Model Predictive Control Based on Linear Programming—The Explicit Solution

Alberto Bemporad, Francesco Borrelli, and Manfred Morari

***Abstract*—We study model predictive control (MPC) schemes for discrete-time linear time-invariant systems with constraints on inputs and states, that can be formulated using a linear program (LP). In particular, we focus our attention on performance criteria based on a mixed $1/\infty$-norm, namely, 1-norm with respect to time and ∞-norm with respect to space. First we provide a method to compute the terminal weight so that closed-loop stability is achieved. We then show that the optimal control profile is a piecewise affine and continuous function of the initial state and briefly describe the algorithm to compute it. The piecewise affine form allows to eliminate online LP, as the computation associated with MPC becomes a simple function evaluation. Besides practical advantages, the availability of the explicit structure of the MPC controller provides an insight into the type of control action in different regions of the state space, and highlights possible conditions of degeneracies of the LP, such as multiple optima.**

***Index Terms*—Constraints, linear programming (LP), model predictive control (MPC), multiparametric programming, piecewise linear control.**

I. INTRODUCTION

FOR COMPLEX constrained multivariable control problems, *model predictive control* (MPC) has become the accepted standard in the process industries [1]. Here at each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon. The optimal command signal is applied to the process only during the immediately following sampling interval. At the next time step a new optimal control problem based on new measurements of the state is solved over a shifted horizon. The optimal solution relies on a linear dynamic model of the process, respects all input and output constraints, and minimizes a performance figure. This is usually expressed as a *quadratic* or a *linear* criterion, so that the resulting optimization problem can be cast as a quadratic program (QP) or linear program (LP), respectively, for which a rich variety of efficient active-set and interior-point solvers are available.

The first MPC industrial algorithms like IDCOM [2] and DMC [3] were developed for unconstrained MPC based on quadratic performance indices, and later followed by algorithms based on QP, like QDMC [4], for solving constrained MPC problems. Later an extensive theoretical effort was devoted to analyze such schemes, provide conditions for guaranteeing feasibility and closed-loop stability, and highlight the relations between MPC and linear quadratic regulation [5], [6].

On the other hand, the use of linear programming was proposed in the early sixties by Zadeh and Whalen for solving optimal control problems [7], and by Propoi [8], who perhaps conceived the first idea of MPC. Later, only a few other authors have investigated MPC based on linear programming [9]–[13], where the performance index is expressed as the sum of the ∞- or 1-norm of the input command and of the deviation of the state from the desired value.

In this paper, we review the basics of MPC based on the minimization of a mixed $1/\infty$-norm, namely, 1-norm with respect to time and ∞-norm with respect to space (this choice will be motivated in Section II), derive the associated linear program, and provide guidelines for choosing the terminal weight so that closed-loop stability is achieved. Then, we determine explicitly the structure of LP-based MPC, and show that it can be equally expressed as a piecewise affine and continuous state feedback law. This provides an insight into the behavior of the MPC controller in different regions of the state space, highlighting regions where saturation or idle control occur, regions where the LP has multiple optima, etc. Besides such insights, the availability of the explicit MPC structure provides a clear computational benefit: no on-line LP solver is needed in the MPC implementation, which requires only the evaluation of a piecewise affine function.

The problem of synthesizing piecewise affine stabilizing feedback controllers for linear discrete-time systems subject to input and state constraints was also addressed in [14]. The authors obtained a piecewise linear feedback law defined over a partition of the set of states into simplicial cones, by computing a feasible input sequence for each vertex via linear programming (this technique was later extended in [15]). The approach presented in this paper provides a piecewise affine control law which not only ensures feasibility and stability, but is also optimal with respect to a performance index. Our approach is close in spirit to the technique proposed recently in [16] for MPC based on a quadratic performance index: As the linear program depends on the current state which appears linearly in the constraints, the key idea is to treat the LP as a *multiparametric linear program* (mp-LP), whose properties have been studied in [17] and [18], and for which we developed an efficient solver [19]. Note that even if LP can be viewed as a special case of QP by letting the Hessian matrix $H = 0$,

Manuscript received January 19, 2001; revised December 14, 2001 and April 9, 2002. Recommended by Associate Editor G. De Nicolao. This work was supported in part by the European Community under Project 33520 "CC—Computation and Control," and by Ministero dell' Istruzione, dell' Universitá e della Ricerca.

A. Bemporad is with the Dipartimento Ingegneria dell'Informazione, University of Siena, 53100 Siena, Italy (e-mail: bemporad@dii.unisi.it).

F. Borrelli and M. Morari are with the Automatic Control Laboratory, ETH Zentrum, 8092 Zurich, Switzerland (e-mail: borrelli@aut.ee.ethz.ch; morari@aut.ee.ethz.ch).

Digital Object Identifier 10.1109/TAC.2002.805688

the results of [16] on *multi-parametric quadratic programming* (mp-QP) are restricted only to the case $H > 0$. As a matter of fact, mp-LP deserves a special analysis, which leads to a deep insight of mp-LP properties and a different algorithm than the mp-QP algorithm described in detail in [16].

The paper concludes with a series of examples that illustrate the different features of the method.

II. Model Predictive Control With $1/\infty$-Norm

Consider the problem of regulating to the origin the discrete-time linear time-invariant system

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{1}$$

while fulfilling the constraints[1]

$$y_{\min} \le y(t) \le y_{\max}, \qquad u_{\min} \le u(t) \le u_{\max} \tag{2}$$

at all time instants $t \ge 0$. In (1) and (2), $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, input, and output vector respectively, and the pair (A, B) is stabilizable. In (2), $y_{\min} \le 0 \le y_{\max}$ and $u_{\min} \le 0 \le u_{\max}$ are vectors of upper and lower bounds (more generally, we can allow only some components of the inputs or outputs to be constrained).

MPC solves such a constrained regulation problem in the following way. Assume that a full measurement of the state $x(t)$ is available at the current time t. Then, the optimization problem

$$\min_{U \triangleq [u_0^T, \ldots, u_{N_u-1}^T]^T} \left\{ J(U, x(t)) \triangleq \left\| P x_{N_y} \right\|_\infty + \sum_{k=1}^{N_y-1} \|Q x_k\|_\infty + \sum_{k=0}^{N_y-1} \|R u_k\|_\infty \right\}$$

s.t.

$$\begin{aligned} & y_{\min} \le y_k \le y_{\max}, \qquad k = 1, \ldots, N_c \\ & u_{\min} \le u_k \le u_{\max}, \qquad k = 0, 1, \ldots, N_{u-1} \\ & x_0 = x(t), \\ & x_{k+1} = Ax_k + Bu_k, \qquad k = 0, 1, \ldots, N_y - 1 \\ & u_k = 0, \qquad N_u \le k \le N_y - 1 \end{aligned} \tag{3}$$

is solved at each time t, where x_k denotes the k-step ahead prediction of the state vector, obtained by applying the input sequence $u_0, \ldots, u_{k-1}$ to model (1) starting from the state $x(t)$, $\|Vx\|_\infty \triangleq \max_{i=1,\ldots,r} (V^{\{i\}}x)$, and $V^{\{i\}}$ is the ith row of a generic matrix $V \in \mathbb{R}^{r \times n}$.

In (3), we assume that $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$ are nonsingular matrices, $P \in \mathbb{R}^{r \times n}$ is a full-column rank matrix, and $N_y \ge N_c \ge N_u$. The performance index $J(U, x(t))$ amounts to the sum of a weighted mixed 1- and ∞-norm of the input and of the state deviation from the origin, namely the 1-norm with respect to time of the ∞-norm with respect to space. Although any combination of 1- and ∞-norms leads to a linear program, our choice is motivated by the fact that ∞-norm over time could result in a poor closed-loop performance (only the largest state deviation and the largest input would be penalized over the prediction horizon), while 1-norm over space leads to an LP with a larger number of variables, as will be clarified in Section III.

Let $U^*(x(t)) \triangleq [u_0^{*T}(x(t)), \ldots, u_{N_u-1}^{*T}(x(t))]^T$ be the optimal solution of (3) at time t. Then, the first sample of $U^*(x(t))$ is applied to (1)

$$u(t) = u_0^*(x(t)). \tag{4}$$

The optimization (3) is repeated at time $t+1$, based on the new state $x(t+1)$, yielding a *moving* or *receding horizon* control strategy.

The two main issues regarding this policy are the feasibility of the optimization problem (3) and the stability of the resulting closed-loop system.

A. Feasibility and Constraint Horizon

Several authors have addressed the problem of guaranteeing at each time step feasibility of the optimization problem associated with MPC. Clearly, if only input constraints are present, there is no feasibility issue at all ($u = 0$ is always feasible). On the other hand, in the presence of output constraints, the MPC problem (3) may become infeasible, even in the absence of disturbances.

One possibility is to soften the output constraints and to penalize the violations [20], [21]. In the case of hard output constraints, Keerthi and Gilbert [22] proved that feasibility (as well as stability) is guaranteed by setting $N_c = N_u = N_y = \infty$, or alternatively, $x_{N_y} = 0$ and $N_c = N_u = N_y - 1$. Setting $N_c = \infty$ leads to an optimization problem with an infinite number of constraints that is impossible to handle. On the other hand, the constraint on the terminal state is undesirable, as it might severely perturb the input trajectory from optimizing performance, especially on short horizons.

By using arguments from maximal output admissible set theory, Gilbert and Tan [23] proved that if the set of feasible state and input vectors is bounded and contains the origin in its interior, a finite horizon N_c is sufficient for ensuring feasibility. The smallest N_c ensuring feasibility of the MPC problem (3) at all time instants can be computed by solving a sequence of linear programs [23], [14], [24], [25].

B. Stability and Terminal Weight

In general, stability is a complex function of the various tuning parameters N_u, N_y, N_c, P, Q, and R. For applications it is most useful to impose some conditions on N_y, N_c and P so that stability is guaranteed for all nonsingular Q and R, and leave Q and R as free tuning parameters for performance. Sometimes the optimization problem (3) is augmented with the so called “stability constraint” (see, e.g., the surveys [26] and [5]). This additional constraint imposed over the prediction horizon explicitly forces the state vector either to shrink in some

[1] Although the form (2) is very common in practical implementations of MPC, the results of this paper also hold for the more general mixed constraints $Ex(t) + Lu(t) \le M$ arising, for example, from constraints on the input rate $\Delta u(t) \triangleq u(t) - u(t-1)$.

norm or to reach an invariant set at the end of the prediction horizon.

Although most of the stability results were developed for MPC formulations based on squared Euclidean norms, for general nonlinear performance functions and models, the idea of Keerthi and Gilbert [22] mentioned above [i.e., N_y, N_u, $N_c \to \infty$ or $x_{N_y} = 0$ (end-point constraint)] not only ensures persistence of solutions (i.e., feasibility at each time step) but also stability, provided that the problem is feasible at time $t = 0$.

Another possibility is to relax the end-point constraint by adopting a *dual-mode* approach [27], namely, by defining an invariant set around the origin, and constrain the terminal state x_{N_y} to lie in that set.

In this paper, rather than constraining the final state, we provide conditions for the weight P over x_{N_y} in (3) that guarantee closed-loop stability, provided that the matrix A is stable, and suggest a procedure for constructing such stabilizing P.

In case the matrix A is unstable, the procedure for constructing P can be still applied by pre-stabilizing system (1) via a linear controller without taking care of the constraints. Then, the output vector can be augmented by including the original (now state-dependent) inputs, and saturation constraints can be mapped into additional output constraints in (3).

Assuming that the constraint horizon N_c is long enough so that the shifted optimal input sequence $[u_1^{*T}(x(t)), \ldots, u_{N_u-1}^{*T}(x(t)), 0]^T$ is feasible at the next time step $t+1$ [N_c is chosen according to the techniques recalled in Section II-A), the following theorem shows that, by appropriately choosing the terminal weight P, the control law (3) stabilizes (1) asymptotically.

Theorem 1: Let A be a stable matrix, and assume that the initial state $x(0)$ is such that a feasible solution of problem (3) exists at time $t = 0$. Assume that there exists a full-column rank matrix P such that

$$-\|Px\|_\infty + \|PAx\|_\infty + \|Qx\|_\infty \le 0 \tag{5}$$

is satisfied for all $x \in \mathbb{R}^n$. Then, for N_c sufficiently large, the MPC law (3) and (4) ensures the fulfillment of the input and output constraints $u_{\min} \le u(t) \le u_{\max}$, $y_{\min} \le y(t) \le y_{\max}$, and $\lim_{t\to\infty} x(t) = 0$, $\lim_{t\to\infty} u(t) = 0$.

Proof: The proof follows from standard Lyapunov arguments, close in spirit to the arguments of [22], [28] where is established the fact that under some conditions the value function $V(t) \triangleq J(U^*(x(t)), x(t))$ attained at the minimizer $U^*(x(t)) = [u_0^{*T}(x(t)), u_1^{*T}(x(t)), \ldots, u_{N_u-1}^{*T}(x(t))]^T$ of (3) is a Lyapunov function for the closed-loop system. Under the assumption that N_c is sufficiently large (see Section II-A), the shifted sequence $U_{\text{shift}} \triangleq \{u_1^*(x(t)), \ldots, u_{N_u-1}^*(x(t)), 0\}$ is feasible at time $t+1$, and

$$V(t+1) - V(t) \le -\|Qx(t)\|_\infty - \|Ru(t)\|_\infty - \left\|Px^*_{N_y}\right\|_\infty + \left\|Px^*_{N_y+1}\right\|_\infty + \left\|Qx^*_{N_y}\right\|_\infty. \tag{6}$$

As condition (5) is satisfied for $x = x^*_{N_y}$, $V(t)$ is a decreasing sequence. Since $V(t)$ is lower-bounded by 0, there exists $V_\infty = \lim_{t\to\infty} V(t)$, which implies $V(t+1) - V(t) \to 0$. Therefore

$$\lim_{t\to\infty} \|Qx(t)\|_\infty + \|Ru(t)\|_\infty = 0 \tag{7}$$

which proves the theorem, as Q and R are nonsingular. □

When weighted squared 2-norms are used instead of ∞-norms in (3), (5) becomes $x^T(-P + A^TPA + Q)x \le 0$, which is satisfied by any pair P, Q solving the Lyapunov equation $P = A^TPA + Q$.

The question now arises if matrices P and Q satisfying (5) exist, and how to find them. Let us focus on a simpler problem by removing the factor $\|Qx\|_\infty$ from (5)

$$-\|\tilde{P}x\|_\infty + \|\tilde{P}Ax\|_\infty \le 0. \tag{8}$$

The existence and the construction of a matrix $\tilde{P}$ that satisfies (8), has been addressed in different forms by several authors [29]–[31], [14], [15], [32]. There are two equivalent ways of tackling this problem: for the autonomous system $x(t+1) = Ax(t)$, find a Lyapunov function of the form

$$\Psi(x) = \|\tilde{P}x\|_\infty \tag{9}$$

with $\tilde{P} \in \mathbb{R}^{r\times n}$ full-column rank, $r \ge n$, or equivalently compute a symmetrical positively invariant polyhedral set [32].

Unlike the 2-norm case, the condition that matrix A has all the eigenvalues in the open disk $\|\lambda_i(A)\| < 1$ is not sufficient for the existence of a Lyapunov function (9) with P square ($r = n$) [29]. The following theorem, proved in [31], [30], states necessary and sufficient conditions for the existence of the Lyapunov function (9).

Theorem 2: The function $\Psi(x) = \|\tilde{P}x\|_\infty$ is a Lyapunov function for the autonomous system $x(t+1) = Ax(t)$ if and only if there exists a matrix $H \in \mathbb{C}^{r\times r}$ such that

$$\tilde{P}A - H\tilde{P} = 0 \tag{10a}$$

$$\|H\|_\infty < 1 \tag{10b}$$

where

$$\|H\|_\infty = \sup_{x\neq 0} \frac{\|Hx\|_\infty}{\|x\|_\infty} = \max_{i=1,\ldots,r} \sum_{j=1}^{r} \left|H^{\{ij\}}\right|$$

is the infinity (induced) norm of H. □

In [31] and [30], the authors proposed an efficient way to compute $\Psi(x)$ by constructing matrices $\tilde{P}$ and H satisfying (10). By using the results of [31] and [30], the construction of a matrix P satisfying (5) can be performed by exploiting the following result.

Proposition 1: Let $\tilde{P}$ and H be matrices satisfying (10), with $\tilde{P}$ full rank. Let $\sigma \triangleq 1 - \|H\|_\infty$, $\rho \triangleq \|Q\tilde{P}^{\#}\|_\infty$, where $\tilde{P}^{\#} \triangleq (\tilde{P}^T\tilde{P})^{-1}\tilde{P}^T$ is the left pseudoinverse of $\tilde{P}$. Then, the square matrix

$$P = \frac{\rho}{\sigma}\tilde{P} \tag{11}$$

satisfies (5).

Proof: Since P satisfies $PA = HP$, we obtain

$$\begin{aligned}&-\|Px\|_\infty + \|PAx\|_\infty + \|Qx\|_\infty \\ &\quad = -\|Px\|_\infty + \|HPx\|_\infty + \|Qx\|_\infty \\ &\quad \le (\|H\|_\infty - 1)\|Px\|_\infty + \|Qx\|_\infty \\ &\quad \le (\|H\|_\infty - 1)\|Px\|_\infty + \left\|Q\tilde{P}^{\#}\right\|_\infty \|\tilde{P}x\|_\infty = 0.\end{aligned}$$

Therefore, (5) is satisfied. □

In [30] the author shows how to construct matrices $\tilde{P}$ and H in (10) with the only assumption that A is stable. However, this approach has the drawback that the number r of rows $\tilde{P}$ may go to infinity when the moduli $|\lambda_i|$ of the eigenvalues of A approach 1.

In [31], the authors construct a square matrix $\tilde{P} \in \mathbb{R}^{n\times n}$ under the assumption that the matrix A in (1) has distinct eigenvalues $\lambda_i = \mu_i + j\sigma_i$ located in the open square $|\mu_i| + |\sigma_i| < 1$.

In [12], the authors use a different approach based on Jordan decomposition to construct a stabilizing terminal weighting function for the MPC law (3) and (4). The resulting function leads to a matrix P with $r = 2^{n-n_0-1} + n2^{n_0-1}$ rows where n_0 is the algebraic multiplicity of the zero eigenvalues of matrix A. The result seems to hold only for matrices A with stable and real eigenvalues and, therefore, in general the approach of [31] is preferable.

Remark 1: If $P \in \mathbb{R}^{n\times n}$ is given in advance rather than computed as in Proposition 1, (5) can be tested numerically, either by enumeration (3^{2n} LPs) or, more conveniently, through a mixed-integer linear program with $(5n+1)$ continuous variables and $4n$ integer variables.

III. Piecewise Affine MPC Law With $1/\infty$-Norm

The MPC formulation (3) can be rewritten as a linear program by using the following standard approach (cf., e.g., [10]). The sum of the components of any vector $[\varepsilon_1^x, \ldots, \varepsilon_{N_y}^x, \varepsilon_1^u, \ldots, \varepsilon_{N_u}^u]^T$ that satisfies

$$\begin{aligned}-\mathbf{1}_n\varepsilon_k^x &\le Qx_k, && k = 1, 2, \ldots, N_y - 1\\ -\mathbf{1}_n\varepsilon_k^x &\le -Qx_k, && k = 1, 2, \ldots, N_y - 1\\ -\mathbf{1}_r\varepsilon_{N_y}^x &\le Px_{N_y} &&\\ -\mathbf{1}_r\varepsilon_{N_y}^x &\le -Px_{N_y} &&\\ -\mathbf{1}_m\varepsilon_{k+1}^u &\le Ru_k, && k = 0, 1, \ldots, N_u - 1\\ -\mathbf{1}_m\varepsilon_{k+1}^u &\le -Ru_k, && k = 0, 1, \ldots, N_u - 1 \end{aligned} \tag{12}$$

represents an upper bound on $J(U, x(t))$, where $\mathbf{1}_j \triangleq [1 \cdots 1]^T \in \mathbb{R}^j$

$$x_k = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \tag{13}$$

and the inequalities (12) hold componentwise. It is easy to prove that the vector $z \triangleq [u_0^T, \ldots, u_{N_u-1}^T, \varepsilon_1^x, \ldots, \varepsilon_{N_y}^x, \varepsilon_1^u, \ldots, \varepsilon_{N_u}^u]^T \in \mathbb{R}^s$, $s \triangleq (m+1)N_u + N_y$, that satisfies (12) and simultaneously minimizes $\varepsilon_1^x + \cdots + \varepsilon_{N_y}^x + \varepsilon_1^u + \cdots + \varepsilon_{N_u}^u$ also solves the original problem (3), i.e., the same optimum $J(U^*(x(t)), x(t))$ is achieved [7], [10]. Therefore, (3) can be reformulated as the following LP problem:

$$\min_z \quad \left\{\varepsilon_1^x + \cdots + \varepsilon_{N_y}^x + \varepsilon_1^u + \cdots + \varepsilon_{N_u}^u\right\} \tag{14a}$$

$$\text{s.t.} \quad -\mathbf{1}_n\varepsilon_k^x \le \pm Q\left[A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{k-1-j}\right] \quad k = 1, \ldots, N_y - 1 \tag{14b}$$

$$-\mathbf{1}_r\varepsilon_{N_y}^x \le \pm P\left[A^{N_y}x(t) + \sum_{j=0}^{N_y-1} A^j B u_{N_y-1-j}\right] \tag{14c}$$

$$-\mathbf{1}_m\varepsilon_{k+1}^u \le \pm Ru_k, \qquad k = 0, \ldots, N_u - 1 \tag{14d}$$

$$y_{\min} \le CA^k x(t) + C\sum_{j=0}^{k-1} A^j B u_{k-1-j} \le y_{\max} \quad k = 1, \ldots, N_c \tag{14e}$$

$$u_{\min} \le u_k \le u_{\max}, \qquad k = 0, \ldots, N_u - 1 \tag{14f}$$

$$u_k = 0, \qquad N_u \le k \le N_y - 1 \tag{14g}$$

where constraints (14b)–(14f) are componentwise, and $\pm$ means that the constraint is duplicated for each sign, as in (12).[2] Problem (14) can be rewritten in the more compact form

$$\begin{aligned}\min_z \quad & f^T z\\ \text{s.t.} \quad & Gz \le S + Fx(t)\end{aligned} \tag{15}$$

where $f \in \mathbb{R}^s$, $G \in \mathbb{R}^{q\times s}$, $F \in \mathbb{R}^{q\times n}$, $S \in \mathbb{R}^q$, $q \triangleq 2(n(N_y - 1) + 2mN_u + pN_c + r)$.

Since problem (15) depends on the current state $x(t)$, in the implementation of MPC one needs to solve the LP (15) on line at each time step. Although efficient LP solvers based on simplex methods or interior point methods are available, computing the input $u(t)$ demands significant online computation effort and control software complexity.

Rather than solving the LP online, we follow the ideas of [16] and propose an approach where all the computation is moved offline. The state feedback control law is defined implicitly by computing the solution of the optimization problem (3), or equivalently (15), as a function of the state vector $x(t)$. Our goal is to make this dependence explicit. In fact, by treating $x(t)$ as a vector of parameters, the LP becomes what is called an mp-LP in the operations research literature [17]–[19].

As will be described in Section III-A, we use the algorithm developed in [19] for solving the mp-LP previously formulated. Once the multiparametric problem (14) has been solved offline for a polyhedral set $X \subset \mathbb{R}^n$ of states, the optimal solution $z^*(x(t))$ of (15) is available explicitly as a piecewise affine function of $x(t)$, and the model predictive controller (3) is also

[2]Note that 1-norm over space requires the introduction of $n(N_y - 1)$ slack variables for the terms $\|Qx_k\|_1$, $\varepsilon_{k,i} \ge \pm Q^{\{i\}}x_k$ $k = 1, 2, \ldots, N_y - 1$, $i = 1, 2, \ldots, n$, plus r slack variables for the terminal penalty $\|Px_{N_y}\|_1$, $\varepsilon_{N_y,i} \ge \pm P^{\{i\}}x_{N_y}$ $i = 1, 2, \ldots, r$, plus mN_u slack variables for the input terms $\|Ru_k\|_1$, $\varepsilon_{k+1,i}^u \ge \pm R^{\{i\}}u_k$ $k = 0, 1, \ldots, N_u - 1$, $i = 1, 2, \ldots, m$.

available explicitly, as the optimal input $u(t)$ consists simply of the first m components of $z^*(x(t))$

$$u(t) = [I_m\ 0\ \dots\ 0] z^*(x(t)). \tag{16}$$

Therefore, the MPC control law has the form

$$u(x) = P_i x + q_i, \quad \text{if} \quad H_i x \le k_i, \qquad i = 1, \dots, N_{\text{mpc}} \tag{17}$$

where $P_i \in \mathbb{R}^{m \times n}$, $q_i \in \mathbb{R}^m$, and the polyhedral sets $X_i \triangleq \{x \in \mathbb{R}^n\colon H_i x \le k_i\}$, $i = 1, \dots, N_{\text{mpc}}$, are a partition of X.

We remark that the implicit form (3) and the explicit form (17) are totally *equal*, and therefore the stability, feasibility, and performance properties mentioned in the previous sections are automatically inherited by the piecewise affine control law (17). Clearly, the explicit form (17) is more advantageous for implementation, and provides insight on the type of action of the controller in different regions X_i of the state space, as will be detailed in Section III-D.

A. Multiparametric Linear Programming

The first method for solving mp-LPs was formulated by Gal and Nedoma [17], and later only a few authors have dealt with multiparametric linear [18], [33], [34], [19], nonlinear [35], quadratic [16], and mixed-integer [36] program solvers.

Multi-parametric programming systematically subdivides the space of parameters into *critical regions* (CRs). A CR is the set of all vectors x of parameters for which a certain combination of constraints is active at the optimizer of problem (15) [37], [33]. For each CR, the optimizer z^* is expressed as a function of x.

In [19], we proposed an iterative algorithm which, rather than visiting different bases of the associated LP tableau [17], uses geometric arguments to directly explore and partition the parameter set X [16]. The resulting algorithm for solving multi-parametric linear programs has computational advantages, namely the simplicity of its implementation in a recursive form and the efficient handling of primal and dual degeneracy.

In order to prove interesting properties of MPC based on LP, we briefly recall here below the main features of mp-LP, by referring the reader to [18], [19] for a more comprehensive description.

Consider again the right-hand side mp-LP (15) and its dual problem

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^q} \quad & (S + Fx)^T \lambda \\ \text{s.t.} \quad & G^T \lambda = f \\ & \lambda \le 0 \end{aligned} \tag{18}$$

where for simplicity we dropped the dependence on t of the state vector x. For a given polyhedral set $X \subseteq \mathbb{R}^n$ of parameters x, $X \triangleq \{x \in \mathbb{R}^n\colon Tx \le v\}$, an mp-LP solver determines the region $X^* \subseteq X$ of parameters x such that the LP (15) is feasible and the optimum is finite, and finds the expression $z^*(x)$ of the optimizer (or one of the optimizers, in case of multiple optima).

For a given $x \in X^*$, let $J^*(x)$ be the optimal value of problem (15) and $Z^*(x)$ be the set of optimizers $z^*(x)$ related to $J^*(x)$. We need the following definition of primal and dual degeneracy.

Definition 1: For a given $x \in X^*$, the LP (15) is said to be *primal degenerate* if there exists a $z^*(x) \in Z^*(x)$ such that the number of active constraints at the optimizer is greater than the number of variables n.

Definition 2: For a given $x \in X^*$ the LP (15) is said to be *dual degenerate* if its dual problem (18) is primal degenerate.

We recall some well-known properties of the optimizer $z^*\colon \mathbb{R}^n \mapsto \mathbb{R}^s$ (when it is uniquely defined), of the value function $J^*\colon \mathbb{R}^n \mapsto \mathbb{R}$, and of the set X^*.

Theorem 3 (cf. [18, p. 179, Th. 2]): Let $X^* \subseteq X$ be the set of parameters $x \in X$ such that the LP (15) is feasible and the optimum $J^*(x) \in \mathbb{R}$ is finite. Then, X^* is a closed polyhedral set in $\mathbb{R}^s$.

Definition 3: A continuous function $h\colon X \mapsto \mathbb{R}^k$, where $X \subseteq \mathbb{R}^s$ is a polyhedral set, is *piecewise affine (PWA)* if there exists a partition of X into convex polyhedra $R_1, \dots, R_N$, and $h(x) = H_i x + k_i, \forall\, x \in R_i, i = 1, \dots, N$.

Theorem 4 (cf. [18, p. 180]): The functions $J^*(\cdot)$ and (if unique[3]) are continuous and piecewise affine over X^*. Moreover, $J^*(\cdot)$ is a convex function over X^*.

Because of (16), the following corollary of Theorem 4 shows that the controller (3) and (4) admits the piecewise affine representation (17).

Corollary 1: The control law $u(x)$, $u\colon \mathbb{R}^n \mapsto \mathbb{R}^m$, defined by the optimization problem (3) and (4) is continuous and piecewise affine.

B. mp-LP Algorithm

The mp-LP algorithm presented in [19] consists of two parts.

1) Determine the minimal dimension $n' \le n$ of the affine subspace that contains X^*. This preliminary step reduces the number of parameters and allows to work with full-dimensional regions.
2) Determine the critical regions and the PWA functions describing the optimum $J^*(x)$ and the optimizer $z^*(x)$.

When the mp-LP algorithm is used to solve the mp-LP (15) generated by the MPC formulation (3), the first step may not be required, because of the following proposition.

Proposition 2: Consider the MPC problem (3) and suppose $u_{\min} < 0 < u_{\max}$, $y_{\min} < 0 < y_{\max}$. Then the set $X^* \subseteq X$ of states (i.e., parameters) $x(t)$ which renders (15) feasible is a full-dimensional subset of $\mathbb{R}^n$.

Proof: We first prove that X^* contains a ball $\mathcal{B}_\varepsilon$ centered in the origin of radius $\varepsilon > 0$, $\mathcal{B}_\varepsilon = \{x \in \mathbb{R}^n\colon \|x\| \le \varepsilon\}$. This is equivalent to show that for each state $x(t) \in \mathcal{B}_\varepsilon$ there exist a feasible control sequence $\{u_k\}_{k=0}^{N_c-1}$ and a corresponding feasible output evolution $\{y_k\}_{k=0}^{N_c}$. To this end, consider a state $x(t) \in \mathcal{B}_\varepsilon$, the input sequence $\{u_k = 0\}_{k=0}^{N_c-1}$, and the corresponding output sequence $\{y_k = CA^k x\}_{k=0}^{N_c}$. Clearly, $u_k = 0$ are feasible inputs. The outputs y_k are also feasible for

$$\varepsilon < \min_{\substack{k=0,\dots,N_c \\ i=1,\dots,p \\ \|C^{\{i\}}A^k\| \neq 0}} \left\{ \min \left\{ \frac{y_{\max}^{\{i\}}}{\left\|C^{\{i\}}A^k\right\|}, \frac{-y_{\min}^{\{i\}}}{\left\|C^{\{i\}}A^k\right\|} \right\} \right\}$$

[3]See Section III-D for details about the case of multiple optima.

where the superscript $\{i\}$ denotes the ith row (or component), and $\|\cdot\|$ the standard Euclidean norm. Hence, for all $x(t) \in \mathcal{B}_\varepsilon$ the LP problem (15) is feasible, i.e., $\mathcal{B}_\varepsilon \subset X^*$. By Theorem 3, X^* is a full-dimensional convex polyhedron of $\mathbb{R}^n$. □

The second part of the algorithm represents the core of the mp-LP algorithm and we refer to [19] for a complete and detailed presentation.

C. Offline Complexity of Explicit MPC Based on LP

An upper-bound to the number N_r of different critical regions that are generated by the mp-LP solver can be found by using the approach of [34], where N_r is shown to be less than or equal to the number μ of extreme points of the feasible region $G^T y = f$, $y \leq 0$ of the dual problem of (16). In the worst case, such a dual polyhedron in $\mathbb{R}^q$ has $s+q$ facets, where s, q are the number of optimization variables and constraints, respectively, in (15). By recalling the result in [38] for computing an upper-bound to the number of extreme points of a polyhedron, we obtain

$$N_r \leq \mu \leq \binom{s+q-\lceil q/2 \rceil}{\lfloor q/2 \rfloor} + \binom{s+q-1-\lceil (q-1)/2 \rceil}{\lfloor (q-1)/2 \rfloor}. \tag{19}$$

In practice, far fewer combinations are usually generated by the mp-LP solver. Furthermore, the gains for the future input moves $u_1, \ldots, u_{N_u-1}$ and slack variables $\varepsilon_1^x, \ldots, \varepsilon_{N_y}^x, \varepsilon_1^u, \ldots, \varepsilon_{N_u}^u$ are not relevant for the control law. Thus, several different combinations of active constraints may lead to the same first m components $u_0^*(x)$ of the solution $z^*(x)$. Indeed, the number $N_{\rm r}$ of regions of the piecewise affine solution of (15) is in general larger than the number $N_{\rm mpc}$ of feedback gains in the MPC law (17), as by post-processing the mp-LP solution two critical regions, where the linear gain is the same, are joined, provided that their union is a convex set [39].

D. Idle Control and Multiple Optima

There are two main issues regarding the implementation of an MPC control law based on linear programming: idle control and multiple solutions. The first corresponds to a control move $u(t)$ which is persistently zero, the second to the degeneracy of the LP problem and the existence of multiple solutions. The approach of this paper allows one to easily recognize both situations.

By analyzing the explicit solution of the MPC law, one can locate immediately the critical regions where the matrices P_i, q_i in (17) are zero, i.e., where the controller provides idle control. A different tuning of the weights is required if such polyhedral regions appear and the overall performance is not satisfactory.

The second issue is the presence of multiple solutions, that might arise from the degeneracy of the dual problem (18). Multiple optima are undesirable, as they might lead to a fast switching between the different optimal control moves when the optimization program (15) is solved online, unless interior-point methods are used. Such behavior can be avoided when the piecewise affine solution (17) is used.

When dual degeneracy occurs there exist critical regions where the optimizer is not uniquely defined. The mp-LP solver [19] can detect such critical regions of degeneracy and partition them into sub-regions were a unique explicit optimizer is defined. This sub-partitioning may led to discontinuity of the optimizer within the dual degenerate regions. The following proposition proves the existence of a continuous optimal control law even in case of dual degeneracy.

Proposition 3: Let CR be a critical region of dual degeneracy. There always exists a polyhedral partition $CR_1, \ldots, CR_{Nd}$ of CR such that the optimizer $z^*(x)$ is affine in each CR_i and continuous in CR.

Proof: The proof follows from results in [18]. See [19, Rem. 4] for details. □

Example IV.3 will illustrate an MPC law where multiple optima and idle control occur.

E. Efficient Computation of mp-LP Solutions

The problem of reducing online computation is crucial, as whenever the number of constraints involved in the optimization problem increases, the number of regions associated with the piecewise affine control mapping may increase exponentially. In [40], an algorithm that efficiently performs the online evaluation of the explicit optimal control law both in terms of storage demands and computational complexity has been presented. Here, we present the main idea.

By Theorem 4, the value function $J^*(\cdot)$ is convex and piecewise affine

$$J^*(x) = T_i^T x + V_i \qquad \forall\, x \in X_i,\ i = 1, \ldots, N_{\rm mpc} \tag{20}$$

where X_i, $i = 1, \ldots, N_{\rm mpc}$, is the polyhedral partition associated with the optimal control law (17). From the equivalence of the representations of piecewise affine convex functions [34], the function $J^*(\cdot)$ in equation (20) can be represented alternatively as

$$J^*(x) = \max_{i=1,\ldots,N_{\rm mpc}} \left\{ T_i^T x + V_i \right\}. \tag{21}$$

By exploiting the equivalence of (20) and (21), the polyhedral region X_j containing a given state $x(t)$ can be simply identified by searching the index i, $i = 1, \ldots, N_{\rm mpc}$ for which $T_i^T x(t) + V_i$ is maximum

$$x(t) \in X_i \Leftrightarrow \left(T_i^T x(t) + V_i\right) = \max_{i=1,\ldots,N_{\rm mpc}} \left\{ T_i^T x(t) + V_i \right\}. \tag{22}$$

Compared to an algorithm for evaluating PWA functions where all the polyhedra are stored in memory and searched sequentially on line, this approach is clearly more efficient both in terms of storage demand and computation complexity, as only the value function must be stored, i.e., $(n+1)N_{\rm mpc}$ real numbers, and as it will give a solution after $nN_{\rm mpc}$ multiplications, $(n-1)N_{\rm mpc}$ sums, and $N_{\rm mpc} - 1$ comparisons, see [40] for details. An alternative approach, although less efficient in terms of memory requirements, was described in [41] and consists of organizing the controller gains of the PWA control law on a balanced search tree, which leads to a $\log N_{\rm mpc}$ average computation complexity.

IV. Examples

Example IV.1: We provide here the explicit solution to the unconstrained MPC regulation example proposed in [12]. The nonminimum phase system

$$y(t) = \frac{s-1}{3s^2+4s+2}\,u(t)$$

is sampled at a frequency of 10 Hz, obtaining the discrete-time state-space model

$$x(t+1) = \begin{bmatrix} 0.872 & -0.0623 \\ 0.0935 & 0.997 \end{bmatrix} x(t) + \begin{bmatrix} 0.0935 \\ 0.004\,78 \end{bmatrix} u(t)$$
$$y(t) = [\,0.333 \quad -1\,].$$

In [12], the authors minimize $\sum_{k=1}^{N_y-1} 5|y_k| + |u_{k-1}|$, with the horizon length $N_y = N_u + 1 = 30$. Such an MPC problem can be rewritten in the form (3), by defining $Q = \left[\begin{smallmatrix} 1.6667 & -5 \\ 0 & 0 \end{smallmatrix}\right]$, $R = 1$ and $P = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$. Note that since Q, P are singular matrices, the sufficient condition for stability of Theorem 1 does not hold. The solution of the mp-LP problem was computed in 20 s by running the mp-LP solver [19] in Matlab on a 450-MHz Pentium III and the corresponding polyhedral partition of the state-space is depicted in Fig. 1(a). The MPC law is

$$u = \begin{cases} 0, & \text{if } \begin{bmatrix} -108.78 & -157.61 \\ 5.20 & 37.97 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \quad \text{(Region \#1)} \\ 0, & \text{if } \begin{bmatrix} -5.20 & -37.97 \\ 108.78 & 157.61 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \quad \text{(Region \#2)} \\ [-10.07 \;\; -14.59]x, & \text{if } \begin{bmatrix} 20.14 & 29.19 \\ 0.09 & 1.86 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \quad \text{(Region \#3)} \\ [-14.55 \;\; -106.21]x, & \text{if } \begin{bmatrix} 29.11 & 212.41 \\ -1.87 & -38.20 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \quad \text{(Region \#4)} \\ [-10.07 \;\; -14.59]x, & \text{if } \begin{bmatrix} -20.14 & -29.19 \\ -0.09 & -1.86 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \quad \text{(Region \#5)} \\ [-14.55 \;\; -106.21]x, & \text{if } \begin{bmatrix} -29.11 & -212.41 \\ 1.87 & 38.20 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \quad \text{(Region \#6)}. \end{cases} \tag{23}$$

In Fig. 1(b), the closed-loop system is simulated from the initial state $x_0 = \left[\begin{smallmatrix} -1.5 \\ -0.2 \end{smallmatrix}\right]$. Note the idle control behavior during the transient.

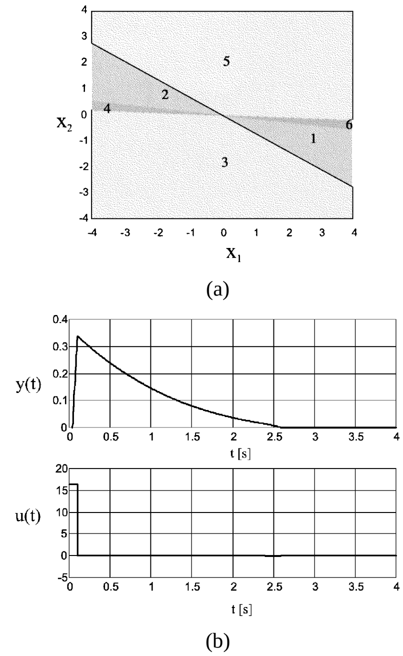


Fig. 1. Example IV.1 with terminal weight $P = 0$.

The same problem is solved by slightly perturbing $Q = \left[\begin{smallmatrix} 1.6667 & -5 \\ 0 & 0.0001 \end{smallmatrix}\right]$ so that it becomes nonsingular, and by adding the terminal weight

$$P = \begin{bmatrix} -705.3939 & -454.5755 \\ 33.2772 & 354.7107 \end{bmatrix} \tag{24}$$

which is computed as shown in Proposition 1 ($\tilde{P} = \left[\begin{smallmatrix} -1.934 & -1.246 \\ 0.0912 & 0.972 \end{smallmatrix}\right]$, $H = \left[\begin{smallmatrix} 0.9345 & 0.0441 \\ -0.0441 & 0.9345 \end{smallmatrix}\right]$, $\rho = 7.8262$, $\sigma = 0.021448$). The explicit solution was computed in 80 s, consists of 44 regions and is depicted in Fig. 2(a).

In Fig. 2(b), the closed-loop system is simulated from the initial state $x_0 = \left[\begin{smallmatrix} -1.5 \\ -0.2 \end{smallmatrix}\right]$.

Example IV.2: Consider the double integrator

$$y(t) = \frac{1}{s^2}\,u(t) \tag{25}$$

and its equivalent discrete-time state-space representation

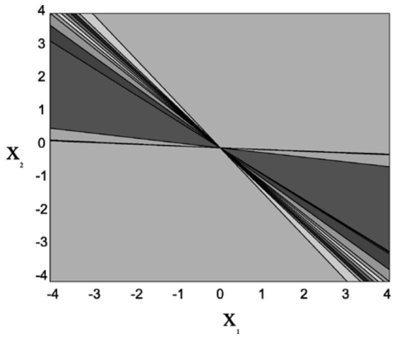
$$\begin{cases} x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = [\,\frac{1}{2} \quad \frac{1}{2}\,]\, x(t) \end{cases} \tag{26}$$

obtained by setting

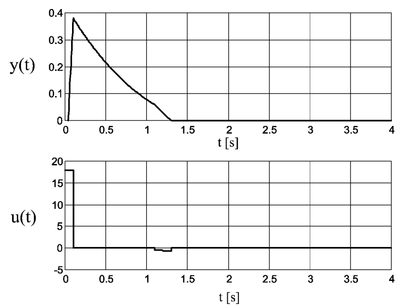
$$\ddot{y}(t) \approx \frac{\dot{y}(t+T) - \dot{y}(t)}{T} \qquad \dot{y}(t) \approx \frac{y(t+T) - y(t)}{T}$$

$T = 1$ s. System (26) and the explicit form of the corresponding constrained linear quadratic regulator was investigated in [16]. Here, instead we want to regulate the system to the origin while minimizing the performance measure

$$\sum_{k=0}^{1} \left\| \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{k+1} \right\|_\infty + |0.8u_k| \tag{27}$$

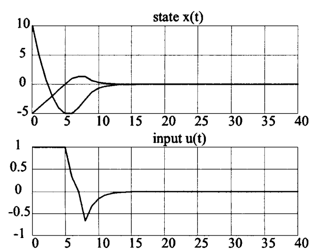


(a)

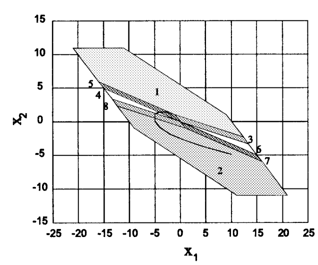


(b)

Fig. 2. (a) Example IV.1 with terminal weight P as in (24). (b) Simulation of the explicit controller and the comparison with LQR control based on the same weights Q, R.



(a)



(b)

Fig. 3. Example IV.2: closed-loop simulation and polyhedral partition of the MPC law.

subject to the input constraints

$$-1 \le u_k \le 1, \qquad k = 0, 1 \tag{28}$$

and the state constraints

$$-10 \le x_k \le 10, \qquad k = 1, 2. \tag{29}$$

This task is addressed by using the MPC algorithm (3) and (4) where $N_y = 2$, $N_u = 2$, $Q = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.8$. The solution of the mp-LP problem was computed in 13.57 s and the corresponding polyhedral partition of the state-space is depicted in Fig. 3(b). The resulting MPC law is

$$u = \begin{cases} -1.00, & \\ \quad \text{if} \begin{bmatrix} 1.00 & 2.00 \\ 0.00 & 1.00 \\ -1.00 & -1.00 \\ -0.80 & -3.20 \\ 1.00 & 1.00 \\ -1.00 & -3.00 \end{bmatrix} x \le \begin{bmatrix} 11.00 \\ 11.00 \\ 10.00 \\ -2.40 \\ 10.00 \\ -2.00 \end{bmatrix} & \\ \quad (\text{Region } \#1) & \\ 1.00, & \\ \quad \text{if} \begin{bmatrix} 0.80 & 3.20 \\ -1.00 & -2.00 \\ -1.00 & -1.00 \\ 1.00 & 1.00 \\ 0.00 & -1.00 \\ 1.00 & 3.00 \end{bmatrix} x \le \begin{bmatrix} -2.40 \\ 11.00 \\ 10.00 \\ 10.00 \\ 11.00 \\ -2.00 \end{bmatrix} & \\ \quad (\text{Region } \#2) & \\ [-0.33 \ \ -1.33]x, & \\ \quad \text{if} \begin{bmatrix} 0.53 & 2.13 \\ 0.67 & 0.67 \\ -1.00 & -1.00 \\ -0.33 & -1.33 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 1.00 \end{bmatrix} & \\ \quad (\text{Region } \#3) & \\ 0, & \\ \quad \text{if} \begin{bmatrix} -0.80 & -3.20 \\ 1.00 & 3.00 \\ -1.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} & \\ \quad (\text{Region } \#4) & \\ [-0.50 \ \ -1.50]x, & \\ \quad \text{if} \begin{bmatrix} -1.00 & -1.00 \\ 0.50 & 0.50 \\ -0.80 & -2.40 \\ 0.50 & 1.50 \end{bmatrix} x \le \begin{bmatrix} 10.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{bmatrix} & \\ \quad (\text{Region } \#5) & \\ 0, & \\ \quad \text{if} \begin{bmatrix} 0.80 & 3.20 \\ -1.00 & -3.00 \\ 1.00 & 1.00 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} & \\ \quad (\text{Region } \#6) & \\ [-0.50 \ \ -1.50]x, & \\ \quad \text{if} \begin{bmatrix} 1.00 & 1.00 \\ -0.50 & -0.50 \\ 0.80 & 2.40 \\ -0.50 & -1.50 \end{bmatrix} x \le \begin{bmatrix} 10.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{bmatrix} & \\ \quad (\text{Region } \#7) & \\ [-0.33 \ \ -1.33]x, & \\ \quad \text{if} \begin{bmatrix} -0.53 & -2.13 \\ -0.67 & -0.67 \\ 1.00 & 1.00 \\ 0.33 & 1.33 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 1.00 \end{bmatrix} & \\ \quad (\text{Region } \#8). & \end{cases}$$

Note that regions #1 and #2 correspond to the saturated controller, and regions #4 and #6 to idle control. The same example

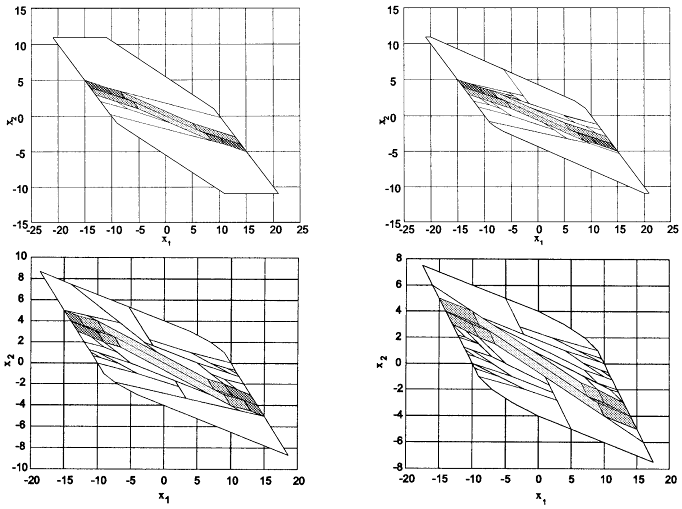


Fig. 4. Example IV.2, MPC control law: partition of the state–space for increasing input horizon N_u.

TABLE I
OFF-LINE COMPUTATION TIMES AND NUMBER OF REGIONS N_{mpc} IN THE MPC CONTROL LAW (17) FOR THE DOUBLE INTEGRATOR EXAMPLE

Free moves N_u	Computation time (s)	N. of regions N_{mpc}
2	13.57	8
3	28.50	16
4	48.17	28
5	92.61	37
6	147.53	44

was solved for an increasing number of degrees of freedom N_u. The corresponding polyhedral partitions are reported in Fig. 4. Note that the white regions correspond to the saturated controller $u(t) = -1$ in the upper part, and $u(t) = -1$ in the lower part. The offline computation times and number of regions N_{mpc} in the MPC control law (17) are reported in Table I.

Example IV.3: Consider again the double integrator of Example IV.2, along with the optimization problem

$$\min_{u_0} \left\| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_1 \right\|_\infty + |u_0| \tag{30}$$

subject to constraints (28) and (29). The associated mp-LP problem is

$$\min_{\varepsilon_1,\varepsilon_2,u_0} \quad \varepsilon_1+\varepsilon_2$$
$$\text{s.t.} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ u_0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 10 \\ 10 \\ 10 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ -1 & -1 \\ 0 & -1 \\ 0 & -1 \\ -1 & -1 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} x(t). \tag{31}$$

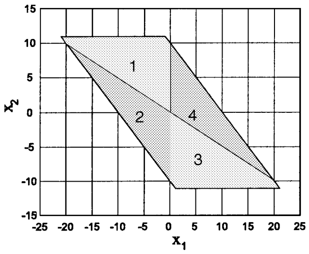


Fig. 5. Polyhedral partition associated with problems (28)–(30).

The solution of (31) was computed in 0.5 s and the corresponding polyhedral partition of the state–space is depicted in Fig. 5. The MPC law is

$$u = \begin{cases} \text{degenerate}, & \text{if } \begin{bmatrix} -1.00 & -2.00 \\ 1.00 & 0.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} \\ & (\text{Region } \#1) \\ 0, & \text{if } \begin{bmatrix} 1.00 & 0.00 \\ 1.00 & 2.00 \\ -1.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & (\text{Region } \#2) \\ \text{degenerate}, & \text{if } \begin{bmatrix} -1.00 & 0.00 \\ 1.00 & 2.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} \\ & (\text{Region } \#3) \\ 0, & \text{if } \begin{bmatrix} -1.00 & -2.00 \\ -1.00 & 0.00 \\ 1.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & (\text{Region } \#4). \end{cases}$$

Note the presence of idle control and multiple optima in regions #2, #4 and #1, #3, respectively. The algorithm in [19] returns two possible subpartitions of the degenerate regions #1, #3. Region #1 can be partitioned either as

$$u_{1A} = \begin{cases} 0, & \text{if } \begin{bmatrix} -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & (\text{Region } \#1\text{a}) \\ [0 \;\; -1]\, x + 10, & \text{if } \begin{bmatrix} 0.00 & -2.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} -20.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} \\ & (\text{Region } \#1\text{b}) \end{cases}$$

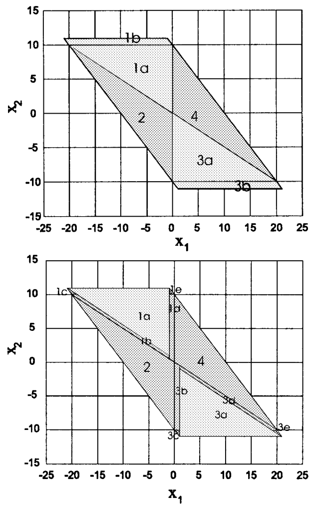


Fig. 6. Example IV.3: example of degeneracy.

or

$$u_{1B} = \begin{cases} -1.00, & \text{if} \begin{bmatrix} -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \le \begin{bmatrix} -1.00 \\ -1.00 \\ 11.00 \end{bmatrix} \\ & \text{(Region \#1a)} \\ 0, & \text{if} \begin{bmatrix} 1.00 & 2.00 \\ -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#1b)} \\ [0 \;\; -1]\, x + 10, & \text{if} \begin{bmatrix} 1.00 & 2.00 \\ 0.00 & -2.00 \\ -1.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#1c)} \\ 0, & \text{if} \begin{bmatrix} -1.00 & 0.00 \\ -1.00 & -2.00 \\ 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ -1.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#1d)} \\ [0 \;\; -1]\, x + 10, & \text{if} \begin{bmatrix} -1.00 & 0.00 \\ 0.00 & -2.00 \\ 1.00 & 1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#1e).} \end{cases}$$

Region #3 can be partitioned symmetrically either as

$$u_{2A} = \begin{cases} 0, & \text{if} \begin{bmatrix} -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#3a)} \\ [0 \;\; -1]\, x - 10, & \text{if} \begin{bmatrix} 0.00 & 2.00 \\ 1.00 & 1.00 \\ -1.00 & -1.00 \\ 0.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} -20.00 \\ 10.00 \\ 10.00 \\ 11.00 \end{bmatrix} \\ & \text{(Region \#3b)} \end{cases}$$

or

$$u_{2B} = \begin{cases} 1.00, & \text{if} \begin{bmatrix} -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} -1.00 \\ -1.00 \\ 11.00 \end{bmatrix} \\ & \text{(Region \#3a)} \\ 0, & \text{if} \begin{bmatrix} 1.00 & 0.00 \\ -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#3b)} \\ [0 \;\; -1]\, x - 10, & \text{if} \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 2.00 \\ -1.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#3c)} \\ 0, & \text{if} \begin{bmatrix} -1.00 & -2.00 \\ -1.00 & 0.00 \\ 1.00 & 2.00 \\ 0.00 & -1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ -1.00 \\ 0.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#3d)} \\ [0 \;\; -1]\, x - 10, & \text{if} \begin{bmatrix} -1.00 & -2.00 \\ 0.00 & 2.00 \\ 1.00 & 1.00 \end{bmatrix} x \le \begin{bmatrix} 1.00 \\ -20.00 \\ 10.00 \end{bmatrix} \\ & \text{(Region \#3e).} \end{cases}$$

As a consequence, two possible explicit solutions to problem (30) are depicted in Fig. 6. Note that the optimal control law corresponding to the choice of u_{1A} and u_{2A} is continuous with respect to x.

V. Conclusion

In this paper, we formulated a model predictive controller based on a $1/\infty$-norm performance objective for linear systems subject to input and output constraints, and gave conditions on the weighting matrices for closed-loop stability. We also provided the explicit representation of such an MPC control law, and showed that it is a piecewise affine function of the state vector. The basic setup can be easily extended to trajectory following, suppression of measured disturbances, and time-varying constraints, and to MPC of linear systems with a performance index expressed by any combination of 1- and ∞-norms. In fact, any combination leads to a linear program, which can be solved multiparametrically by using the results of Section III-A. The approach can also be extended for solving explicitly optimal control/MPC problems for hybrid systems [42], as shown in [43], and for solving min–max constrained control of systems affected by norm-bounded input disturbances and/or polyhedral parametric uncertainties in the state-space matrices [44].

Acknowledgment

The authors would like to thank D. R. Saffer III for pointing out a mistake in Example IV.1 in an earlier version of this paper.

REFERENCES

[1] S. J. Qin and T. A. Badgwell, "An overview of industrial model predictive control technology," in *Chemical Process Control—V*: AIChe Symp. Series—Amer. Inst. Chem. Eng., 1997, vol. 93, pp. 232–256.

[2] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control-application to industrial processes," *Automatica*, vol. 14, pp. 413–428, 1978.

[3] C. R. Cutler and B. C. Ramaker, "Dynamic matrix control—A computer control algorithm," in *Proc. Amer. Control Conf.*, vol. WP5-B, San Francisco, CA, 1980.

[4] C. E. García and A. M. Morshedi, "Quadratic programming solution of dynamic matrix control (QDMC)," *Chem. Eng. Commun.*, vol. 46, pp. 73–87, 1986.

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

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

[7] L. A. Zadeh and L. H. Whalen, "On optimal control and linear programming," *IEEE Trans. Automat. Contr.*, vol. AC-7, pp. 45–46, Jan. 1962.

[8] A. I. Propoi, "Use of linear programming methods for synthesizing sampled-data automatic systems," *Automat. Rem. Control*, vol. 24, no. 7, pp. 837–844, 1963.

[9] T. S. Chang and D. E. Seborg, "A linear programming approach for multivariable feedback control with inequality constraints," *Int. J. Control*, vol. 37, no. 3, pp. 583–597, 1983.

[10] P. J. Campo and M. Morari, "Model predictive optimal averaging level control," *AIChE J.*, vol. 35, no. 4, pp. 579–591, 1989.

[11] H. Genceli and M. Nikolaou, "Robust stability analysis of constrained ℓ_1-norm model predictive control," *AIChE J.*, vol. 39, no. 12, pp. 1954–1965, 1993.

[12] C. V. Rao and J. B. Rawlings, "Linear programming and model predictive control," *J. Process Control*, vol. 10, pp. 283–289, 2000.

[13] P. O. Gutman, "A linear programming regulator applied to hydroelectric reservoir level control," *Automatica*, vol. 22, no. 5, pp. 533–541, 1986.

[14] P. O. Gutman and M. Cwikel, "Admissible sets and feedback control for discrete-time linear dynamical systems with bounded control and states," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 373–376, Apr. 1986.

[15] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 428–433, Feb. 1994.

[16] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[17] T. Gal and J. Nedoma, "Multiparametric linear programming," *Manag. Sci.*, vol. 18, pp. 406–442, 1972.

[18] T. Gal, *Postoptimal Analyzes, Parametric Programming, and Related Topics*, 2nd ed. Berlin, Germany: de Gruyter, 1995.

[19] F. Borrelli, A. Bemporad, and M. Morari, "A geometric algorithm for multi-parametric linear programming," *J. Optim. Theory*, to be published.

[20] J. B. Rawlings and K. R. Muske, "The stability of constrained receding-horizon control," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1512–1516, 1993.

[21] A. Zheng and M. Morari, "Stability of model predictive control with mixed constraints," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1818–1823, 1995.

[22] S. S. Keerthi and E. G. Gilbert, "Optimal infinite-horizon feedback control laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," *J. Opt. Theory Appl.*, vol. 57, pp. 265–293, 1988.

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

[24] A. Bemporad, "Reducing conservativeness in predictive control of constrained systems with disturbances," in *Proc. 37th IEEE Conf. Decision Control*, Tampa, FL, 1998, pp. 1384–1391.

[25] E. C. Kerrigan and J. M. Maciejowski, "Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control," in *Proc. 39th IEEE Conf. Decision Control*, 2000, pp. 4951–4956.

[26] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*, A. Garulli, A. Tesi, and A. Vicino, Eds. New York: Springer-Verlag, 1999, no. 245 in Lecture Notes in Control and Information Sciences, pp. 207–226.

[27] D. Q. Mayne and H. Michalska, "Robust receding horizon control of constrained nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1623–1633, Nov. 1993.

[28] A. Bemporad, L. Chisci, and E. Mosca, "On the stabilizing property of SIORHC," *Automatica*, vol. 30, no. 12, pp. 2013–2015, 1994.

[29] G. Bitsoris, "On the positive invariance of polyhedral sets for discrete-time systems," *Syst. Control Lett.*, vol. 11, pp. 243–248, 1988.

[30] A. Polański, "On infinity norms as Lyapunov functions for linear systems," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1270–1273, July 1995.

[31] H. Kiendl, J. Adamy, and P. Stelzner, "Vector norms as Lyapunov functions for linear systems," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 839–842, June 1992.

[32] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, pp. 1747–1767, Nov. 1999.

[33] C. Filippi, "On the geometry of optimal partition sets in multiparametric linear programming," Dept. Pure Appl. Math., Univ. Padova, Padova, Italy, Tech. Rep. 12, June 1997.

[34] M. Schechter, "Polyhedral functions and multiparametric linear programming," *J. Optim. Theory Appl.*, vol. 53, pp. 269–280, May 1987.

[35] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. London, U.K.: Academic, 1983.

[36] V. Dua and E. N. Pistikopoulos, "An algorithm for the solution of multiparametric mixed integer linear programming problems," *Ann. Oper. Res.*, pp. 123–139, 2000.

[37] I. Adler and R. D. C. Monteiro, "A geometric view of parametric linear programming," *Algorithmica*, vol. 8, no. 2, pp. 161–176, 1992.

[38] J. E. Goodman and J. O'Rourke, Eds., *Handbook of Discrete and Computational Geometry*. Boca Raton, FL: CRC Press, 1997, Discrete Mathematics and Its Applications.

[39] A. Bemporad, K. Fukuda, and F. D. Torrisi, "Convexity recognition of the union of polyhedra," *Comput. Geometry*, vol. 18, pp. 141–154, 2001.

[40] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, "Efficient on-line computation of constrained optimal control laws," in *Proc. 40th IEEE Conf. Decision Control*, Orlando, FL, 2001, pp. 1187–1192.

[41] P. Tøndel, T. A. Johansen, and A. Bemporad, "Computation and approximation of piecewise affine control via binary search trees," *Automatica*, to be published.

[42] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, Mar. 1999.

[43] A. Bemporad, F. Borrelli, and M. Morari, "Piecewise linear optimal controllers for hybrid systems," in *Proc. Amer. Control Conf.*, Chicago, IL, 2000, pp. 1190–1194.

[44] A. Bemporad, F. Borrelli, and M. Morari, "Min–max control of constrained uncertain discrete-time linear systems," *IEEE Trans. Automat. Contr.*, to be published.

Alberto Bemporad received the M.S. degree in electrical engineering and the Ph.D. in control engineering the University of Florence, Florence, Italy, in 1993 and 1997, respectively.

He spent the academic year 1996–1997 at the Center for Robotics and Automation, Department of Systems Science and Mathematics, Washington University, St. Louis, MO, as a Visiting Researcher. In 1997–1999, he held a Postdoctoral position at the Automatic Control Laboratory, ETH, Zurich, Switzerland, where he later became a Senior Researcher. Since 1999, he has been an Assistant Professor at the University of Siena, Siena, Italy. He has published papers in the area of hybrid systems, model predictive control, computational geometry, and robotics. He is coauthoring the new version of the *Model Predictive Control Toolbox* (Natick, MA: The Mathworks, Inc.).

Dr. Bemporad received the IEEE Center and South Italy section "G. Barzilai" and the Italian Electrical Association (AEI) "R. Mariani" awards. He is an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and Chair of the IEEE Control Systems Society Technical Committee on Hybrid Systems.

Francesco Borrelli was born in Milan, Italy, in 1974. He received the Laurea degree in computer science engineering from the University of Naples "Federico II," Naples, Italy, and the Ph.D. degree from the ETH Zurich, Switzerland, in 1998 and 2002, respectively.

He is currently a Research Assistant at the Automatic Control Laboratory of the ETH Zurich. His research interests include constrained optimal control, model predictive control, robust control, parametric programming, singularly perturbed systems, and automotive applications of automatic control.

Manfred Morari received the Diploma from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, and the Ph.D. degree from the University of Minnesota, Minneapolis, in 1974 and 1977, respectively.

In 1994, he was appointed Head of the Automatic Control Laboratory at ETH. Before that, he was the McCollum–Corcoran Professor and Executive Officer for Control and Dynamical Systems at the California Institute of Technology, Pasadena. He has held appointments with Exxon Research & Engineering and ICI, and has consulted internationally for a number of major corporations. His interests are in hybrid systems and the control of biomedical systems.

Dr. Morari has received numerous awards in recognition of his research, among them the Eckman Award of the AACC, the Colburn Award, the Professional Progress Award, and the Computing and System Technology Division Award of the AIChE. He was elected to the National Academy of Engineering (U.S.) in 1993.