicit solution of $P_{n-1}(x)$.

For each region CR_{n-1}^i, we define the optimal one step cost function $Jh_n^i(x)$ as a MPC optimization problem with a prediction horizon of one as follows:

Definition 7: Given a critical region CR_{n-1}^i of $P_{n-1}(x)$, we define the one step optimal cost function, $Jh_n^i(x)$, as:

$$\begin{aligned} Jh_n^i(x) = \quad & \min_{u_n^i} x^T Q x + u_n^{iT} R u_n^i + J_{n-1}^i(Ax + Bu_n^i) \\ & x \in X, \\ & u_n^i \in U, \\ & Ax + Bu_n^i \in CR_{n-1}^i. \end{aligned} \tag{9}$$

For the first iteration, $J_0(x) = x^T P x$ and CR_0 is Ω.

Property 4: The optimization Problem (9) has an optimizer $u_n^i(x)$ continuous piecewise affine and the optimal solution $Jh_n^i(x)$ is continuous, convex and piecewise quadratic. The critical regions are denoted CR_n^{ij}, $j \in S_i(n)$, and the corresponding affine control law and quadratic cost function are denoted $u_n^{ij}(x)$ and $Jh_n^{ij}(x)$.

Proof: Taking into account that $J_{n-1}^i(x)$ is a quadratic function on x, Problem (9) can be posed as:

$$\begin{aligned} Jh_n^i(x) = \quad & x^T Y_n^i x + \min_{u_n^i} u_n^{iT} H_n^i u_n^i + 2(x^T F_n^{iT} + f_n^i) u_n^i \\ & G_n^i u_n^i \leq W_n^i + E_n^i x. \end{aligned}$$

Making the variable change

$$\mathbf{z} = u_n^i + (H_n^i)^{-1}(F_n^i x + f_n^i),$$

this optimization problem is equivalent to Problem (6). ■

The set

$$\chi_n^i = \bigcup_{j \in S_i(n)} CR_n^{ij},$$

is made up of the state vectors that can reach CR_{n-1}^i in one step. By definition and taking into account Property 1, it is clear to see that the following statements hold

$$\begin{aligned} \chi_n &= \bigcup_{i \in S(n-1)} \chi_n^i, \\ J_n(x) &= \min_i \{Jh_n^i(x),\ i \in S(n-1)\}. \end{aligned}$$

These sets in general overlap in the state space. See Figure 2. In this paper, we propose to use this information to build a hypothesis collection Γ_n of $P_n(x)$.

For each critical region of the one step optimal cost function, CR_n^{ij}, we define a feedback affine hypothesis control law $\mathbf{u}_n^{ij}(x)$ as follows:

Definition 8: Given a critical region CR_n^{ij} of the optimal one step functional $Jh_n^i(x)$, we define the hypothesis affine feedback control law for $P_n(x)$ as

$$\mathbf{u}_n^{ij}(x) = \left[\begin{array}{c} u_n^{ij}(x) \\ \mathbf{u}_{n-1}^i(Ax + Bu_n^{ij}(x)) \end{array} \right].$$

The first control input is given by $u_n^{ij}(x)$, while the rest of the input trajectory is defined by $\mathbf{u}_{n-1}^i(Ax + Bu_n^{ij}(x))$.

Property 5: The hypothesis feedback control law $\mathbf{u}_n^{ij}(x)$ is a feasible input for $P_n(x)$ for all $x \in CR_n^{ij}$.

Property 6: If $x \in \chi_n$ then there exists CR_n^{ij} such that

$$\mathbf{u}_n^*(x) = \mathbf{u}_n^{ij}(x).$$

Properties 5 and 6 are proven by Property 1.

Not all these control laws are optimal, but the optimal control laws that defines the explicit solution of $P_n(x)$ are included in this set.

Property 7: The collection of constraints sets A_n^{ij} defined applying Definition 5 over all possible hypothesis control laws $\mathbf{u}_n^{ij}(x)$, is a hypothesis collection of constraints for $P_n(x)$.

Proof: From Property 6 follows that given $x \in CR_n^k$, there exist CR_n^{ij} such that $x \in CR_n^{ij}$ and

$$\mathbf{u}_n^*(x) = \mathbf{u}_n^k(x) = \mathbf{u}_n^{ij}(x).$$

Then from Property 3 follows that

$$\Phi_n \subseteq \Gamma_n = \bigcup_{i \in S(n-1)} \bigcup_{j \in S_i(n)} A_n^{ij}.$$

■

The following algorithm builds an hypothesis collection of constraints sets Γ_n for $P_n(x)$ from $\Phi_{n-1}(x)$, the solution collection of $P_{n-1}(x)$.

Algorithm 2:

- for each set A_{n-1}^i in Φ_{n-1}
  - build CR_{n-1}^i, $\mathbf{u}_{n-1}^i(x)$, $J_{n-1}^i(x)$
  - solve $Jh_n^i(x)$ with a mpQp solver
  - for $j \in S_i(n)$
    - build $\mathbf{u}_n^{ij}(x)$
    - build A_n^{ij}
  - end
- end
- $\Gamma_n = \bigcup A_n^{ij}$

The algorithm builds all possible A_n^{ij}. Figure 1 shows a flow chart of the proposed algorithm.

Definition 9: We define $\Gamma = \text{Dyn}(\Phi)$ as the collection of constraints sets obtained from applying Algorithm 2 to the collection of constraints sets Φ.

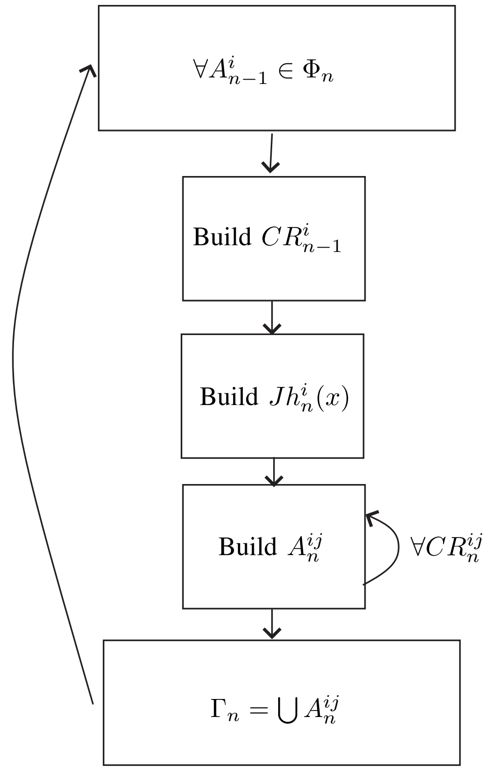

Fig. 1. Flow chart of the algorithm for determining an hypothesis set.

A. Algorithm for determining the explicit solution

The collection of active constraints sets of $P_n(x)$ can be obtained from the one of $P_{n-1}(x)$ as

$$\Phi_n = \text{Rev}(\text{Dyn}(\Phi_{n-1})). \tag{10}$$

The explicit solution of $P_n(x)$ can be obtained as follows:

Algorithm 3:

- $\Phi_0 = \emptyset$ (see Definition 7)
- for l = 1 to n, do $\Phi_l = \text{Rev}(\text{Dyn}(\Phi_{l-1}))$
- end

This algorithm not only characterizes the solution for a given prediction horizon n, but for all prediction horizon from 1 to n.

V. Example

Consider the linear system described by matrices

$$A = \left[\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array} \right], B = \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \tag{11}$$

subject to the following constraints

$$-10 \leq x_k \leq 10, \quad -1 \leq u_k \leq 1.$$

The control performance objective is described by

$$Q = \left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right], \ R = 1.$$

TABLE I

SIZE OF THE SOLUTION (Φ_N) AND HYPOTHESIS (Γ_N) COLLECTION AND THE COMPUTATION TIMES ($T_1[s]$ AND $T_2[s]$) OF THE EXPLICIT SOLUTION FOR DIFFERENT PREDICTION HORIZONS N FOR SYSTEM (11).

N	1	2	3	4	5
Φ_N	7	15	27	41	55
Γ_N	7	29	45	73	105
$T_1[s]$	0.14	0.61	1.38	2.62	4.26
$T_2[s]$	0.82	0.92	1.71	3.41	5.03
N	6	7	8	9	10
Φ_N	67	79	89	97	101
Γ_N	137	164	189	212	231
$T_1[s]$	6.13	8.45	11.24	13.51	15.59
$T_2[s]$	7.56	10.01	12.61	15.64	18.94

The terminal cost function matrix P is defined as the solution to the algebraic Riccati equation, and Ω as the maximal control invariant set of the unconstrained LQR control law.

Figure 2 shows the state partition of $J_1(x)$ and of two one step optimal cost functions ($J_2^3(x)$ and $J_2^5(x)$). These figures show how the critical regions for the one step cost functions may overlap, so they do not define the optimum solution directly. Figure 3 shows the state partition for predictions horizons six, eight and twelve. As the terminal cost is the maximum invariant for the LQR control law, and the terminal cost function is defined as the solution to the algebraic Riccati equation; these figures show how as the prediction horizon is increased, some regions remain the same. The regions that do not change define the explicit solution to the infinite horizon constrained LQR problem (see [10], [6]). The dynamic programming approach of algorithm 3 allow us to detect this invariant regions and thus characterize the region where the optimal control law is obtained. Future works will develop a specific algorithm for solving this problem in an efficient way.

Table I shows the size of the hypothesis set Γ_N and the solution set Φ_N for different prediction horizons as well as the computation times T_1 and T_2. Entry T_1 is the time needed to obtain the explicit solution of $P_n(x)$ **from** the one of $P_{n-1}(x)$ using (10) and T_2 is the time needed to obtain the explicit solution of $P_n(x)$ using the mpQp algorithm provided in [9].

VI. CONCLUSIONS

In this paper an alternative way of determining the explicit solution for a MPC controller has been proposed. The main contribution is to use a dynamic programming approach algorithm. that provides the explicit description of the optimal control input as a function of the current state over the whole control horizon. This approach allow us to build the explicit solution for a given prediction horizon N from the one of $N-1$.

Simulation times are comparable with *state of the art* solvers for evaluating the explicit description of the optimal control input as a function of the current state not only for

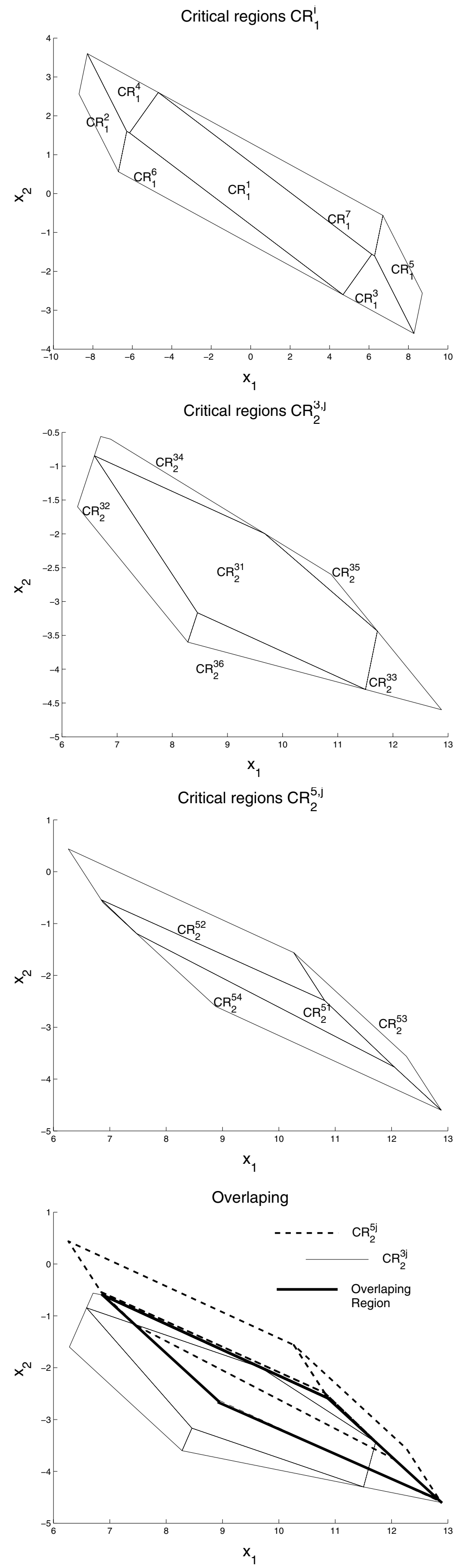

Fig. 2. State space partition of $J_1(x)$ and of two one step optimal cost functions ($J_2^3(x)$ and $J_2^5(x)$) for system (11). Note that axis are displayed with different scales.

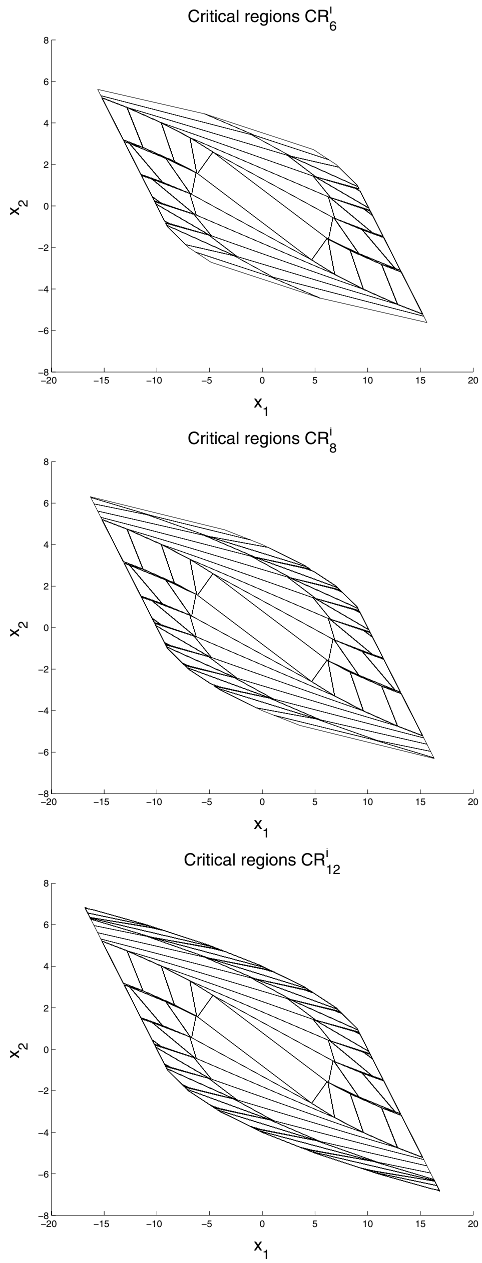

Fig. 3. State space partition of $J_6(x)$, $J_8(x)$ and $J_{12}(x)$ for system (11).

a prediction horizon n, but for all predictions horizons from 1 to n.

The proposed algorithm has not been optimized and is open to improvements. In particular, a *ring* approach as the one proposed in [1] may considerably improve the performance because would eliminate the need to evaluate the one step cost function for all the critical regions of the solution in $N-1$, but only for the ones that may potentially define new constraint sets.

Dynamic programming is a natural way of dealing with MPC problems and gives an insight of the structure of the controllers. Moreover, although it has not been proved, the authors believe that for high dimension problems and long horizons, the benefits of a recursive approach would be more stressed.

REFERENCES

[1] M. Baotic, F.J. Christophersen, and M. Morari. Infinite time optimal control of hybrid systems with a linear performance index. In *Conference on Decision and Control*, Maui, Hawaii, USA, 2003.

[2] A. Bemporad, F. Borrelli, and M. Morari. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Trans. Automatic Control*, 48(9):1600–1606, 2003.

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

[4] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari. An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In *American Control Conference*, pages 4717 – 4722, Denver, Colorado, 2003.

[5] E.F. Camacho and C. Bordóns. *Model Predictive Control*. Springer-Verlag, 1999.

[6] S.S. Keerthi and E.G. Gilbert. Optimal infinite-horizon feedback control laws for general class of constrained discrete-time systems: Stability and moving horizon approximation. *Journal of Optical Theory and Applications*, 57:265–293, 1988.

[7] D.Q. Mayne and S.V. Rakovic. Optimal control of constrained piecewise affine discrete time systems using reverse transformation. In *Conference on Decision and Control*, pages 1546–1551, Las Vegas,Nevada USA, 2002.

[8] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[9] M.Baotic. An efficient algorithm for multiparametric quadratic programming. In *Technical Report AUT02-04*, Automatic Control Laboratory, ETH, Zurich, Switzerland, 2002.

[10] M.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *Transactions on Automatic Control*, 43(8):1163–1169, 1998.

[11] M.M. Seron, G.C. Goodwin, and J.A. De Doná. Finitely parameterised implementation of receding horizon control for constrained linear systems. In *American Control Conference*, pages 4481–4485, Anchorage, AK, 2002.

[12] P. Tøndel, T.A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, 39(3):489–497, 2003.

[13] P. Tøndel, T.A. Johansen, and A. Bemporad. Further results on multiparametric quadratic programming. In *Proc. Conference on Decision and Control*, pages 3173–3178, Maui, Hawaii, USA, 2003.