

A Survey on Explicit Model Predictive Control^{*}

Alessandro Alessio and Alberto Bemporad

Abstract. Explicit model predictive control (MPC) addresses the problem of removing one of the main drawbacks of MPC, namely the need to solve a mathematical program on line to compute the control action. This computation prevents the application of MPC in several contexts, either because the computer technology needed to solve the optimization problem within the sampling time is too expensive or simply infeasible, or because the computer code implementing the numerical solver causes software certification concerns, especially in safety critical applications.

Explicit MPC allows one to solve the optimization problem off-line for a given range of operating conditions of interest. By exploiting multiparametric programming techniques, explicit MPC computes the optimal control action off line as an “explicit” function of the state and reference vectors, so that on-line operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases, so that the MPC controller maps into a lookup table of linear gains.

In this paper we survey the main contributions on explicit MPC appeared in the scientific literature. After recalling the basic concepts and problem formulations of MPC, we review the main approaches to solve explicit MPC problems, including a novel and simple suboptimal practical approach to reduce the complexity of the explicit form. The paper concludes with some comments on future research directions.

Keywords: Model predictive control, explicit solutions, multiparametric programming, piecewise affine controllers, hybrid systems, min-max control.

1 Model Predictive Control

In Model Predictive Control (MPC) the control action is obtained by solving a finite horizon open-loop optimal control problem at each sampling instant. Each

Alessandro Alessio and Alberto Bemporad

Department of Information Engineering, University of Siena, Italy

e-mail: {alessio, bemporad}@dii.unisi.it

^{*} This work was partially supported by the European Commission under the HYCON Network of Excellence, contract number FP6-IST-511368.

optimization yields a sequence of optimal control moves, but only the first move is applied to the process: At the next time step, the computation is repeated over a shifted time-horizon by taking the most recently available state information as the new initial condition of the optimal control problem. For this reason, MPC is also called *receding* or *rolling horizon control*.

The solution relies on a dynamic model of the process, respects all input and output (state) constraints, and optimizes a performance index. This is usually expressed as a quadratic or a linear criterion, so that, for linear prediction models, the resulting optimization problem can be cast as a quadratic program (QP) or linear program (LP), respectively, while for hybrid prediction models, the resulting optimization problem can be cast as a mixed integer quadratic or linear program (MIQP/MILP), as will be reviewed in the next sections. The main difference between MPC and conventional control is therefore that in the latter the control function is pre-computed off-line. The reason for the success of MPC in industrial applications is due to its ability to handle processes with many manipulated and controlled variables and constraints on them in a rather systematic manner.

The process to be controlled is usually modeled by the system of difference equations

$$x(t+1) = f(x(t), u(t)) \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, and $u(t) \in \mathbb{R}^m$ is the input vector. We assume for simplicity that $f(0,0) = 0$. The control and state sequences are requested to satisfy the constraints

$$x(t) \in \mathcal{X}, u(t) \in \mathcal{U} \quad (2)$$

where $\mathcal{U} \subseteq \mathbb{R}^m$ and $\mathcal{X} \subseteq \mathbb{R}^n$ are closed sets containing the origin in their interior¹. Assuming that the control objective is to steer the state to the origin, MPC solves the constrained regulation problem as follows. Assume that a full measurement of the state $x(t)$ is available at the current time t . Then, the following finite-horizon optimal regulation problem is solved

$$\mathcal{P}_N(x(t)) : \min_z \sum_{k=0}^{N-1} l(x_k, u_k) + F(x_N) \quad (3a)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k), \quad k = 0, \dots, N-1 \quad (3b)$$

$$x_0 = x(t), \quad (3c)$$

$$u_k \in \mathcal{U}, \quad k = 0, \dots, N_u - 1 \quad (3d)$$

$$x_k \in \mathcal{X}, \quad k = 1, \dots, N-1 \quad (3e)$$

$$x_N \in \mathcal{X}_N, \quad (3f)$$

$$u_k = \kappa(x_k), \quad k = N_u, \dots, N-1 \quad (3g)$$

where $z \in \mathbb{R}^\ell$, is the vector of optimization variables, $z = [u'_0 \dots u'_{N_u-1}]'$, $\ell = mN_u$ (more generally, z includes command inputs and additional optimization variables),

¹ Mixed constraints on (x, u) can be treated as well, for instance to handle constraints on outputs with direct feedthrough $y(t) = f_y(x(t), u(t))$.

and the choice of the closed set $\mathcal{X}_N \subseteq \mathcal{X}$, terminal cost F , and terminal gain κ ensure closed-loop stability of the MPC scheme [54]. At each time-step t , x_k denotes the predicted state vector at time $t+k$, obtained by applying the input sequence u_0, \dots, u_{k-1} to model (1), starting from $x_0 = x(t)$. The number $N > 0$ is the prediction horizon, N_u is the input horizon ($1 \leq N_u \leq N$), and “ \leq ” denotes component-wise inequalities. Because N is finite, if f , l and F are continuous and \mathcal{U} is also compact the minimum in (3a) exists. At each time-step t a solution to problem $\mathcal{P}_N(x(t))$ is found by solving the mathematical program

$$\begin{aligned} \min_{z} \quad & h(z, x(t)) \\ \text{s.t.} \quad & g(z, x(t)) \leq 0, \quad g \in \mathbb{R}^q \end{aligned} \quad (4)$$

obtained from (3), yielding the optimal control sequence $z^*(x(t))$. Only the first input is applied to system (1)

$$u(t) = u_0^*(x(t)) \quad (5)$$

and the optimization problem (3) is repeated at time $t+1$, based on the new state $x(t+1)$.

The basic MPC setup (3) can be specialized to different cases, depending on the prediction model, performance index, and terminal conditions used.

1.1 Linear Model and Quadratic Cost

A finite-horizon optimal control problem (3) with quadratic stage costs is formulated by setting

$$l(x_k, u_k) = x_k' Q x_k + u_k' R u_k, \quad F(x_N) = x_N' P x_N \quad (6)$$

in (3a), where $Q = Q' \geq 0$, $R = R' > 0$, and $P = P' \geq 0$ are weight matrices of appropriate dimensions. Let (3b) be a deterministic linear discrete-time prediction model

$$f(x_k, u_k) = A x_k + B u_k \quad (7)$$

$\kappa(x) = Kx$ in (3g), \mathcal{U} , \mathcal{X} be polyhedral sets, for example $\mathcal{U} = \{u \in \mathbb{R}^m : u_{\min} \leq u \leq u_{\max}\}$, $\mathcal{X} = \{x \in \mathbb{R}^n : x_{\min} \leq x \leq x_{\max}\}$, and also \mathcal{X}_N be polyhedral. Then, by substituting $x_k = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$, problem (4) becomes a quadratic program (QP):

$$h(z, x(t)) = \frac{1}{2} z' H z + x'(t) C' z + \frac{1}{2} x'(t) Y x(t) \quad (8a)$$

$$g(z, x(t)) = G z - W - S x(t), \quad (8b)$$

where $H = H' > 0$ and C , Y , G , W , S are matrices of appropriate dimensions [18]. Note that Y is not really needed to compute $z^*(x(t))$, it only affects the optimal value of (8a).

1.2 Linear Model and Linear Cost

Let ∞ - or 1-norms be used to measure performance

$$l(x_k, u_k) = \|Qx_k\|_p + \|Ru_k\|_p, \quad F(x_N) = \|Px_N\|_p, \quad p = 1, \infty \quad (9)$$

where $R \in \mathbb{R}^{n_R \times m}$, $Q \in \mathbb{R}^{n_Q \times n}$, $P \in \mathbb{R}^{n_P \times n}$, and use the same setup as described in Section 1.1. In case of ∞ -norms, by introducing auxiliary variables $\varepsilon_0^u, \dots, \varepsilon_{N-1}^u, \varepsilon_1^x, \dots, \varepsilon_N^x$ satisfying $\varepsilon_k^u \geq \|Ru_k\|_\infty$ ($k = 0, \dots, N_u - 1$), $\varepsilon_k^x \geq \|Qx_k\|_\infty$ ($k = 1, \dots, N - 1$), $\varepsilon_N^x \geq \|Px_N\|_\infty$, or, equivalently,

$$\begin{aligned} \varepsilon_k^u &\geq \pm R^i u_k, \quad i = 1, \dots, n_R, \quad k = 0, \dots, N_u - 1 \\ \varepsilon_k^x &\geq \pm Q^i x_k, \quad i = 1, \dots, n_Q, \quad k = 1, \dots, N - 1 \\ \varepsilon_N^x &\geq \pm P^i x_N, \quad i = 1, \dots, n_P \end{aligned} \quad (10)$$

where the superscript i in (10) denotes the i th row, problem (3) can be mapped into the linear program (LP) [12]

$$h(z, x(t)) = \underbrace{[1 \dots 1]}_{N_u-1} \alpha \underbrace{[1 \dots 1]}_N \underbrace{[0 \dots 0]}_{mN_u} z \quad (11a)$$

$$g(z, x(t)) = Gz - W - Sx(t), \quad (11b)$$

where $\alpha = N - N_u + 1$ is the number of times the last input move is repeated over the prediction horizon, $z \triangleq [\varepsilon_0^u \dots \varepsilon_{N_u-1}^u \varepsilon_1^x \dots \varepsilon_N^x u'_0 \dots u'_{N_u-1}]'$ is the optimization vector, and G, W, S are obtained from weights Q, R, P , model matrices A, B , (10), constraint sets $\mathcal{W}, \mathcal{X}, \mathcal{X}_N$, and gain K . The case of 1-norms can be treated similarly by introducing slack variables $\varepsilon_{ik}^u \geq \pm R^i u_k$, $\varepsilon_{ik}^x \geq \pm Q^i x_k$, $\varepsilon_N^x \geq \pm P^i x_N$.

Note that the above reformulation extends beyond $1/\infty$ -norms to *any* convex piecewise affine cost l, F , that, thanks to the result of [62], can be rewritten as the max of a finite set of affine functions. The use of linear programming in optimization-based control dates back to the early sixties [60].

1.3 Linear Uncertain Model and Min-Max Costs

Robust MPC formulations [73] explicitly take into account uncertainties in the prediction model

$$f(x_k, u_k, w_k, v_k) = A(w_k)x_k + B(w_k)u_k + E v_k \quad (12)$$

where

$$\begin{aligned} A(w) &= A_0 + \sum_{i=1}^q A_i w_i, \quad B(w) = B_0 + \sum_{i=1}^q B_i w_i \\ w_k &\in \mathcal{W} \in \mathbb{R}^{n_w}, \quad v_k \in \mathcal{V} \in \mathbb{R}^{n_v} \end{aligned}$$

Let v_k, w_k be modeled as unknown but bounded exogenous disturbances and parametric uncertainties, respectively, and \mathcal{W}, \mathcal{V} be polytopes. A robust MPC strategy often used consists of solving a *min-max* problem that minimize the worst-case

performance while enforcing input and state constraints for all possible disturbances. The following min-max control problem is referred to as *open-loop constrained robust optimal control problem* (OL-CROC) [13]

$$\begin{aligned} \min_{u_0, \dots, u_{N-1}} \quad & \left\{ \max_{\substack{v_0, \dots, v_{N-1} \in \mathcal{V} \\ w_0, \dots, w_{N-1} \in \mathcal{W}}} \sum_{k=0}^{N-1} l(x_k, u_k) + F(x_N) \right\} \\ \text{s.t.} \quad & \text{dynamics (12)} \\ & (3d), (3e), (3f) \text{ satisfied } \forall v_k \in \mathcal{V}, \forall w_k \in \mathcal{W} \end{aligned} \quad (13)$$

The min-max problem (12)–(13) can be solved via linear programming if 1- or ∞ -norms are used [24, 4], or by quadratic programming if quadratic costs are used [56]. Being based on an *open-loop* prediction, in some cases the approach can be quite conservative. It is possible to reformulate the robust MPC problem using a *closed-loop* prediction scheme as described in [63], whose approach reminds the methods used in multi-stage stochastic optimization based on scenario trees. An alternative method in which the min and max problems are interleaved and dynamic programming is used is described in [13] to solve the (*closed-loop constrained robust optimal control problem*, CL-CROC):

For $j = N - 1, \dots, 0$ solve

$$\begin{aligned} J_j^*(x_j) &\triangleq \min_{u_j \in \mathcal{U}} J_{\max, j}(x_j, u_j) \\ \text{s.t. } & A(w_j)x_j + B(w_j)u_j + Ev_j \in \mathcal{X}^{j+1}, \forall v_j \in \mathcal{V}, \forall w_j \in \mathcal{W} \end{aligned}$$

where

$$J_{\max, j}(x_j, u_j) \triangleq \max_{\substack{v_j \in \mathcal{V} \\ w_j \in \mathcal{W}}} \{l(x_j, u_j) + J_{j+1}^*(A(w_j)x_j + B(w_j)u_j + Ev_j)\} \quad (14)$$

and

$$\mathcal{X}^j = \{x \in \mathcal{X} : \exists u \in \mathcal{U} \text{ such that } A(w)x + B(w)u + Ev \in \mathcal{X}^{j+1}, \forall v \in \mathcal{V}, w \in \mathcal{W}\} \quad (15)$$

with boundary condition

$$J_N^*(x_N) = \|Px_N\|_p. \quad (16)$$

1.4 Hybrid Model and Linear or Quadratic Costs

The MPC setup also extends to the case in which (1) is a hybrid dynamical model. When the hybrid dynamics and possible mixed linear/logical constraints on discrete and continuous input and state variables are modeled using the language HYSDEL [71], (3b) can be automatically transformed into the set of linear equalities and inequalities

$$f(x_k, u_k, \delta_k, \zeta_k) = Ax_k + B_1 u_k + B_2 \delta_k + B_3 \zeta_k \quad (17a)$$

$$E_2 \delta_k + E_3 \zeta_k \leq E_1 u_k + E_4 x_k + E_5, \quad (17b)$$

involving both real and binary variables, denoted as the Mixed Logical Dynamical (MLD) model, where $x_k \in \mathbb{R}^{n_c} \times \{0, 1\}^{n_b}$ is the state vector, $u_k \in \mathbb{R}^{m_c} \times \{0, 1\}^{m_b}$ is the input vector, and $\zeta_k \in \mathbb{R}^{r_c}$, $\delta_k \in \{0, 1\}^{r_b}$ are auxiliary variables implicitly defined by (17b) for any given pair (x_k, u_k) . Matrices A , B_i , ($i = 1, 2, 3$), and E_i ($i = 1, \dots, 5$) denote real constant matrices, and inequalities (17b) must be interpreted component-wise.

The associated finite-horizon optimal control problem based on quadratic costs takes the form (8) with $z = [u'_0 \dots u'_{N-1} \delta'_0 \dots \delta'_{N-1} \zeta'_0 \dots \zeta'_{N-1}]'$ and subject to the further restriction that some of the components of z must be binary. The hybrid MPC problem maps into a Mixed-Integer Quadratic Programming (MIQP) problem when the quadratic costs (6) are used in (3a) [17], or a Mixed-Integer Linear Programming (MILP) problem when ∞ - or 1-norms are used as in (9) [11]. For both problems very efficient commercial and public domain solvers are available (see, e.g., <http://plato.asu.edu/bench.html>).

1.5 Extensions of the MPC Formulation

Tracking

The basic MPC regulation setup (3) can be extended to solve tracking problems. Given an output vector $y(t) = Cx(t) \in \mathbb{R}^p$ and a reference signal $r(t) \in \mathbb{R}^p$ to track under constraints (3d)–(3e), the cost function (3a) is replaced by

$$\sum_{k=0}^{N-1} (y_k - r(t)) Q_y (y_k - r(t)) + \Delta u'_k R \Delta u_k \quad (18)$$

where $Q_y = Q'_y \geq 0 \in \mathbb{R}^{p \times p}$ is a matrix of output weights, and the increments of command variables $\Delta u_k \triangleq u_k - u_{k-1}$, $u_{-1} \triangleq u(t-1)$, are the new optimization variables, possibly further constrained by $\Delta u_{\min} \leq \Delta u_k \leq \Delta u_{\max}$. In the above tracking setup vector $[x'(t) \ r'(t) \ u'(t-1)]'$ replaces $x(t)$ in (4) and the control law becomes $u(t) = u(t-1) + \Delta u_0^*(x(t), r(t), u(t-1))$.

Rejection of Measured Disturbances

In order to take into account measured disturbances $v(t) \in \mathbb{R}^{n_v}$ entering the system, one can simply change (3b) into

$$f(x_k, u_k) = Ax_k + Bu_k + Ev(t) \quad (19)$$

where $v(t)$ is the most recent available measurement of the disturbance entering the process, and is supposed constant over the prediction horizon. In this case the extended vector $[x'(t) \ v'(t)]$ enters problem (4).

Soft Constraints and Time-Varying Constraints

Soft constraints $g(z, x(t)) \leq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \varepsilon$, $\varepsilon \geq 0$, are also easily taken into account by adding a large penalty on ε or ε^2 in $h(z, x(t))$ in (4) and by including the new optimization variable ε in vector z .

Time-varying constraints $g(z, x(t)) \leq \gamma(t)$, $\gamma(t) \in \mathbb{R}^q$, $t = 0, 1, \dots$, can be also immediately taken into account, in this case the optimization problem (4) depends on both $x(t)$ and $\gamma(t)$.

2 Explicit Model Predictive Control

The MPC strategies described in Section 1 require running on-line optimization algorithms to solve the optimization problem (4) at each time-step t , based on the value of the current state vector $x(t)$ (or, more generally, based on $x(t)$, $r(t)$, $u(t-1)$, $v(t)$, $\gamma(t)$). For this reason, MPC has been traditionally labeled as a technology for slow processes. Advances in microcontroller and computer technology are progressively changing the concept of “slow”, but still solving (4) on line prevents the application of MPC in many contexts, even for the simplest case of QP or LP. Computation speed is not the only limitation: the code implementing the solver might generate concerns due to software certification issues, a problem which is particularly acute in safety critical applications.

The idea of explicit MPC is to solve the optimization problem (4) off-line for all $x(t)$ within a given set X , that we assume here polytopic

$$X = \{x \in \mathbb{R}^n : S_1 x \leq S_2\} \subset \mathbb{R}^n \quad (20)$$

and to make the dependence of $u(t)$ on $x(t)$ *explicit*, rather than *implicitly* defined by the optimization procedure that solves problem (4). It turns out that, as intuited in [74], such a dependence is piecewise affine in most of the formulations seen in Section 1, so that the MPC controller defined by (3), (5) can be represented in a totally equivalent way as

$$u(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq k_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if } H_M x \leq k_M. \end{cases} \quad (21)$$

Consequently, on-line computations are reduced to the simple evaluation of (21), which broadens the scope of applicability of MPC to fast-sampling applications. Explicit solutions of the form (21) to MPC problems can be obtained by solving *multiparametric programs*.

2.1 Multiparametric Programming: General Formulation

Consider the following mathematical program

$$\mathcal{NP}(x) : \min_z f(z, x) \quad (22a)$$

$$\text{s.t.} \quad g(z, x) \leq 0 \quad (22b)$$

$$Az + Bx + d = 0 \quad (22c)$$

where $z \in \mathbb{R}^\ell$ collects the decision variables, $x \in \mathbb{R}^n$ is a vector of parameters, $f : \mathbb{R}^\ell \times \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, $g : \mathbb{R}^\ell \times \mathbb{R}^n \rightarrow \mathbb{R}^q$, A is a $q_e \times \ell$ real matrix, B is a $q_e \times n$ real matrix, and $d \in \mathbb{R}^{q_e}$. Problem (22) is referred as a *multiparametric programming problem*. We are interested in characterizing the solution of problem (22) for a given polytopic set X of parameters. The solution of a multiparametric problem is a triple (V^*, Z^*, X_f) , where (i) the *set of feasible parameters* X_f is the set of all $x \in X$ for which problem (22) admits a solution, $X_f = \{x \in X : \exists z \in \mathbb{R}^\ell, g(z, x) \leq 0, Az + Bx + d = 0\}$; (ii) The *value function* $V^* : X_f \rightarrow \mathbb{R}$ associates with every $x \in X_f$ the corresponding optimal value of (22); (iii) The *optimal set* $Z^* : X_f \rightarrow 2^{\mathbb{R}^\ell}$ associates to each parameter $x \in X_f$ the corresponding set of optimizers $Z^*(x) = \{z \in \mathbb{R}^\ell : f(z, x) = V^*(x)\}$ of problem (22); (iv) An *optimizer function* $z^* : X_f \rightarrow \mathbb{R}^\ell$ associates to each parameter $x \in X_f$ an optimizer $z \in Z^*(x)$ ($Z^*(x)$ is just a singleton if $\mathcal{NP}(x)$ is strictly convex).

Let z be a feasible point of (22) for a given parameter x . The *active constraints* are the constraints that fulfill (22b) at equality, while the remaining constraints are called *inactive constraints*. The *active set* $\mathcal{A}(z, x)$ is the set of indices of the active constraints, that is,

$$\mathcal{A}(z, x) \triangleq \{i \in \{1, \dots, q\} | g_i(z, x) = 0\}.$$

The *optimal active set* $\mathcal{A}^*(x)$ is the set of indices of the constraints that are active for all $z \in Z^*(x)$, for a given $x \in X$,

$$\mathcal{A}^*(x) \triangleq \{i | i \in \mathcal{A}(z, x), \forall z \in Z^*(x)\}.$$

Given an index set $\mathcal{A} \subseteq \{1, \dots, q\}$, the *critical region* $CR_{\mathcal{A}}$ is the set of parameters for which the optimal active set is equal to \mathcal{A} , that is

$$CR_{\mathcal{A}} \triangleq \{x \in X | \mathcal{A}^*(x) = \mathcal{A}\}. \quad (23)$$

The following basic result for convex multiparametric programming was proved in [50] in the absence of equality constraints:

Lemma 1. *Consider the multiparametric problem (22) and let f , and the components g_i of g be jointly convex functions of (z, x) , for all $i = 1, \dots, q$. Then, X_f is a convex set and V^* is a convex function of x .*

The result can be easily generalized to the presence of linear equality constraints. For generic nonlinear functions f , g_i , the critical regions $CR_{\mathcal{A}}$ and an optimizer

function z^* are not easily characterizable, and suboptimal methods have been proposed [38, 15]. Procedures for solving at optimality the multiparametric programs arising from the MPC problems introduced in Section 1 are described in the following sections.

2.2 Multiparametric Quadratic Programming

By treating $x(t)$ as the vector of parameters, the QP problem arising from the linear MPC formulation of Section 1.1 can be treated as the multiparametric QP (mpQP)

$$\mathcal{QP}(x) : V^*(x) = \frac{1}{2}x'Yx + \min_z \frac{1}{2}z'Hx + x'F'z \quad (24a)$$

$$\text{s.t.} \quad Gz \leq W + Sx. \quad (24b)$$

In [18], the authors investigated the analytical properties of the mpQP solution, that are summarized by the following theorem.

Theorem 1. *Consider a multiparametric quadratic program with $H > 0$, $\begin{bmatrix} H & F \\ F' & Y \end{bmatrix} \geq 0$. The set X_f of parameters x for which the problem is feasible is a polyhedral set, the value function $V^* : X_f \mapsto \mathbb{R}$ is continuous, convex, and piecewise quadratic, and the optimizer $z^* : X_f \mapsto \mathbb{R}^\ell$ is piecewise affine and continuous.*

The immediate corollary of the above theorem is that the linear MPC approach based on linear costs described in Section 1.2 admits a continuous piecewise-affine explicit solution of the form (21).

The arguments used to prove Theorem 1 rely on the first-order Karush-Kuhn-Tucker (KKT) optimality conditions for the mpQP problem (24)

$$Hz + Fx + G'\lambda = 0 \quad \lambda \in \mathbb{R}^q \quad (25a)$$

$$\lambda_i(G^i z - W^i - S^i x) = 0, i = 1, \dots, q, \quad (25b)$$

$$\lambda \geq 0, \quad (25c)$$

$$Gz \leq W + Sx, \quad (25d)$$

where the superscript i denotes the i th row. Let us solve (25a) for z ,

$$z = -H^{-1}(G'\lambda + Fx) \quad (26)$$

and substitute the result in (25b). Taken any point $x_0 \in X_f$ (easily determined by a linear feasibility test on $[G \ S] \begin{bmatrix} z \\ x \end{bmatrix} \leq W$, for instance), solve $\mathcal{QP}(x_0)$ and determine the optimal active set $\mathcal{A}_0 = \mathcal{A}^*(x_0)$. Let $\hat{\lambda}$ and $\tilde{\lambda}$ denote the Lagrange multipliers corresponding to inactive and active constraints, respectively, and assume that the rows of \tilde{G} are linearly independent. For inactive constraints, set $\hat{\lambda}^*(x) = 0$. For active constraints, $-\tilde{G}H^{-1}\tilde{G}'\tilde{\lambda} - \tilde{W} - \tilde{S}x = 0$, and therefore set

$$\tilde{\lambda}^*(x) = -(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x), \quad (27)$$

where \tilde{G} , \tilde{W} , \tilde{S} correspond to the set of active constraints, and $(\tilde{G}H^{-1}\tilde{G}')^{-1}$ exists because the rows of \tilde{G} are linearly independent. Thus, $\lambda^*(x)$ is an affine function of x . By simply substituting $\lambda^*(x)$ into (26) it is easy to obtain

$$z^*(x) = H^{-1}\tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x) \quad (28)$$

and note that z^* is also an affine function of x . Vector z^* in (28) must satisfy the constraints in (25d), and by (25c), the Lagrange multipliers in (27) must remain non-negative. The set of inequalities defining the critical region $CR_{\mathcal{A}_0}$ in the x -space is hence given by the polyhedron

$$CR_{\mathcal{A}_0} = \{x \in \mathbb{R}^n : \hat{G}z^*(x) \leq \hat{W} + \hat{S}x, \tilde{\lambda}^*(x) \geq 0\}. \quad (29)$$

The critical region $CR_{\mathcal{A}_0}$ is the largest set of parameters for which the fixed combination of constraints \mathcal{A}_0 is the optimal active set and for which $z^*(x)$ is the optimizer function, because it satisfies all the KKT conditions (25) together with the dual solution $\lambda^*(x)$.

Different algorithms proposed to complete the characterization of the optimal solution $z^*(x)$ on the remaining set $CR^{rest} = X \setminus CR_{\mathcal{A}_0}$ are reviewed below.

In [18] the authors provided an algorithm for exploring CR^{rest} in order to generate all possible critical regions. The method proceeds recursively by (i) partitioning the set $X \subseteq \mathbb{R}^n$ into a finite number of polyhedra $\{CR_{\mathcal{A}_0}, R_1, \dots, R_N\}$, where $R_i = \{x \in X : A^i x > B^i, A^j x \leq B^j, \forall j < i\}$ and A, B define a minimal representation $Ax \leq B$ of $CR_{\mathcal{A}_0}$, (ii) computing a new critical region CR_i in each region R_i through the KKT conditions (25), partitioning $R_i \setminus CR_i$, and so on. The procedure terminates when no more new optimal combinations of active constraints are found. Note that the number of critical regions for an mpQP problem is upper-bounded by the number 2^q of possible combinations of active constraints. Such an upper-bound is usually very conservative, as most of the combinations are never optimal for any $x \in X$.

A faster method is suggested in [68]. The active set of a neighboring region is found by using the active set of the current region and the knowledge of the *type* of the hyperplane that is crossed. That is, if an hyperplane of type $\hat{G}^i z^*(x) \leq \hat{W}^i + \hat{S}^i x$ is crossed, then the corresponding constraint is added to the active set, while if an hyperplane of type $\tilde{\lambda}_i^*(x) \geq 0$ is crossed, the corresponding constraint is withdrawn from the active set.

The approach described in [5] is based on the idea that neighboring regions of a given region CR can be determined by stepping out from each facet of CR by a very small distance², in order to determine new parameter vectors x around which to blow a new critical region.

The approaches of [68, 5] implicitly rely on the assumption that the intersection of the closures of two adjacent critical regions is a facet of both closures, a condition which is referred to as the *facet-to-facet* property. Conditions for the property to hold are provided in [68]. The property is usually satisfied, however an example of

² This is usually a fixed tolerance. To be exact, a simple mono-parametric QP should be partially solved over a line stemming out of the facet, to avoid missing narrow critical regions.

non-degenerate mpQP where the facet-to-facet property does not hold is shown in [65], where the authors suggest to combine the algorithms of [68, 18] to overcome the issue.

Once the optimizer function z^* and X_f are found, the explicit MPC controller is immediately obtained by saving the component u_0^* of z^* . The regions where $u_0^*(x)$ is the same can be combined, provided that their union is polyhedral [16].

An alternative approach was taken in [64], where the authors derive a closed-form expression for the global, analytical solution to the MPC problem for linear, time-invariant, discrete-time models with a quadratic performance index and magnitude constraints on the input. The approach exploits the geometric properties of constrained MPC to obtain global analytic solutions which can be precomputed off-line. A further approach to determine explicit MPC solutions was suggested independently in [42], based on the idea of approximating the constrained LQR problem.

A *reverse transformation* procedure was introduced in [52] and later refined in [51] to solve the finite-horizon problem (3) with linear prediction model and quadratic costs using dynamic programming, by treating each stage of the dynamic programming recursion as a parametric piecewise quadratic programming problem.

In [53, 55] also suggest reverse transformation to solve mpQP problems arising from MPC formulations. In particular in [55] the authors presented an algorithm based on dynamic programming, which explores all the possible solution conditions in such a way that the combinatorial explosion of enumeration of active sets is avoided.

From a practical viewpoint, the approaches mostly used for solving explicit MPC based on mpQP are those of [68, 5].

Degeneracy in Multiparametric QP

Definition 1. *Given an active set \mathcal{A} , the linear independence constraint qualification (LICQ) property is said to hold if the set of active constraint gradients are linearly independent, i.e., the associated matrix \tilde{G} has full row rank.*

When LICQ is violated, we refer to as *primal degeneracy*. In this case the solution $\tilde{\lambda}^*(x)$ may not be uniquely defined (instead, $z^*(x)$ is always unique if $H > 0$). We refer to *dual degeneracy* if the dual problem of (24) is primal degenerate. In this case the solution $z^*(x)$ may not be unique, so clearly dual degeneracy can only happen if $H \geq 0$, $\det H = 0$. The absence of primal degeneracy ensures the following property of the value function V^* [6]:

Theorem 2. *Assume that the multiparametric QP (24) is not primal degenerate. Then the value function V^* is of class \mathcal{C}^1 .*

In [18] the authors suggest two ways to handle primal degeneracy. The first one is to compute a QR decomposition of \tilde{G} , recognize whether the associated critical region is full-dimensional, and in this case project away the Lagrange multipliers to get the critical region. The second way consists simply of collecting ℓ linearly independent constraints arbitrarily chosen, and proceed with the new reduced set, therefore avoiding the computation of projections. Due to the recursive nature of

the algorithm, the remaining other possible subsets of combinations of constraints leading to full-dimensional critical regions will be automatically explored later.

A different approach is suggested in [70] for extending the algorithm of [68] to handle both primal and dual degeneracy. In [43] the authors propose an algorithm for solving multiparametric linear complementarity (mpLC) problems defined by positive semidefinite matrices. The approach is rather elegant, as it covers in a unified way both (degenerate) mpQP and the multiparametric LP case, which is described in the next Section 2.3.

2.3 Multiparametric Linear Programming

By treating $x(t)$ as the vector of parameters, the linear MPC formulation of Section 1.2 can be treated as the multiparametric LP (mpLP)

$$\mathcal{LP}(x) : \min_z c'z \quad (30a)$$

$$\text{s.t.} \quad Gz \leq W + Sx, \quad (30b)$$

where $z \in \mathbb{R}^\ell$ is the optimization vector, $x \in X \subset \mathbb{R}^n$ is the vector of parameters, c, G, W, S are suitable constant matrices and X is the set of parameters of interest defined in (20). The following result is proved in [29].

Theorem 3. *Consider the mpLP problem (30). Then, the set X_f is a convex polyhedral set, there exists an optimizer function $z^* : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$ which is a continuous and piecewise affine function of x , and the value function $V^* : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous, convex, and piecewise affine function of x .*

The first methods for solving parametric linear programs appeared in 1952 in the master thesis published in [58], and independently in [32]. Since then, extensive research has been devoted to sensitivity and (multi)parametric analysis, see the references in [29] and also [30] for advances in the field. Gal and Nedoma provided the first method for solving multiparametric linear programs in 1972 [31]. Subsequently, until the beginning of the new millennium only a few authors [29, 28, 62] have dealt with multiparametric linear programming solvers. Subsequently, a renewed interest has arisen, mainly pushed by the application of mpLP in explicit MPC. For a recent survey on multiparametric linear programming the reader is referred to [44].

The KKT conditions related to the mpLP (30) are

$$Gz \leq W + Sx, \quad (31a)$$

$$G' \lambda = c, \quad (31b)$$

$$\lambda \geq 0, \quad (31c)$$

$$(G_i z - W_i - S_i x) \lambda_i = 0, \forall i. \quad (31d)$$

For a given fixed parameter $x_0 \in X$, by solving $\mathcal{LP}(x_0)$ one finds a solution z_0^*, λ_0^* and defines an optimal set of active constraints $\mathcal{A}(x_0)$. By forming submatrices $\tilde{G}, \tilde{W}, \tilde{S}$ of active constraints and $\hat{G}, \hat{W}, \hat{S}$ of inactive constraints, as in the mpQP case, and by assuming \tilde{G} square and full rank, one gets that

$$z^*(x) = (\tilde{G}^{-1}\tilde{S})x + (\tilde{G}^{-1}\tilde{W}), \quad (32)$$

is an affine function of x , $\lambda^*(x) \equiv \lambda^*(x_0)$ is constant, and the value function

$$V^*(x) = (W + Sx)' \lambda^*(x_0), \quad (33)$$

is also affine. Eqs. (32)–(33) provide the solution to (30) for all $x \in X$ such that

$$\hat{G}((\tilde{G}^{-1}\tilde{S})x + (\tilde{G}^{-1}\tilde{W})) < \hat{W} + \hat{S}x, \quad (34)$$

which defines the polyhedral *critical region* $CR_{\mathcal{A}_0}$ of all x for which the chosen combination of active constraints $\mathcal{A}(x_0)$ is the optimal one.

In [31] a critical region is defined as the set of all parameters x for which a certain basis is optimal for problem (30). The algorithm proposed in [31] for solving mpLPs generates non-overlapping critical regions by generating and exploring the graph of bases. In the graph of bases, the nodes represent optimal bases of the given multiparametric problem and two nodes are connected by an edge if it is possible to pass from one basis to another by one pivot step (in this case, the bases are called *neighbors*).

In [22] the authors proposed a geometric algorithm based on the direct exploration of the parameter space as in [18]. The definition of critical regions in [22] is not associated with the bases but with the set of active constraints and is related directly to the definition given in [28, 2].

The approach of [5] is also easily applicable to solve multiparametric linear programs. The *facet-to-facet* property however is not always guaranteed to hold also in mpLP problems. Conditions for ensuring that the property holds are reported in [46] and depend on degeneracy situations.

Degeneracy in Multiparametric LP

An LP problem (30) is said to be *primal degenerate* for some $x \in X$ if there exists a $z^*(x) \in Z^*(x)$ such that the number of active constraints at the optimizer larger than the number ℓ of optimization variables. In this case more than one basis describes the optimal primal solution. *Dual degeneracy* occurs when the dual problem of (30) is primal degenerate. In this case more than one primal solution $z^*(x)$ is optimal.

Dual degeneracy is undesired in linear MPC formulations based on linear costs as, depending on the on-line LP solver used, the solution may chatter from one optimizer to another, with consequent stress of the actuators without a real performance benefit.

Ways to handle degeneracy in mpLP are reported in [22] and in [46].

2.4 Explicit Solution of Min-Max MPC

In [13] the authors proposed an approach based on a combination of multiparametric linear programming and dynamic programming to obtain solutions to constrained robust optimal control problems in state feedback form. In particular, the solution

of *closed-loop* constrained robust optimal control problems is a piecewise affine function of the state, obtained by solving N mpLPs, while the solution z^* to *open-loop* constrained robust optimal control problems with parametric uncertainties in the B matrix only can be found by solving an mpLP. A different approach was taken in [47] to solve the CL-CROC formulation of [63] through a single (but larger) mpLP.

For explicit MPC based on linear parameter-varying (LPV) models one could in principle embed the problem into the robust min-max formulation of [13, 47]. However, this would unavoidably lead to a loss of performance. The ability to measure the parameter information is exploited in [19] to derive an explicit LPV-MPC approach also based on dynamic programming and mpLP.

For min-max problems based on quadratic costs, the solution was proved to be piecewise affine in [61], where the authors also showed that the critical regions were defined not only by a set of active constraints, but also by a set of active vertices. In [56] the problem was reformulated as a single mpQP.

2.5 Explicit Solution of Hybrid MPC

As detailed in [11], the MPC formulation based on ∞ - or 1-norms subject to the MLD dynamics (17) can be solved explicitly by treating the optimization problem associated with MPC as a multiparametric mixed integer linear programming (mp-MILP) problem.

An algorithm for solving mpMILPs was proposed in [1], based on a branch and bound procedure in which an mpLP problem is solved at each node of the tree. In [27] an alternative algorithm was proposed, which only solves mpLPs where the integer variables are fixed to the optimal value determined by an MILP, instead of solving mpLP problems with relaxed integer variables. More in detail, the mpMILP problem is alternatively decomposed into an mpLP and an MILP subproblem. It is easy to show that the explicit hybrid MPC controller has the form (21), but the control law $u(x)$ may be discontinuous across neighboring polyhedra [20].

It turns out that mpMILP is not very efficient in practice for solving hybrid MPC problems based on linear costs explicitly. In addition, good algorithms for multiparametric mixed integer quadratic programming (mpMIQP) problems for dealing with quadratic costs have not yet been proposed in the literature. Better ways to handle hybrid MPC problems explicitly in the case of linear and quadratic costs use piecewise affine (PWA) hybrid models. PWA models are defined by partitioning the space of states and inputs into polyhedral regions and associating with each region a different linear state-update equation

$$f(x_k, u_k) = A_i x_k + B_i u_k + f_i \text{ if } \begin{bmatrix} x_k \\ u_k \end{bmatrix} \in \mathcal{X}_i, \quad (35)$$

where $\{\mathcal{X}_i\}_{i=1}^s$ is a polyhedral partition of the state+input set. Hybrid systems of the form (35) can be obtained for instance by converting HYSDEL/MLD models using the method of [8] or [34].

In [20] the authors propose an algorithm for computing the solution to problem (3) with linear or quadratic costs and PWA models. The procedure is based on dynamic programming (DP) iterations. Multiparametric QPs are solved at each iteration, and quadratic value functions are compared to possibly eliminate regions that are proved to never be optimal. In typical situations the total number of solved mpQPs (as well as of generated polyhedral regions) grows exponentially with the prediction horizon N , and may suffer the drawback of an excessive partitioning of the state space. Nonetheless, the approach is preferable to gridding approaches usually employed to solve DP problems, especially in the case of nontrivial state-space dimensions.

A different approach was proposed in [52, 53], where the authors propose a reverse transformation procedure to enumerate all possible switching sequences, and for each sequence convert the PWA dynamics into a time-varying system and solve an optimal control problem explicitly via mpQP. In [3] the authors propose a related approach that exploits dynamic programming ideas (more precisely, backwards reachability analysis) to obtain all the feasible mode sequences (therefore avoiding an explicit enumeration of all of them), and that, after solving an mpQP for each sequence, post-processes the resulting polyhedral partitions to eliminate all the regions (and their associated control gains) that never provide the lowest cost, using a DC (Difference of Convex functions) algorithm. A similar algorithm is also used for explicit hybrid MPC based linear costs in the Hybrid Toolbox for Matlab [9].

The partition associated with the fully explicit optimal solution to problem (3) in the hybrid system case may not be polyhedral [20]. In this case, most explicit hybrid MPC algorithms for quadratic costs keep possible overlapping critical regions (generated by different mpQPs) and also store either the value function V^* or the full optimizer z^* , so that the optimal MPC move can be determined on-line by comparing the optimal values of all regions (in most cases just one, or a only a few) where $x(t)$ belongs.

Finally, we mention that for the special class of hybrid models given by linear systems with quantized inputs $u(t)$ problem (4) maps into a multiparametric nonlinear integer programming problem, for which a solution algorithm was provided in [7].

3 Reducing Complexity of Explicit MPC

The complexity of the piecewise affine explicit MPC law can be easily assessed after the MPC optimization problem is pre-solved off-line via the multiparametric techniques described in the previous sections. For a fixed number of parameters (states and reference signals), the complexity of the solution is given by the number M of regions that form the explicit solution (21). The number M mainly depends (exponentially, in the worst case) on the number q of constraints (and also of binary variables/system modes in the hybrid case) included in the MPC problem formulation (3), and only mildly on the number n of states. It also depends on the number ℓ of optimization variables, but mainly because q depends on ℓ (consider for instance

input constraints). In [10, Table I], a table shows such dependencies on random linear MPC problems. In the multiparametric QP case, an upper-bound to M is 2^q , which is the number of all possible combination of active constraints at optimality.

At the price of a possible loss of closed-loop performance, one way of reducing complexity is to shorten the prediction horizons (and/or blocking input moves [67]) in order to decrease q and ℓ . In practice, explicit MPC is limited to relatively small problems (typically one/two inputs, up to five-ten states, up to three/four free control moves) but allows one to reach very high sampling frequencies (up to 1 MHz) and requires a very simple control code to be embedded in the system.

To address such limitations, several attempts have been made in different directions. Starting with a given piecewise affine solution, in [33] the authors provide an approach to reduce the number of partitions by optimally merging regions where the affine gain is the same, so that the original solution is maintained but equivalently expressed with a minimal number of partitions. However, techniques for achieving a more drastic reduction of complexity require changing the solution, by accepting a certain level of suboptimality with respect to the original problem formulation (3).

Suboptimal mpQP solutions were proposed in [14] by relaxing the KKT conditions, such as nonnegativity of optimal dual variables. For explicit MPC controllers based on linear costs, a quite efficient “beneath/beyond” approximate mpLP algorithm has been proposed in [44]. In [40, 39] the authors propose recursive rectangular partitions of the parameter space to determine a suboptimal solution to general classes of multiparametric programming problems. Simplicial recursive partitions are instead proposed in [15] for general classes of nonlinear multiparametric programs that are jointly convex with respect to optimization variables and parameters, used in [57] for solving robust MPC formulations explicitly in approximate form. Rather than looking directly at the multiparametric programming problem, in [36] the authors propose to change the MPC formulation by recursively solving a sequence of simpler explicit MPC problems with horizon $N = 1$, where the terminal set \mathcal{X}_N is varied at each iteration and obtained from the previous iteration. A related approach is given in [35] for PWA systems, which leads to minimum-time controllers. Based on a dynamic programming formulation of the finite-horizon optimal control problem, in [49] the authors propose an approach to relax optimality within a prescribed bound in favor of the reduced complexity of the solution, which in the case of linear systems and piecewise affine convex costs leads to another form of computing approximate mpLP solutions. In the hybrid case, suboptimal solutions can be simply obtained by constraining the sequence of switching modes a priori, like for example has been proposed in [37], and then applying any of the solution methods proposed in Section 2.5.

More recently, authors have realized that a different way of obtaining suboptimal explicit MPC solutions is to avoid storing a full partition, by keeping only a subset of critical regions. For treating large MPC problems, in [59] the authors suggest the partial enumeration of the possible combinations of active constraints at optimality, so that the solution can be searched on line (explicitly or implicitly) within that small subset. An off-line training session of simulations is suggested to identify the subset of most important combinations of active constraints. In [26], the authors also

suggest to remove combinations of active constraints repeatedly, as long as closed-loop stability of the suboptimal solution is preserved. The drawback of the approach is that it need to start from the full exact solution, which may be very expensive to compute. A different novel approach which has analogies with both [59] and [26] is described below.

3.1 A Novel Practical Approach for Reducing Complexity of Explicit MPC

Under the assumption that only input constraints and soft state constraints can be enforced, given a maximum number L of regions accepted in the suboptimal solution, the proposed approach can be summarized as follows:

Off-line computations

1. Run extensive simulations of closed-loop MPC (using on-line QP evaluations) from several initial conditions and for several references and disturbances, as in [59]. Collect the L most recurrent combinations of active constraints I_1, I_2, \dots, I_L at optimality. Alternatively, if a full explicit solution is available, pick up the L regions with largest Chebychev radius.
2. Generate the corresponding critical regions CR_1, CR_2, \dots, CR_L , where $CR_i = \{x : H_i x \leq K_i\}$.

On-line computations

Let $x = x(t)$ be the current state (or, more generally, the current parameter vector including also $r(t), u(t-1), v(t), \gamma(t)$).

1. Compute $\beta_i(x) = \max_j \{H_i^j x - K_i^j\}$, that is the maximum violation of a facet inequality³, for all regions $j = 1, \dots, L$;
2. Let $h \in \{1, \dots, L\}$ such that $\beta_h(x) = \min_{i=1, \dots, L} \beta_i(x)$;
3. If $\beta_h(x) \leq 0$, then $x \in CR_h$ and $u(x) = F_h x + g_h$; stop;
4. Otherwise all $\beta_i(x) > 0$, and either compute the average

$$\bar{u}(x) = \left(\sum_{i=1, \dots, L} \frac{1}{\beta_i(x)} \right)^{-1} \sum_{i=1, \dots, L} \frac{1}{\beta_i(x)} (F_i x + g_i) \quad (36a)$$

or extrapolate the solution from the critical region CR_h corresponding to the least violation $\beta_h(x)$

$$\bar{u}(x) = F_h x + g_h \quad (36b)$$

³ Alternatively, one can compute the largest distance of x from a violated facet half-space

$$\beta_i(x) = \max_j \left\{ \frac{H_i^j x - K_i^j}{\sqrt{H_i^j (H_i^j)^T}} \right\}.$$

5. Create a suboptimal input move by saturating $u(x) = \text{sat} \bar{u}(x)$, where sat is a suitable saturation function (for instance, the standard saturation function in case \mathcal{U} is the box $\{u \in \mathbb{R}^m : u_{\min} \leq u \leq u_{\max}\}$ and $\mathcal{X} = \mathbb{R}^n$).

Note that the procedure is very general, as it can be applied to any of the MPC formulations described in Section 1, and in particular to linear and hybrid MPC with linear or quadratic costs⁴. While stability properties of the suboptimal solution can only be checked a posteriori, contrary to [59] the number of regions is bounded a priori, and off-line computations are much simpler, as numerical manipulations over the exact full partition are not required. Note that for a given reduced partition and a set X of parameters of interest as defined in (20), the largest minimal violation $\beta_{\max} = \max_{x \in X} \beta_h(x)$ can be computed by solving the mixed-integer linear programming problem

$$\begin{aligned}
 \beta_{\max} = \max_{x, \varepsilon, \{\delta_i^j\}} \quad & \varepsilon \\
 \text{s.t.} \quad & [\delta_i^j = 1] \leftrightarrow [H_i^j x \leq K_i^j + \mathbf{1}\varepsilon], \quad \forall j = 1, \dots, n_i, \quad \forall i = 1, \dots, L \\
 & \sum_{j=1}^{n_i} \delta_i^j \leq n_i - 1 \\
 & \delta_i^j \in \{0, 1\}^{n_i}, \quad \forall i = 1, \dots, L \\
 & S_1 x \leq S_2
 \end{aligned} \tag{37}$$

where $n_i = \dim(K_i)$ is the number of inequalities defining the i th critical region, superscript j denotes the j th row or component, and $\mathbf{1}$ is a vector of all ones. Note that, as the aim is to have L very small, the number $\sum_{i=1}^L n_i$ of binary variables in the MILP (37) is also small. The quantity β_{\max} can be used as a tuning parameter to decide the number L of regions to include in the suboptimal explicit MPC controller.

An Example

Consider the double integrator example described in [18, Example 7.3] with $N = N_u = 8$, but with the different sampling time $T_s = 0.1$. The exact explicit solution, computed in about 3 s on a Laptop PC Intel Core Duo 1.2 GHz using the Hybrid Toolbox for Matlab [9], consists of $M = 99$ regions and is depicted in Figure 1(a),

The solution with $L = 3$ regions reported in Figure 1(b), corresponds to $\varepsilon = 1.4661$ in (37) and leads to the closed loop trajectories of Figure 1(c), obtained by averaging as in (36a).

For the hybrid explicit MPC controller described in [23, Figure 6b], which consists of 671 regions, experiments have shown that, after simulation training, the suboptimal approach described above with $L = 60$ regions leads to almost indistinguishable closed-loop MPC trajectories.

⁴ In case of hybrid MPC based on quadratic costs, polyhedral critical regions may be overlapping. In this case, the explicit form of the cost function $V(x)$ on those regions must be also taken into account, in order to define a value for $\bar{u}(x)$ that penalizes both a large $\beta(x)$ and a large cost $V(x)$.

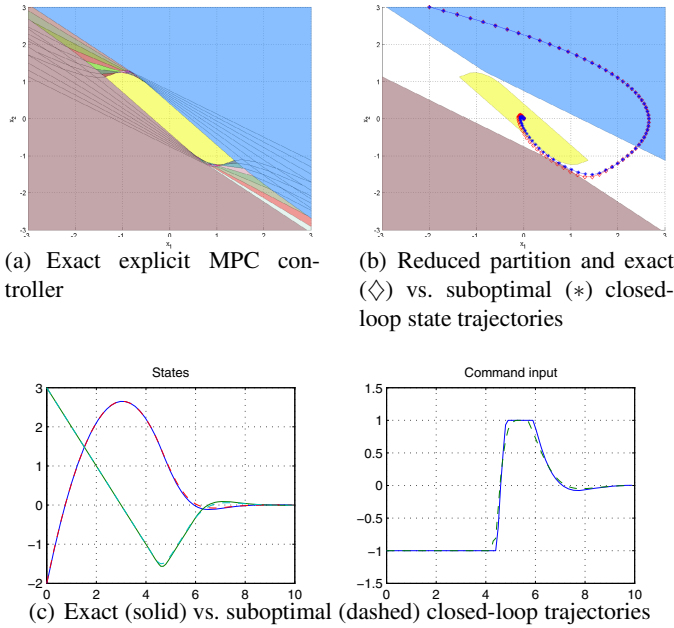


Fig. 1 Exact and suboptimal explicit MPC controller ($L = 3$)

4 Implementation of Explicit MPC

The explicit MPC solution (21) pre-computed off-line is a lookup table of linear feedback gains. The right gain is selected on-line by finding the region $\{x : H_i x \leq k_i\}$ of the polyhedral partition where the current state $x(t)$ (or, more generally, the current vector of parameters) lies. This problem has been referred to as *point-location problem*. The most simple solution is to store all the M polyhedra of the partition and on-line search through them until the right one is found. While this procedure is extremely easy to implement in a computer code, more efficient ways have been proposed for evaluating explicit MPC controllers, which become appealing when the number M of regions is large.

By exploiting the properties of multiparametric linear and quadratic solutions, in [6] two new algorithms are proposed that avoid storing the polyhedral regions, significantly reducing the on-line storage demands and computational complexity of the evaluation of control. In [69] the authors suggest to organize the hyperplanes defining the regions on a binary search tree (possibly further subdividing some of the regions), so that the time to locate the state vector on-line within the partition becomes logarithmic in the number of stored cells and memory space is saved. For the multiparametric linear case, another method is given in [45], where additively weighted nearest neighbour search ideas are used to solve the point location problem logarithmically in the number N of regions. Assuming that a bound between the evolution of the real process and the prediction model is known, in [66] the authors

suggest to limit the point location problem only to those regions that are reachable in one time step from the currently active one. In [25] the authors propose a search tree algorithm based on bounding boxes and interval trees. A subdivision walking method is proposed in [72].

From the hardware synthesis viewpoint, [41] showed that explicit MPC solutions can be implemented in an application specific integrated circuit (ASIC) with about 20,000 gates, leading to computation times in order of 1 μ s.

Whether the explicit form (21) is preferable to the one based on on-line optimization depends on available CPU time, data memory, and program memory (see e.g. [10, Table II] for a comparison in the linear quadratic case).

5 Tools for Explicit MPC

The Hybrid Toolbox for Matlab [9] allows one to design explicit MPC control laws for linear and hybrid systems. Various functions for the design, visualization, simulation, and C-code generation of explicit MPC controllers are provided. Similar and other functionalities can be found in the Multi Parametric Toolbox for Matlab [48].

6 Explicit MPC in Applications

Industrial problems addressed through explicit MPC techniques have been reported in dozens of technical papers, starting from what is probably the first work in this domain [21]. The most suitable applications for explicit MPC are fast-sampling problems (in the order of 1-50 ms) and relatively small size (1-2 manipulated inputs, 5-10 parameters). Most of the applications of explicit MPC have been reported in the automotive domain and electrical power converters.

7 Conclusions

At the time of writing this survey, exact explicit characterizations of MPC for linear and hybrid systems have been quite well studied. Efficient algorithms exist to compute solutions to multiparametric linear and quadratic programming problems and explicit control laws, and Matlab toolboxes are available to let control engineers apply these ideas in practical applications. Nonetheless, the field is far from being mature. The clear limitation of exact explicit MPC solutions is that the number of regions composing the solution grows massively with the size of the problem (mainly with the number of constraints in the MPC problem).

Future research efforts should therefore pursue three main directions. First, new suboptimal methods that allow trading off between loss of closed-loop performance and number of partitions are needed; the whole concept of looking for optimal open-loop performance is actually weak, as the MPC cost function is usually chosen by trial and error. Second, PWA solutions are not necessarily the best way to define suboptimal MPC laws, other piecewise-nonlinear ways of defining the solution may

lead to more compact representations; in this respect, links to nonlinear approximation approaches could be sought to approximate samples of the MPC control profile computed through (4) within guaranteed error bounds. Third, semi-explicit methods should be also sought, in order to pre-process of line as much as possible of the MPC optimization problem without characterizing all possible optimization outcomes, but rather leaving some optimization operations on-line.

Acknowledgements. The authors thank Gabriele Pannocchia, Ilya Kolmanovsky, Davor Hrovat, and Stefano Di Cairano for discussing the ideas related to the novel practical approach for reducing complexity of explicit MPC described in this paper, and David Muñoz de la Peña, David Q. Mayne, and Ilya Kolmanovsky for suggesting improvements on the original manuscript.

References

1. Acevedo, J., Pistikopoulos, E.N.: A multiparametric programming approach for linear process engineering problems under uncertainty. *Ind. Eng. Chem. Res.* 36, 717–728 (1997)
2. Adler, I., Monteiro, R.D.C.: A geometric view of parametric linear programming. *Algoritmica* 8(2), 161–176 (1992)
3. Alessio, A., Bemporad, A.: Feasible mode enumeration and cost comparison for explicit quadratic model predictive control of hybrid systems. In: 2nd IFAC Conference on Analysis and Design of Hybrid Systems, Alghero, Italy, pp. 302–308 (2006)
4. Allwright, J.C., Papavasiliou, G.C.: On linear programming and robust model-predictive control using impulse-responses. *Systems & Control Letters* 18, 159–164 (1992)
5. Baotić, M.: An efficient algorithm for multi-parametric quadratic programming. Technical Report AUT02-05, Automatic Control Institute, ETH, Zurich, Switzerland (2002)
6. Baotić, M., Borrelli, F., Bemporad, A., Morari, M.: Efficient on-line computation of constrained optimal control. *SIAM Journal on Control and Optimization* 47(5), 2470–2489 (2008)
7. Bemporad, A.: Multiparametric nonlinear integer programming and explicit quantized optimal control. In: Proc. 42nd IEEE Conf. on Decision and Control, Maui, Hawaii, USA, pp. 3167–3172 (2003)
8. Bemporad, A.: Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Trans. Automatic Control* 49(5), 832–838 (2004)
9. Bemporad, A.: Hybrid Toolbox – User’s Guide (January 2004), <http://www.dii.unisi.it/hybrid/toolbox>
10. Bemporad, A.: Model-based predictive control design: New trends and tools. In: Proc. 45th IEEE Conf. on Decision and Control, San Diego, CA, pp. 6678–6683 (2006)
11. Bemporad, A., Borrelli, F., Morari, M.: Piecewise linear optimal controllers for hybrid systems. In: American Control Conference, Chicago, IL, pp. 1190–1194 (June 2000)
12. Bemporad, A., Borrelli, F., Morari, M.: Model predictive control based on linear programming — The explicit solution. *IEEE Trans. Automatic Control* 47(12), 1974–1985 (2002)
13. Bemporad, A., Borrelli, F., Morari, M.: Min-max control of constrained uncertain discrete-time linear systems. *IEEE Trans. Automatic Control* 48(9), 1600–1606 (2003)

14. Bemporad, A., Filippi, C.: Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. *Journal of Optimization Theory and Applications* 117(1), 9–38 (2003)
15. Bemporad, A., Filippi, C.: An algorithm for approximate multiparametric convex programming. *Computational Optimization and Applications* 35(1), 87–108 (2006)
16. Bemporad, A., Fukuda, K., Torrisi, F.D.: Convexity recognition of the union of polyhedra. *Computational Geometry: Theory and Applications* 18, 141–154 (2001)
17. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. *Automatica* 35(3), 407–427 (1999)
18. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. *Automatica* 38(1), 3–20 (2002)
19. Besselmann, T., Löfberg, J., Morari, M.: Explicit model predictive control for systems with linear parameter-varying state transition matrix. In: *Proc. 17th IFAC World Congress, Seoul, Korea* (2008)
20. Borrelli, F., Baotić, M., Bemporad, A., Morari, M.: Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica* 41(10), 1709–1721 (2005)
21. Borrelli, F., Bemporad, A., Fodor, M., Hrovat, D.: A hybrid approach to traction control. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) *HSCC 2001. LNCS*, vol. 2034, pp. 162–174. Springer, Heidelberg (2001)
22. Borrelli, F., Bemporad, A., Morari, M.: A geometric algorithm for multi-parametric linear programming. *Journal of Optimization Theory and Applications* 118(3), 515–540 (2003)
23. Di Cairano, S., Bemporad, A., Kolmanovsky, I., Hrovat, D.: Model predictive control of magnetically actuated mass spring dampers for automotive applications. *International Journal of Control* 80(11), 1701–1716 (2007)
24. Campo, P.J., Morari, M.: Robust model predictive control. In: *American Control Conference*, vol. 2, pp. 1021–1026 (1987)
25. Christophersen, F.J., Kvasnica, M., Jones, C.N., Morari, M.: Efficient evaluation of piecewise control laws defined over a large number of polyhedra. In: *Proc. European Control Conf., Kos, Greece*, pp. 2360–2367 (2007)
26. Christophersen, F.J., Zeilinger, M.N., Jones, C.N., Morari, M.: Controller complexity reduction for piecewise affine systems through safe region elimination. In: *IEEE Conference on Decision and Control, New Orleans, LA*, pp. 4773–4778 (2007)
27. Dua, V., Pistikopoulos, E.N.: An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research* 1, 123–139 (2000)
28. Filippi, C.: On the geometry of optimal partition sets in multiparametric linear programming. Technical Report 12, Department of Pure and Applied Mathematics, University of Padova, Italy (June 1997)
29. Gal, T.: *Postoptimal Analyses, Parametric Programming, and Related Topics*, 2nd edn. de Gruyter, Berlin (1995)
30. Gal, T., Greenberg, H.J. (eds.): *Advances in Sensitivity Analysis and Parametric Programming*. International Series in Operations Research & Management Science, vol. 6. Kluwer Academic Publishers, Dordrecht (1997)
31. Gal, T., Nedoma, J.: Multiparametric linear programming. *Management Science* 18, 406–442 (1972)
32. Gass, S.I., Saaty, T.L.: The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly* 2, 39–45 (1955)

33. Geyer, T., Torrisi, F.D., Morari, M.: Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica* 44, 1728–1740 (2008)
34. Geyer, T., Torrisi, F.D., Morari, M.: Efficient Mode Enumeration of Compositional Hybrid Models. In: Maler, O., Pnueli, A. (eds.) *HSCC 2003*. LNCS, vol. 2623, pp. 216–232. Springer, Heidelberg (2003)
35. Grieder, P., Kvasnica, M., Baotić, M., Morari, M.: Low complexity control of piecewise affine systems with stability guarantee. In: *American Control Conference*, Boston, MA (2004)
36. Grieder, P., Morari, M.: Complexity reduction of receding horizon control. In: *Proc. 42th IEEE Conf. on Decision and Control*, Maui, Hawaii, USA, pp. 3179–3184 (2003)
37. Ingimundarson, A., Ocampo-Martinez, C., Bemporad, A.: Model predictive control of hybrid systems based on mode-switching constraints. In: *Proc. 46th IEEE Conf. on Decision and Control*, New Orleans, LA, pp. 5265–5269 (2007)
38. Johansen, T.A.: On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In: *Proc. 41th IEEE Conf. on Decision and Control*, Las Vegas, Nevada, USA, pp. 2768–2773 (December 2002)
39. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica* 40, 293–300 (2004)
40. Johansen, T.A., Grancharova, A.: Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Trans. Automatic Control* 58(5), 810–815 (2003)
41. Johansen, T.A., Jackson, W., Schriber, R., Tøndel, P.: Hardware synthesis of explicit model predictive controllers. *IEEE Trans. Contr. Systems Technology* 15(1), 191–197 (2007)
42. Johansen, T.A., Petersen, I., Slupphaug, O.: On explicit suboptimal LQR with state and input constraints. In: *Proc. 39th IEEE Conf. on Decision and Control*, Sydney, Australia, pp. 662–667 (December 2000)
43. Jones, C., Morari, M.: Multiparametric linear complementarity problems. In: *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, pp. 5687–5692 (2006)
44. Jones, C.N., Baric, M., Morari, M.: Multiparametric linear programming with applications to control. *European Journal of Control* 13, 152–170 (2007)
45. Jones, C.N., Grieder, P., Raković, S.V.: A logarithmic-time solution to the point location problem for parametric linear programming. *Automatica* 42(12), 2215–2218 (2006)
46. Jones, C.N., Kerrigan, E.C., Maciejowski, J.M.: Lexicographic perturbation for multiparametric linear programming with applications to control. *Automatica* 43(10), 1808–1816 (2007)
47. Kerrigan, E.C., Maciejowski, J.M.: Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution. *Int. J. Robust Nonlinear Control* 14(4), 395–413 (2004)
48. Kvasnica, M., Grieder, P., Baotić, M.: *Multi Parametric Toolbox, MPT* (2006), <http://control.ee.ethz.ch/~mpt/>
49. Lincoln, B., Rantzer, A.: Relaxing dynamic programming. *IEEE Trans. Automatic Control* 51(8), 1249–1260 (2006)
50. Mangasarian, O.L., Rosen, J.B.: Inequalities for stochastic nonlinear programming problems. *Operations Research* 12, 143–154 (1964)
51. Mayne, D.Q., Raković, S.V., Kerrigan, E.C.: Optimal control and piecewise parametric programming. In: *Proc. European Control Conf.*, Kos, Greece, pp. 2762–2767 (2007)

52. Mayne, D.Q.: Control of constrained dynamic systems. Technical Report EEE/C&P/DQM/9/2001, Imperial College, London, U.K. (September 2001)
53. Mayne, D.Q., Rakovic, S.: Optimal control of constrained piecewise affine discrete time systems using reverse transformation. In: Proc. 41th IEEE Conf. on Decision and Control, Las Vegas, Nevada, USA, pp. 1546–1551 (December 2002)
54. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. *Automatica* 36(6), 789–814 (2000)
55. Muñoz de la Peña, D., Alamo, T., Bemporad, A., Camacho, E.F.: A dynamic programming approach for determining the explicit solution of MPC controllers. In: Proc. 43th IEEE Conf. on Decision and Control, Paradise Island, Bahamas, pp. 2479–2484 (2004)
56. Muñoz de la Peña, D., Alamo, T., Ramírez, D.R., Camacho, E.F.: Min-max model predictive control as a quadratic program. *IET Control Theory & Applications* 1(1), 328–333 (2007)
57. Muñoz de la Peña, D., Bemporad, A., Filippi, C.: Robust explicit MPC based on approximate multi-parametric convex programming. *IEEE Trans. Automatic Control* 51(8), 1399–1403 (2006)
58. Orchard-Hays, W.: Notes on linear programming (part 6): the Rand code for the simplex method (sx4). Technical Report 1440, Rand Corporation (1955)
59. Pannocchia, G., Rawlings, J.B., Wright, S.J.: Fast, large-scale model predictive control by partial enumeration. *Automatica* 43, 852–860 (2007)
60. Propoi, A.I.: Use of linear programming methods for synthesizing sampled-data automatic systems. *Automation and Remote Control* 24(7), 837–844 (1963)
61. Ramírez, D.R., Camacho, E.F.: Piecewise affinity of min-max mpc with bounded additive uncertainties and a quadratic criterion. *Automatica* 42(2), 295–302 (2006)
62. Schechter, M.: Polyhedral functions and multiparametric linear programming. *Journal of Optimization Theory and Applications* 53(2), 269–280 (1987)
63. Scokaert, P.O.M., Mayne, D.Q.: Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. Automatic Control* 43(8), 1136–1142 (1998)
64. Seron, M.M., DeDoná, J.A., Goodwin, G.C.: Global analytical model predictive control with input constraints. In: Proc. 39th IEEE Conf. on Decision and Control, Sydney, Australia, pp. 154–159 (2000)
65. Spjøtvold, J., Kerrigan, E.C., Jones, C.N., Tøndel, P., Johansen, T.A.: On the facet-to-facet property of solutions to convex parametric quadratic programs. *Automatica* 42(12), 2209–2214 (2006)
66. Spjøtvold, J., Rakovic, S.V., Tøndel, P., Johansen, T.A.: Utilizing reachability analysis in point location problems. In: Proc. 45th IEEE Conf. on Decision and Control, San Diego, CA, pp. 4568–4569 (2006)
67. Tøndel, P., Johansen, T.A.: Complexity reduction in explicit linear model predictive control. In: Proc. 15th IFAC World Congress, Barcelona, Spain (2002)
68. Tøndel, P., Johansen, T.A., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39(3), 489–497 (2003)
69. Tøndel, P., Johansen, T.A., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. *Automatica* 39(5), 945–950 (2003)
70. Tøndel, P., Johansen, T.A., Bemporad, A.: Further results on multiparametric quadratic programming. In: Proc. 42th IEEE Conf. on Decision and Control, Maui, Hawaii, USA, pp. 3173–3178 (2003)

71. Torrisi, F.D., Bemporad, A.: HYSDEL — A tool for generating computational hybrid models. *IEEE Trans. Contr. Systems Technology* 12(2), 235–249 (2004)
72. Wang, Y., Jones, C.N., Maciejowski, J.: Efficient point location via subdivision walking with application to explicit MPC. In: *Proc. European Control Conf.*, Kos, Greece, pp. 447–453 (2007)
73. Witsenhausen, H.S.: A min-max control problem for sampled linear systems. *IEEE Trans. Automatic Control* 13(1), 5–21 (1968)
74. Zafiriou, E.: Robust model predictive control of processes with hard constraints. *Computers & Chemical Engineering* 14(4/5), 359–371 (1990)