A Multiparametric Quadratic Programming Algorithm with Polyhedral Computations Based on Nonnegative Least Squares

Alberto Bemporad, *Fellow, IEEE*

***Abstract*—Model Predictive Control (MPC) is one of the most successful techniques adopted in industry to control multivariable systems under constraints on input and output variables. To circumvent the main drawback of MPC, i.e., the need to solve a Quadratic Program (QP) on line to compute the control action, explicit MPC was proposed in the past to precompute the control law off line using multiparametric QP (mpQP). The resulting form of the MPC law is piecewise affine, which is extremely easy to code, can be computed online by simple arithmetic operations, and requires a maximum number of iterations that can be exactly determined a priori. On the other hand, the offline computations to solve the mpQP problem require detecting emptiness, full-dimensionality, and minimal hyperplane representations of polyhedra, and other computational geometric operations. While most of the existing methods solve such operations via linear programming, the approach proposed in this paper relies on a nonnegative least squares (NNLS) solver that is very simple to code, fast to execute, and provides solutions up to machine precision. In addition, the new approach exploits QP duality to identify and construct critical regions and to handle degeneracy issues.**

***Index Terms*—Multiparametric programming, Model predictive control, Quadratic programming, Nonnegative least squares.**

I. Introduction

MODEL Predictive Control (MPC) is a well known methodology for synthesizing feedback control laws that optimize closed-loop performance subject to prescribed operating constraints on inputs, states, and outputs [1]–[3]. In MPC, the control action is obtained by solving a finite horizon open-loop optimal control problem at each sampling instant. Each optimization yields a sequence of optimal control moves, but only the first sample is applied to the process. At the next time step, the computation is repeated over a shifted time-horizon, taking the most recently available state information as the initial condition of the new optimal control problem. In most practical applications, MPC is based on a linear discrete-time time-invariant model of the controlled system and quadratic penalties on tracking errors and actuation efforts; in this case, the optimal control problem can be recast as a quadratic programming (QP) problem, whose linear term of the cost function and right-hand side of the constraints depend on a vector of parameters that may change from one step to another (usually the current state and reference signals).

Several *on-line* solution algorithms have been studied for embedding quadratic optimization in control hardware, such as active-set methods [4], [5], interior-point methods [6], alternative direction of multipliers (ADMM) methods [7], and fast gradient projection methods [8]. Explicit MPC uses instead *multiparametric quadratic programming* (mpQP) to pre-solve the QP *off-line*, converting the MPC law into a continuous and piecewise affine (PWA) function of a vector of parameters [9].

A few algorithms have been proposed to solve mpQP problems, we mention here the approaches of [10]–[13], referring the reader to the survey paper [14] for more comprehensive list, including extensions to linear MPC based on convex PWA costs, such as 1- and ∞-norms [15], to min-max MPC problems for robustness with respect to additive and/or multiplicative unknown-but-bounded uncertainty [16], and to hybrid MPC [17], [18].

This paper introduces a novel and effective mpQP algorithm that, beyond collecting and extending effective ideas scattered in the literature, exploits the dual mpQP formulation and proposes an original set of technical results for solving the required polyhedral computations by using a nonnegative least squares (NNLS) algorithm [19, Chapter 23], in addition to a QP solver. The latter must be available anyway, as the design flow for explicit MPC is typically to first tune an *implicit* (i.e., based on on-line QP) MPC controller in simulation, then generate its explicit version for deployment. Recently, an approach to also solve quadratic programs based on NNLS was developed by the author in [20], therefore showing that the entire mpQP solver can be totally based on a simple NNLS algorithm.

One advantage of using NNLS is to speed up polyhedral computations, as the numerical experiments of this paper show, which are the bottleneck of all mpQP solvers proposed in the literature. Moreover, it is very simple to code, and, being an active-set method applied to solve the small-dimensional problems that arise in mpQP problems of practical interest, provides numerical solutions with accuracy up to machine precision after a limited number iterations, a feature that is not enjoyed by iterative solvers and that is instead very important in the operations of computational geometry required by the multiparametric solver. In addition, the mpQP approach proposed in the paper relies on identifying critical regions via the dual QP formulation, which eases dealing with degeneracy issues. The overall mpQP approach based on NNLS, as evidenced in numerical experiments, provides superior performance compared to existing mpQP solvers.

This paper is organized as follows. Section II defines

A. Bemporad is with the IMT Institute for Advanced Studies Lucca, Italy. Email: alberto.bemporad@imtlucca.it

the MPC framework and the associated multiparametric programming problem. The reformulation of several polyhedral computational problems via nonnegative least-squares is dealt with in Section III, which is the basis for the mpQP algorithm presented in Section IV. Section V analyzes the complexity of the solution and effective ways to evaluate it. Numerical results are reported in Section VI and some conclusions are drawn in Section VII.

A. Notation

Let $\mathbb{R}^n$, $\mathbb{N}$ denote the set of real vectors of dimension n and the set of natural integers, respectively. Let $\mathcal{I} \subset \mathbb{N}$ be a finite set of integers and denote by $\#\mathcal{I}$ its cardinality. For a vector $a \in \mathbb{R}^n$, a_i denotes the i-th entry of a, $a_{\mathcal{I}}$ the subvector obtained by collecting the entries a_i for all $i \in \mathcal{I}$, $\|a\|_2$ the Euclidean norm of a, the condition $a > 0$ is equivalent to $a_i > 0$, $\forall i = 1, \ldots, n$ (and similarly for $\geq$, $\leq$, $<$ inequalities), $\operatorname{diag}(a)$ is the diagonal matrix whose (i,i)-th element is a_i, and $B(a, \epsilon) = \{x \in \mathbb{R}^n : \|x - a\|_2 \leq \epsilon\}$ the Euclidean ball centered in a of radius ϵ. Given two vectors $a, b \in \mathbb{R}^n$, $0 \leq a \perp b \geq 0$ means the complementarity condition $a_i b_i = 0$, $a_i \geq 0$, $b_i \geq 0$, $\forall i = 1, \ldots, n$. For a matrix $A \in \mathbb{R}^{n \times m}$, A' denotes its transpose, A_i denotes the i-th row of A, $A_{\mathcal{I}}$ the submatrix of A obtained by collecting the rows A_i for all $i \in \mathcal{I}$, $A_{\mathcal{I}\mathcal{J}}$ the submatrix of A obtained by collecting the rows and columns of A indexed by $i \in \mathcal{I}$ and $j \in \mathcal{J}$, respectively, $A_{\|} \in \mathbb{R}^n$ the vector whose i-th entry is $\|A_i\|_2$, and $A^{\#} \in \mathbb{R}^{m \times n}$ a pseudoinverse matrix of A, namely $AA^{\#}A = A$, $A^{\#}AA^{\#} = A^{\#}$, $AA^{\#} = (AA^{\#})'$, $A^{\#}A = (A^{\#}A)'$ (if A is full column rank, $A^{\#} \triangleq (A'A)^{-1}A'$). For a square matrix $A \in \mathbb{R}^{n \times n}$, $\det A$ denotes the determinant of A, $A > 0$ denotes positive definiteness of A (and similarly $\geq$, $<$, $\leq$ denote positive semidefiniteness, negative definiteness, negative semidefiniteness, respectively). Matrix I_n denotes the identity matrix of order n, where sometimes the subscript n is dropped if the dimension is clear from the context.

II. Explicit Model Predictive Control

Consider the following MPC formulation

$$
\begin{aligned}
V^*(x) = \min_z \quad & \ell_N(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) && \text{(1a)}\\
\text{s.t.} \quad & x_{k+1} = \mathcal{A}x_k + \mathcal{B}u_k && \text{(1b)}\\
& C_x x_k + C_u u_k \leq c_0 && \text{(1c)}\\
& \quad k = 0, \ldots, N-1 && \\
& C_N x_N \leq c_N && \text{(1d)}\\
& x_0 = x, && \text{(1e)}
\end{aligned}
$$

where N is the prediction horizon, $u_k \in \mathbb{R}^{n_u}$ is the vector of manipulated variables at prediction time k to be optimized, $k = 0, \ldots, N-1$, $x \in \mathbb{R}^m$ is the current state vector of the controlled system whose state-space model matrices are $\mathcal{A} \in \mathbb{R}^{m \times m}$ and $\mathcal{B} \in \mathbb{R}^{m \times n_u}$,

$$
\begin{aligned}
\ell(x, u) &= \frac{1}{2}x'Q_x x + u'R_u u && \text{(2a)}\\
\ell_N(x) &= \frac{1}{2}x'P_x x && \text{(2b)}
\end{aligned}
$$

are the stage cost and terminal cost, respectively, and $Q_x = Q_x' \geq 0$, $P_x = P_x' \geq 0$, $R_u = R_u' > 0$.

Let $n_c \in \mathbb{N}$ be the number of constraints imposed at prediction time $k = 0, \ldots, N-1$, namely $C_x \in \mathbb{R}^{n_c \times m}$, $C_u \in \mathbb{R}^{n_c \times n_u}$, $c_0 \in \mathbb{R}^{n_c}$, let n_N be the number of terminal constraints, namely $C_N \in \mathbb{R}^{n_N \times m}$, $c_N \in \mathbb{R}^{n_N}$, and let $q \triangleq Nn_c + n_N$ be the total number of linear inequality constraints imposed in the MPC problem formulation (1).

By eliminating the states $x_k = A^k x + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$ from problem (1), the finite-time optimal control problem (1) can be expressed in condensed form as the strictly convex quadratic program (QP)

$$
\begin{aligned}
V^*(x) \triangleq \min_z \quad & \frac{1}{2}z'Qz + (Fx + c)'z + \frac{1}{2}x'Yx && \text{(3a)}\\
\text{s.t.} \quad & Gz \leq W + Sx, && \text{(3b)}
\end{aligned}
$$

where $z \triangleq [\, u_0' \, \ldots \, u_{N-1}' \,]' \in \mathbb{R}^n$, $n \triangleq n_u N$, is the optimization vector, $Q = Q' \in \mathbb{R}^{n \times n}$ is the Hessian matrix, $F \in \mathbb{R}^{n \times m}$, $c \in \mathbb{R}^n$, $Y = Y' \in \mathbb{R}^{m \times m}$ has no influence on the optimizer as it only affects the optimal value of (3a), and the matrices $G \in \mathbb{R}^{q \times n}$, $S \in \mathbb{R}^{q \times m}$, $W \in \mathbb{R}^q$ define the constraints imposed in (1) in a compact form. Because of the assumptions made on the weights Q_x, R_x, P_x, matrix $Q > 0$ and $\left[\begin{smallmatrix} Q & F' \\ F & Y \end{smallmatrix}\right] \geq 0$. Although the value of vector c arising from (1) is always zero, we include it for the sake of generality of the proposed mpQP approach.

The MPC control law is defined by setting

$$
u(x) = [I_{n_u} \; 0 \; \ldots \; 0]z(x), \tag{4}
$$

where $z(x)$ is the optimizer of the QP problem (3) for the current value of x.

In the sequel, we will make use of the dual problem of (3)

$$
\begin{aligned}
\Phi^\star(x) \triangleq \min_y \quad & \frac{1}{2}y'Hy + (Dx + d)'y && \text{(5a)}\\
\text{s.t.} \quad & y \geq 0, && \text{(5b)}
\end{aligned}
$$

where $H \triangleq GQ^{-1}G'$, $D \triangleq GQ^{-1}F + S$, $d \triangleq GQ^{-1}c + W$. The relation between the optimal primal vector $z(x)$ and optimal dual vector $y(x)$ is

$$
z(x) = -Q^{-1}(G'y(x) + Fx + c). \tag{6}
$$

A. Multiparametric solution

Rather than using a numerical QP solver to compute the optimizer $z(x)$ of (3) on-line for each given current state vector x, the basic idea of explicit MPC is to characterize the solution of the QP off-line for an entire set $X \subseteq \mathbb{R}^m$ of states x of interest,

$$
X \triangleq \{x \in \mathbb{R}^m : \; E_x x \leq e_x\}, \tag{7}
$$

that is to get z, and hence the MPC control law u, as an explicit function of x.

The main tool to get such an explicit solution is *multiparametric quadratic programming* (mpQP). For mpQP problems of the form (3), we recall the following result from [9]:

Theorem 1: Consider problem (3) with $Q = Q' > 0$ and let $X = \mathbb{R}^n$.

i) The set X_f of parameters x for which the problem is feasible is a polyhedron;

ii) The optimizer function $z^* : X_f \to \mathbb{R}^n$ is piecewise affine and continuous over X_f;

iii) If in addition matrix $\left[\begin{smallmatrix} Q & F' \\ F & Y \end{smallmatrix}\right]$ is symmetric and positive semidefinite, the value function $V^* : X_f \to \mathbb{R}$ associating with every $x \in X_f$ the corresponding optimal value of (3) is continuous, convex, and piecewise quadratic.

When $X \subset \mathbb{R}^n$, the results of Theorem 1 hold by replacing X_f with $X_f \cap X$.

An immediate corollary of Theorem 1 is that the explicit version of the MPC control law u in (4), being the first n_u components of the optimal vector $z(x)$, is also a continuous and piecewise-affine state-feedback law defined over a partition of the set $X_f \cap X$ of states into M polyhedral cells

$$u^*(x) = \begin{cases} K^1 x + h^1 & \text{if} \quad E^1 x \le e^1 \\ \quad \vdots & \qquad \vdots \\ K^M x + h^M & \text{if} \quad E^M x \le e^M. \end{cases} \tag{8}$$

An example of such a partition is reported in Figure 1 of Section VI-B. The explicit representation (8) has mapped the MPC law (4) into a lookup table of affine gains, meaning that for each given x the values computed by solving the QP (3) on-line and those obtained by evaluating (8) are exactly the same.

B. Generalization of the MPC formulation

The explicit approach described above can be extended to the following MPC setting:

$$\min_z \sum_{k=0}^{N-1} \frac{1}{2}(y_k - \mathbf{r_k})' Q_y (y_k - \mathbf{r_k}) + \frac{1}{2}\Delta u_k' R_{\Delta u} \Delta u_k + (u_k - \mathbf{u_k^r})' R_u (u_k - \mathbf{u_k^r})' + \rho_\epsilon \epsilon^2 \tag{9a}$$

$$\text{s.t. } x_{k+1} = \mathcal{A} x_k + \mathcal{B}_u u_k + \mathcal{B}_v \mathbf{v_k} \tag{9b}$$

$$x_0 = \mathbf{x_0}$$

$$y_k = \mathcal{C} x_k + \mathcal{D}_u u_k + \mathcal{D}_v \mathbf{v_k} \tag{9c}$$

$$u_k = u_{k-1} + \Delta u_k,\ k = 0, \dots, N-1 \tag{9d}$$

$$u_{-1} = \mathbf{u_{-1}}$$

$$\Delta u_k = 0,\ k = N_u, \dots, N-1 \tag{9e}$$

$$\mathbf{u_{\min}^k} \le u_k \le \mathbf{u_{\max}^k},\ k = 0, \dots, N_u - 1 \tag{9f}$$

$$\mathbf{\Delta u_{\min}^k} \le \Delta u_k \le \mathbf{\Delta u_{\max}^k},\ k = 0, \dots, N_u - 1 \tag{9g}$$

$$\mathbf{y_{\min}^k} - \epsilon V_{\min} \le y_k \le \mathbf{y_{\max}^k} + \epsilon V_{\max} \tag{9h}$$

$$k = 0, \dots, N_c - 1$$

where $R_{\Delta u} = R_{\Delta u}' > 0$, $Q_y = Q_y' \ge 0$, $R_u = R_u' \ge 0$, $\mathbf{x_0}$ is the current state, $\mathbf{v_k}$ is a vector of measured disturbances, $y_k \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{r_k} \in \mathbb{R}^{n_y}$ its corresponding reference to be tracked, Δu_k the vector of input increments, $\mathbf{u_{-1}}$ is the command input applied during the previous sampling interval, $\mathbf{u_k^r}$ the input reference, $\mathbf{u_{\min}^k}$, $\mathbf{u_{\max}^k}$, $\mathbf{\Delta u_{\min}^k}$, $\mathbf{\Delta u_{\max}^k}$, $\mathbf{y_{\min}^k}$, $\mathbf{y_{\max}^k}$ are bounds, and N, N_u, N_c are, respectively, the prediction, control, and constraint horizons. The extra variable ϵ is introduced to soften output constraints via the relaxation vectors $V_{\min}, V_{\max} > 0$ of $\mathbb{R}^{n_y}$ and penalized by the (usually large) weight ρ_ϵ in the cost function (9a).

Everything marked in bold-face in (9) can be treated as a parameter with respect to which solve the mpQP problem and obtain the explicit form of the MPC controller. For example, for a tracking problem with no anticipative action ($\mathbf{r_k} \equiv r_0$, $\forall k = 0, \dots, N-1$), no measured disturbance, fixed upper and lower bounds, the explicit solution is a continuous piecewise affine function of the parameter vector $[\mathbf{x_0}' \ \mathbf{r_0}' \ \mathbf{u_{-1}}']'$.

III. Polyhedral Computations Based on NNLS

Finding a solution to the mpQP problem (3) requires solving several problems of computational geometry, as will be detailed in Section IV. The goal of this section is to provide an alternative to existing methods that rely on the availability of a linear programming (LP) solver, building upon a standard and easy-to-code solver for the Non-Negative Least-Squares (NNLS) problem

$$\begin{aligned} r^* = \min_v \quad & \|Av - b\|_2^2 \\ \text{s.t.} \quad & v \ge 0, \end{aligned} \tag{10}$$

where $v \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $r^* \in \mathbb{R}$ is the minimum squared Euclidean norm of the residual $w^* = Av^* - b$. A well-known and simple, yet very effective, active-set method for solving the NNLS problem (10) is described in [19, p.161] and is summarized in Algorithm 1. At convergence after a finite number of steps, the algorithm provides the optimal solution vector v^*, with $v_i^* > 0$, $\forall i \in \mathcal{P}$, and $v_i^* = 0$, $\forall i \in \{1, \dots, m\} \setminus \mathcal{P}$.

Algorithm 1 NNLS solver [19, p.161]

Input: Matrices A, b.

1) $\mathcal{P} \leftarrow \emptyset$, $v \leftarrow 0$;
2) $w \leftarrow A'(Av - b)$;
3) **if** $w \ge 0$ **or** $\mathcal{P} = \{1, \dots, m\}$ **then go to** Step 11;
4) $i \leftarrow \arg\min_{i \in \{1,\dots,m\} \setminus \mathcal{P}} w_i$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{i\}$;
5) $y_\mathcal{P} \leftarrow \arg\min_{z_\mathcal{P}} \|((A')_\mathcal{P})' z_\mathcal{P} - b\|_2^2$, $y_{\{1,\dots,m\} \setminus \mathcal{P}} \leftarrow 0$;
6) **if** $y_\mathcal{P} \ge 0$ **then** $v \leftarrow y$ and **go to** Step 2;
7) $j \leftarrow \arg\min_{h \in \mathcal{P}:\ y_h \le 0} \left\{ \frac{v_h}{v_h - y_h} \right\}$;
8) $v \leftarrow v + \frac{v_j}{v_j - y_j}(y - v)$;
9) $\mathcal{I} \leftarrow \{h \in \mathcal{P} : v_h = 0\}$, $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{I}$;
10) **go to** Step 5;
11) $v^* \leftarrow v$; **end**.

Output: A vector v^* solving (10)

Algorithm 1 can be easily modified to warm-start from a set $\mathcal{P} \ne \emptyset$ of active constraints, see, e.g., [21, Algorithm 2]. Moreover, since solving Step 5 is the most time consuming operation of Algorithm 1, iterative methods have been proposed for QR factorization [19, Chap. 24] and LDL^T factorization [20] to exploit the incremental changes of the active set $\mathcal{P}$ in Steps 4 and 9.

In the sequel, we will also refer to the unconstrained problem

$$r^* = \min_v \quad \|Av - b\|_2^2 \tag{11}$$

as the (unconstrained) Least-Squares (LS) problem, whose optimizer is $v^* = A^\# b$, where $A^\#$ is a pseudoinverse of A.

As shown in next Lemma 1, a NNLS algorithm can be immediately used also to solve the Partially Non-Negative Least Squares (PNNLS) problem

$$\begin{array}{rl} r^* = \min_{v,u} & \|Av + Bu - c\|_2^2 \\ \text{s.t.} & v \geq 0,\ u \text{ free}, \end{array} \tag{12}$$

where $u \in \mathbb{R}^p$, $B \in \mathbb{R}^{m \times p}$, $c \in \mathbb{R}^m$.

Lemma 1 (PNNLS): A solution (v^*, u^*) to the PNNLS problem (12) is obtained by solving the following NNLS problem

$$\begin{array}{rl} v^* \in \ \arg\min_v & \|\bar{A}v - \bar{b}\|_2^2 \\ \text{s.t.} & v \geq 0 \end{array} \tag{13}$$

and setting $u^* = -B^\#(Av^* - c)$, where $\bar{A} \triangleq (I - BB^\#)A$, $\bar{b} \triangleq (I - BB^\#)c$, and $B^\#$ is a pseudoinverse of B.

Proof: Let $(v^*, \bar{u})$ be a solution of the PNNLS problem (12), and let $u(v) \in \arg\min_u \|Av + Bu - c\|_2^2 = -B^\#(Av - c)$ for some pseudoinverse $B^\#$ of B. Since by definition $\|Av^* + Bu(v^*) - c\|_2^2 \leq \|Av^* + Bu - c\|_2^2$ for all $u \in \mathbb{R}^p$, we have in particular that $\|Av^* + Bu(v^*) - c\|_2^2 \leq \|Av^* + B\bar{u} - c\|_2^2$, and hence $(v^*, u(v^*))$ is also an optimal solution of (12). Then, the search for optimal PNNLS solutions can be restricted to the affine subspace $u + B^\# Av = B^\# c$, and therefore a solution v^* can be computed as in (13) after substituting $u = u(v)$ in (12). ■

In the following sections we show how to address different computational geometric operations over polyhedral sets using the above NNLS and PNNLS formulations.

A. Feasibility

Lemma 2 (Feasibility): Let P be the polyhedral set $P = \{u \in \mathbb{R}^n :\ Eu \leq e\}$, $E \in \mathbb{R}^{m \times n}$, $e \in \mathbb{R}^m$. P is nonempty if and only if the PNNLS problem

$$\begin{array}{rl} (v^*, u^*) \in \arg\min_{v,u} & \|Iv + Eu - e\|_2^2 \\ \text{s.t.} & v \geq 0, u \text{ free} \end{array} \tag{14}$$

is such that $r^* = \|v^* + Eu^* - e\|_2^2 = 0$. Moreover, if P is nonempty then $u^* \in P$.

Proof: If $r^* = 0$ then $Eu^* = e - v^* \leq e$ and hence $u^* \in P$. Vice versa, if P is nonempty, for any $u^* \in P$, the pair $(u^*, e - Eu^*)$ is feasible and has a zero residual, so $r^* = 0$ for problem (14). ■

The idea of Lemma 2 can be immediately adopted to get a feasible point on a facet of P lying on the hyperplane $E_i u = e_i$ and inside convex cones, as described in the following Lemma 3 and Lemma 4, respectively.

Lemma 3 (Point on a facet): Let P be the polyhedral set $P = \{u \in \mathbb{R}^n :\ Eu \leq e\}$, $E \in \mathbb{R}^{m \times n}$, $e \in \mathbb{R}^m$ and let

$$\tilde{P}_i = P \cap \{u \in \mathbb{R}^n :\ E_i u = e_i\}.$$

Then $\tilde{P}_i \neq \emptyset$ if and only if the PNNLS problem

$$\begin{array}{rl} (v^*, u^*) \in \arg\min_{v,u} & \|C_i v + Eu - e\|_2^2 \\ \text{s.t.} & v \geq 0,\ u \text{ free} \end{array} \tag{15}$$

has a zero residual, where $v \in \mathbb{R}^{m-1}$ and matrix $C_i \in \mathbb{R}^{m \times m-1}$ is obtained by eliminating the i-th column from the identity matrix I_m. In this case, $u^* \in \tilde{P}_i$.

Proof: Assume $\tilde{P}_i \neq \emptyset$. Then a vector $\bar{u} \in \mathbb{R}^n$ exists such that $E_j \bar{u} - e_j \leq 0$, $\forall j = 1, \ldots, m$, and $E_i \bar{u} - e_i = 0$. By letting

$$\bar{v} \triangleq \begin{bmatrix} e_1 - E_1 \bar{u} \\ \vdots \\ e_{i-1} - E_{i-1}\bar{u} \\ e_{i+1} - E_{i+1}\bar{u} \\ \vdots \\ e_m - E_m \bar{u} \end{bmatrix} \geq 0$$

we get $\|C_i \bar{v} + E\bar{u} - e\|_2^2 = 0$. Then, all optimal solutions (u^*, v^*) of (15) have zero residuals. Vice versa, if v^*, u^* in (15) are such that $C_i v^* + Eu^* - e = 0$, then $Eu^* = e - C_i v^* \leq 0$ and $E_i u^* = e_i$, or equivalently $u^* \in \tilde{P}_i$. ■

Lemma 4 (Strictly feasible ray): Let $C = \{u \in \mathbb{R}^n :\ Eu \leq 0\}$ be a convex cone, $E \in \mathbb{R}^{m \times n}$. Consider the LS problem

$$u^* \in \arg\min_u \ \|Eu + E_{\|}\|_2^2 \tag{16}$$

where $E_{\|} \in \mathbb{R}^m$ is a vector whose i-th entry $(E_{\|})_i = \|E_i\|_2$. The cone C is full-dimensional if and only if $Eu^* + E_{\|} = 0$. In this case, $u^* \in C$ is such that the unit ball $B(u^*, 1) \subset C$, implying that $R = \{u \in \mathbb{R}^n :\ u = \alpha u^*,\ \alpha \geq 0\}$ is a non-extreme ray of C.

Proof: Let $Eu^* + E_{\|} = 0$ and consider a generic vector $u \in B(u^*, 1)$, that is $u = u^* + w$ for some $w \in B(0, 1)$. Then $E_i u = E_i u^* + E_i w = -\|E_i\|_2 + E_i w \leq -\|E_i\|_2 + \|E_i\|_2 = 0$, $\forall i = 1, \ldots, n$, and therefore $B(u^*, 1) \subset C$, which proves that C is full-dimensional. Vice versa, assume C is full dimensional. Then a vector $\bar{u}$ in the interior of P exists such that $d \triangleq E\bar{u} < 0$. Let $u \in \mathbb{R}^n$ such that $u_i = -\frac{\|E_i\|_2}{d_i}\bar{u}_i$. Then, $(Eu + E_{\|})_i = -\|E_i\|_2 + \|E_i\|_2 = 0$, that is $Eu + E_{\|} = 0$, and hence $\|Eu^* + E_{\|}\|_2^2 = 0$ for all optimizers u^* of (16). ■

B. Minimal hyperplane representation

Definition 1: Given a polyhedron $P = \{u \in \mathbb{R}^n :\ Eu \leq e\}$, the inequality $E_i u \leq e_i$ is said *redundant* if $P = \{u \in \mathbb{R}^n :\ E_j u \leq e_j,\ j = 1, \ldots, i-1, i+1, \ldots, m\}$.

Definition 2: A redundant inequality is said *weakly redundant* if it is redundant and $P \cap \{u \in \mathbb{R}^n :\ E_i u = e_i\} \neq \emptyset$, or *strongly redundant* otherwise.

Definition 3: A polyhedron $P = \{u \in \mathbb{R}^n :\ Eu \leq e\}$ is said in *minimal H-representation* if no inequality $E_i u \leq e_i$ in its hyperplane description is redundant, or in *weakly minimal H-representation* if no inequality $E_i u \leq e_i$ is strongly redundant.

In the sequel, given a polyhedron $P = \{u \in \mathbb{R}^n :\ Eu \leq e\}$ and its i-th inequality $E_i u \leq e_i$ we will denote by

$$E^i \ \triangleq \ \begin{bmatrix} E_1 \\ \vdots \\ E_{i-1} \\ -E_i \\ E_{i+1} \\ \vdots \\ E_m \end{bmatrix},\ e^i \triangleq \begin{bmatrix} e_1 \\ \vdots \\ e_{i-1} \\ -e_i \\ e_{i+1} \\ \vdots \\ e_m \end{bmatrix}, \tag{17a}$$

$$\begin{aligned} P^i &\triangleq \{u \in \mathbb{R}^n : \ E^i u \le e^i\}, \quad (17b)\\ \bar{P}^i &\triangleq \{u \in \mathbb{R}^n : \ E_j u \le e_j, j \in \{1,\dots,m\} \setminus \{i\}\}, \end{aligned}$$

for $i = 1,\dots,m$. Clearly, $\tilde{P}_i = P^i \cap \{u \in \mathbb{R}^n : \ E_i u = e_i\}$ and $\bar{P}^i = P \cup P^i$.

Lemma 5: Let $P = \{u \in \mathbb{R}^n : \ Eu \le e\}$, $E \in \mathbb{R}^{m\times n}$, $e \in \mathbb{R}^m$, be a nonempty polyhedron. An inequality $E_i u \le e_i$ is strongly redundant if and only if the polyhedron P^i defined in (17) is empty.

Proof: If $P^i = \emptyset$ and therefore $\bar{P}^i = P \cup P^i = P$, the i-th inequality is redundant. We prove that such an inequality is also strongly redundant by contradiction. Assume the inequality $E_i u \le e_i$ is weakly redundant. Then there exists a vector u such that $u \in \tilde{P}_i \subseteq P^i$, and therefore $P^i \ne \emptyset$. Since this is a contradiction, $E_i u \le e_i$ is strongly redundant. Vice versa, let $E_i u \le e_i$ be strongly redundant, that is $\bar{P}^i = P$ and $\tilde{P}_i = \emptyset$, and assume $P^i \ne \emptyset$. Let $u \in P^i$. As $\tilde{P}_i = \emptyset$, $E_i u > e_i$, and hence $u \notin P$, which yields $P = \bar{P}^i \setminus P^i \cup \tilde{P}_i = \bar{P}^i \setminus P^i \ne \bar{P}^i$, a contradiction proving that $P^i = \emptyset$. ■

Theorem 2 (Redundancy Elimination I): Let $P = \{u \in \mathbb{R}^n : \ Eu \le e\}$, $E \in \mathbb{R}^{m\times n}$, $e \in \mathbb{R}^m$, be a nonempty polyhedron. The weakly minimal H-representation of P can be obtained by collecting all constraints $E_i u \le e_i$ such that the PNNLS problem

$$\begin{array}{ll} \min_{v,u} & \|Iv + E^i u - e^i\|_2^2 \\ \text{s.t.} & v \ge 0, \ u \text{ free} \end{array} \quad (18)$$

has zero residual, where E^i, e^i are defined in (17), $i = 1,\dots,m$.

Proof: By Lemma 2 applied to P^i, Problem (18) has a zero residual $w = v + E^i u - e_i$ iff $P^i \ne \emptyset$; moreover, by Lemma 5, $P^i \ne \emptyset$ iff inequality $E_i u \le e_i$ is not strongly redundant. Therefore, all non-redundant and weakly-redundant inequalities are identified by detecting the indices $i = 1,\dots,m$ for which Problem (18) has a zero residual. ■

Note that, from a numerical point of view, by slightly perturbing the coefficient e_i to $e_i + \|E_i\|_2\delta$ in (17), where $\delta > 0$ is a small tolerance, weakly redundant inequalities become strongly redundant and can be eliminated to obtain a minimal H-representation.

In alternative to solving (18), one can test hyperplane redundancy in accordance with the following theorem.

Theorem 3 (Redundancy Elimination II): Let $P = \{u \in \mathbb{R}^n : \ Eu \le e\}$, $E \in \mathbb{R}^{m\times n}$, $e \in \mathbb{R}^m$, be a nonempty polyhedron. The weakly minimal H-representation of P can be obtained by collecting all the inequalities $E_i u \le e_i$ such that the NNLS problem

$$\begin{array}{ll} \min_{v} & \left\| \begin{bmatrix} E^i & e^i \end{bmatrix}' v - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix} \right\|_2^2 \\ \text{s.t.} & v \ge 0 \end{array} \quad (19)$$

has nonzero residual, where E^i, e^i are defined in (17).

Proof: By Lemma 5, strong-redundancy of the i-th inequality $E_i u \le e_i$ is equivalent to emptiness of $P^i = \{u \in \mathbb{R}^n : \ E^i u \le e^i\}$. We need to prove that P^i is empty iff Problem (19) has zero residual.

If Problem (19) has zero residual, then there exists a vector $v \in \mathbb{R}^{m+1}$ such that $(E^i)'v = 0$, $v \ge 0$, $(e^i)'v = -1$. By Farkas's Lemma [22, p. 201], this is equivalent to infeasibility of $E^i u \le e^i$. Vice versa, if $E^i u \le e^i$ is infeasible then a vector $\bar{v}$ exists such that $(E^i)'\bar{v} = 0$, $\bar{v} \ge 0$, $(e^i)'\bar{v} < 0$. By defining $v \triangleq -\frac{1}{(e^i)'\bar{v}}\bar{v}$, we get

$$\begin{bmatrix} (E^i)'v \\ (e^i)'v \end{bmatrix} = -\frac{1}{(e^i)'\bar{v}} \begin{bmatrix} (E^i)'\bar{v} \\ (e^i)'v \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix},$$

which proves that Problem (19) has a zero residual. ■

Next Theorem 4 provides a third method to check redundancy that is based on testing emptiness of the lower-dimensional facets $\tilde{P}_i$.

Theorem 4 (Redundancy Elimination III): Let $P = \{u \in \mathbb{R}^n : \ Eu \le e\}$, $E \in \mathbb{R}^{m\times n}$, $e \in \mathbb{R}^m$, be a nonempty polyhedron. The weakly minimal H-representation of P can be obtained by collecting all the inequalities $E_i u \le e_i$ such that the corresponding PNNLS problem (15) has a zero residual.

Proof: If Problem (15) has a zero residual, by Lemma 3 $\tilde{P}_i \ne \emptyset$, which implies $P^i \ne \emptyset$, and hence, by Lemma 5, the inequality $E_i u \le e_i$ is not strongly redundant. Vice versa, if $E_i u \le e_i$ is not strongly redundant then $P^i \ne 0$. Take two arbitrary vectors $u_1 \in P^i$, $u_2 \in P \setminus P^i$. Clearly, $E_j u_h \le e_j$, $\forall j \in \{1,\dots,m\} \setminus \{i\}$, $h = 1,2$, and $E_i u_1 = e_i + v_1$, $E_i u_2 = e_i - v_2$ for some scalars $v_1, v_2 \ge 0$. Let $\alpha \triangleq \frac{v_2}{v_1+v_2}$ and set $u \triangleq \alpha u_1 + (1-\alpha)u_2$. Then $E_i u = \alpha E_i u_1 + (1-\alpha)E_i u_2 = \alpha e_i + \alpha v_1 + (1-\alpha)e_i - (1-\alpha)v_2 = \alpha v_1 + e_i - v_2 + \alpha v_2 = e_i$, which implies $u \in \tilde{P}_i$. Therefore, by Lemma 3, Problem (15) has a zero residual. ■

Theorems 2, 3 and 4 provide three alternative methods to obtain the weakly minimal H-representation of a given polyhedron P. By referring to the *dimension* of the NNLS problem (10) as the pair (n,m), that is the dimensions of the vector of unknowns and of the residual, they require solving m NNLS problems of dimension (m,m), $(m,m+1)$, $(m-1,m)$ respectively. The approaches of Theorems 2 and 4 also require the computation of a pseudoinverse matrix. Section VI-A below compares the proposed three methods numerically (the results are summarized in Table I). According to our numerical experience, Theorem 4 is the fastest one, while the approaches of Theorems 2, 3 are numerically more robust.

C. Full-dimensionality

Definition 4: A polyhedron P is said *full-dimensional* if there exists a ball $B(u,\epsilon) \subset P$ for some $u \in P$ and $\epsilon > 0$.

Theorem 5: Let $P = \{u \in \mathbb{R}^n : \ Eu \le e\}$, $E \in \mathbb{R}^{m\times n}$, $e \in \mathbb{R}^m$, be a nonempty polyhedron, let $u_0 \in P$, and let $\gamma_i \triangleq e_i - E_i u_0$, $i = 1,\dots,m$. Let $\mathcal{I}$ denote the set of indices i such that $\gamma_i = 0$ (active inequalities at u_0). The following statements are true:

i) If $\mathcal{I} = \emptyset$, then P is full-dimensional. Moreover, $B(u_0,\epsilon) \subset P$ for all $\epsilon \in \mathbb{R}$ such that $0 < \epsilon \le \min_{i=1,\dots,m:\ E_i \ne 0} \left\{ \frac{\gamma_i}{\|E_i\|_2} \right\}$.

ii) If $\mathcal{I} \ne \emptyset$, let u^* be a solution of the LS problem (16) with E replaced by $E_{\mathcal{I}}$ and $E_{\|}$ by $(E_{\|})_{\mathcal{I}}$. Then P is full-dimensional if and only if the residual $E_{\mathcal{I}} u^* + (E_{\mathcal{I}})_{\|} =$

0. In this case, by defining

$$\alpha_{\mathcal{I}} \triangleq \min_{i \notin \mathcal{I}:\ E_i u^* > 0} \frac{\gamma_i}{E_i u^*}, \tag{20}$$

where $\alpha_{\mathcal{I}} \triangleq +\infty$ if $E_i u^* \leq 0$ for all $i = 1, \ldots, m$, and

$$\epsilon_\alpha \triangleq \min_{i \in \{1,\ldots,m\}} \frac{e_i - E_i u_0 - \alpha E_i u^*}{\|E_i\|_2}, \tag{21}$$

it holds that $\alpha_{\mathcal{I}} > 0$, $0 < \epsilon_\alpha \leq \alpha$, and $B(u_0 + \alpha u^*, \epsilon) \subset P$ for all α, ϵ such that $0 < \alpha < \alpha_{\mathcal{I}}$ and $\epsilon \leq \epsilon_\alpha$.

Proof: Part (i). If $\mathcal{I} = \emptyset$ then $Eu_0 < e$, and hence $\bar{\epsilon} \triangleq \min_{i=1,\ldots,m:\ E_i \neq 0} \left\{ \frac{\gamma_i}{\|E_i\|_2} \right\} > 0$. Moreover, for any $w \in B(u_0, \bar{\epsilon})$ we have $E_i w - e_i = -\gamma_i + E_i(w - u_0) \leq -\gamma_i + \bar{\epsilon}\|E_i\|_2$. Hence, if $E_i = 0$ then $E_i w - e_i \leq -\gamma_i \leq 0$, while if $E_i \neq 0$ then $E_i w - e_i \leq \|E_i\|_2 \left(-\frac{\gamma_i}{\|E_i\|_2} + \bar{\epsilon} \right) \leq 0$. This proves that $B(u_0, \epsilon) \subset P$, $\forall \epsilon \leq \bar{\epsilon}$.

Part (ii). Let $\mathcal{I} \neq \emptyset$. Assume that P is full-dimensional and by contradiction that the residual $E_{\mathcal{I}} u^* + (E_\|)_{\mathcal{I}} \neq 0$. By Lemma 4 the cone $C_0 \triangleq \{u \in \mathbb{R}^n :\ E_{\mathcal{I}}(u - u_0) \leq 0\}$ is not full-dimensional, and therefore the set $C \triangleq \{u_0\} \oplus C_0$ is not full-dimensional. Consider any $u \in P$. Then $e_{\mathcal{I}} \geq E_{\mathcal{I}} u = E_{\mathcal{I}} u + (e_{\mathcal{I}} - E_{\mathcal{I}} u_0)$, which implies $E_{\mathcal{I}}(u - u_0) \leq 0$ and hence $u - u_0 \in C_0$, or $u \in C$. Therefore, $P \subseteq C$, and since C is not full-dimensional, P is also not full-dimensional, a contradiction.

Vice versa, assume the residual $E_{\mathcal{I}} u^* + (E_\|)_{\mathcal{I}} = 0$. We want to prove that P is full-dimensional. If no $i \notin \mathcal{I}$ exists such that $E_i u^* > 0$ then $\alpha_{\mathcal{I}} = +\infty > 0$. Otherwise, $\alpha_{\mathcal{I}} > 0$ by definition, being the minimum of ratios of positive scalars. Consider a generic α, $0 < \alpha < \alpha_{\mathcal{I}}$ and the quantity $\sigma_i \triangleq \frac{\gamma_i - \alpha E_i u^*}{\|E_i\|_2}$. For $i \in \mathcal{I}$, $\sigma_i = \alpha > 0$. For $i \notin \mathcal{I}$, if $E_i u^* \leq 0$ then $\sigma_i = \frac{\gamma_i}{\|E_i\|_2} > 0$; otherwise, if $E_i u^* > 0$, then $\sigma_i \geq \frac{E_i u^*}{\|E_i\|_2} \left(\frac{\gamma_i}{E_i u^*} - \alpha \right) \geq \frac{E_i u^*}{\|E_i\|_2} (\alpha_{\mathcal{I}} - \alpha) > 0$. Hence, $\epsilon_\alpha = \min_{i \in \{1,\ldots,m\}} \sigma_i > 0$ and moreover $\epsilon_\alpha = \min_{i \in \{1,\ldots,m\}} \left\{ \frac{e_i - E_i u_0}{\|E_i\|_2} - \alpha \frac{E_i u^*}{\|E_i\|_2} \right\} \geq \alpha$. Let ϵ such that $0 < \epsilon < \epsilon_\alpha$, let $u_\alpha \triangleq u_0 + \alpha u^*$, and let $w \in B(u_\alpha, \epsilon)$. For $i \in \mathcal{I}$, for all $w \in B(u_\alpha, \epsilon)$ we have $E_i w - e_i = E_i u_0 - e_i + \alpha E_i u^* + E_i(w - u_\alpha) \leq -\gamma_i + \alpha E_i u^* + \epsilon \|E_i\|_2 = \|E_i\|_2 \left(-\frac{\gamma_i - \alpha E_i u^*}{\|E_i\|_2} + \epsilon \right) \leq 0$ for all $\epsilon \leq \epsilon_\alpha$. Hence, $B(u_\alpha, \epsilon) \subset P$, for all $\epsilon \leq \epsilon_\alpha$, and for all $\alpha < \alpha_{\mathcal{I}}$, and therefore P is full-dimensional. ■

Theorem 5 provides a simple method, based on solving just a single LS problem, to determine if a given polyhedron P is full-dimensional, once a vector $u_0 \in P$ is known. However, Theorem 5 is not able to quantify the largest ball $B(u_c, \epsilon_c)$ included in P, that is the Chebychev center u_c and radius ϵ_c of P, or if P contains a ball of a given radius ρ that is larger than the radius determined by Theorem 5. Algorithm 2 shows how to compute u_c, ϵ_c via PNNLS and bisection. At each iteration, the interval $[r_{\min}, r_{\max}]$ is halved, which ensures a good convergence of the algorithm.

Remark 1: By Lemma (2), the PPNLS problem (22) solved at Step 2.2 determines if the set of inequalities $Eu + rE_\| \leq e$ is satisfiable for some $u \in \mathbb{R}^m$, which implies that the ball $B(u, r) \subset P$. If one is only interested in testing if P contains a ball larger or equal than a given radius ρ, Algorithm 2 reduces to simply executing Step 2.2 for $r = \rho$.

Algorithm 2 Chebychev center and radius via PNNLS and bisection

Input: $P = \{u :\ Eu \leq e\}$, $u_0 \in P$, tolerance $\delta > 0$.

1. $r_{\min} \leftarrow \min_i \frac{e_i - E_i u_0}{\|E_i\|_2}$, $r_{\max} \leftarrow +\infty$, $u_c = \leftarrow u_0$;
2. **while** $r_{\max} - r_{\min} \geq \delta$ **do**
   2.1 $r \leftarrow \frac{r_{\min} + r_{\max}}{2}$;
   2.2 **solve** the PNNLS problem
   $$\begin{array}{ll} (v^*, u^*) \in \arg\min_{v,u} & \|Iv + Eu - (e^i - rE_\|)\|_2^2 \\ & \text{s.t.}\quad v \geq 0,\ u \text{ free}; \end{array} \tag{22}$$
   2.3 **if** $\|v^* + Eu^* - (e - rE_\|)\|_2 \leq \delta$ **then** $r_{\min} \leftarrow r$, $u_c \leftarrow u^*$ **else** $r_{\max} \leftarrow r$;
3. $\epsilon_c \leftarrow r_{\min}$; **end**.

Output: $B(u_c, \epsilon_c)$ = largest ball (within precision δ) in P.

Remark 2: Algorithm 2 can be easily modified to determine lower dimensional balls contained in facets of P, say the facet contained in the hyperplane $E_i u = e_i$, with an approach similar to the one proposed by Lemma 3, by simply changing the PNNLS problem to the one of minimizing $\|Eu + C_i v - (e^i - rE_\|^i)\|_2^2$, where $E_\|^i$ is obtained from $E_\|$ by zeroing its i-th entry.

Remark 3: The PNNLS solver for (12) can be also employed to solve the Linear Program (LP) $\min_{\{u:\ Gu \leq W\}} f'u$, $G \in \mathbb{R}^{q \times n}$, $W \in \mathbb{R}^q$, $f \in \mathbb{R}^n$, by considering the dual LP problem $\max_{\{y \geq 0:\ G'y = -f\}} -W'y$ and imposing the optimality conditions $f' + G'y = 0$, $Gu + s = W$, $s \geq 0$, $y \geq 0$, and the complementarity condition $y'(Gu - W) = 0$, where the latter is equivalent to the condition $f'u + W'y = 0$ (zero duality gap). As such conditions are all linear conditions with respect to u, y, s, finding an optimal solution of the LP problem is equivalent to determining a solution u of (12) with zero residual, for matrices $A = \left[\begin{smallmatrix} W' & 0 \\ 0 & I \\ G' & 0 \end{smallmatrix}\right]$, $B = \left[\begin{smallmatrix} f' \\ G \\ 0 \end{smallmatrix}\right]$, $c = \left[\begin{smallmatrix} 0 \\ W \\ -f \end{smallmatrix}\right]$. Then, this approach can be used as an alternative to Algorithm 2 to finding the Chebychev radius of the polyhedron P via the LP $\max_{\{u,\epsilon:\ Eu \leq e + E_\| \epsilon\}} \epsilon$.

Remark 4: Practical evidence has shown that the hyperplane representation $\{Eu \leq e\}$ should be normalized before applying any of the results for polyhedral computation presented in this section to help avoiding ill-conditioning issues, that is to replace each row E_i of matrix E by $\frac{1}{\|E_i\|} E_i$ and each component e_i by $\frac{1}{\|E_i\|} e_i$. The cost of this operation is negligible, and often done anyway (see, e.g., Algorithm 2).

In the next section we will use the above results as the building blocks of a mpQP algorithm. They can be used also for other polyhedral computations such as computing the union of convex polyhedra [23, Algorithm 4.1] and to remove redundancy in Fourier-Motzkin elimination methods [24] for projection of polyhedra.

IV. Multiparametric QP based on NNLS

A few algorithms have been proposed in the literature to solve the mpQP problem (3). All of them rely on the fact that for a given combination of active constraints at optimality, the optimal solution $z(x)$ is a linear function of x, and that such a function is the optimal solution within an entire polyhedral set (also called *critical region*) of the parameter space; then, they enumerate all possible optimal combinations of active constraints (which are a finite number) until the entire set $X \subseteq \mathbb{R}^m$ defined in (7) has been characterized.

While most of the proposed approaches construct the solution by exploiting the Karush-Kuhn-Tucker (KKT) conditions for optimality of the *primal* problem (3), in this paper we refer instead most often to the optimality conditions for the *dual* problem (5)

$$w = Hy + Dx + d \tag{23a}$$
$$0 \leq y \perp w \geq 0. \tag{23b}$$

Note that problem (23) corresponds to a parametric linear complementarity problem (mpLCP) in the variables y, w [13].

Definition 5: A combination $\mathcal{I} \subseteq \{1, \ldots, q\}$ is said *optimal* if (23a) is satisfied for some $x \in \mathbb{R}^m$, $y \in \mathbb{R}^q$, $w \in \mathbb{R}^q$ with $y_\mathcal{I} \geq 0$ and $w_\mathcal{I} = 0$.

Definition 6: Given an optimal combination $\mathcal{I} \subseteq \{1, \ldots, q\}$, the corresponding *critical region* $CR_\mathcal{I}$ is the set of all parameter vectors $x \in \mathbb{R}^m$ such that (23a) is satisfied for some $y \in \mathbb{R}^q$, $w \in \mathbb{R}^q$ with $y_\mathcal{I} \geq 0$ and $w_\mathcal{I} = 0$. When solving an mpQP problem, we are interested in collecting all full-dimensional critical regions $CR_\mathcal{I}$ (in minimal H-representation) such that $CR_\mathcal{I} \cap X \neq \emptyset$.

Given a vector $x \in \mathbb{R}^n$, an optimal combination $\mathcal{I}$, and a pair (y, w) satisfying (23a), by (6) the corresponding optimal solution is

$$z(x) = -Q^{-1}(G'_\mathcal{I} y_\mathcal{I}(x) + Fx + c). \tag{24}$$

By (23a) and recalling the definitions of H, D, d in (5) we get that $w(x) = GQ^{-1}G'_\mathcal{I} y_\mathcal{I}(x) + (GQ^{-1}F + S)x + GQ^{-1}c + W = -Gz + Sx + W$ is the slack vector, so that $w(x) \geq 0$ is the condition of primal feasibility. Therefore, $w_\mathcal{I} = 0$ is equivalent to $G_\mathcal{I} z = S_\mathcal{I} x + W_\mathcal{I}$, so that $\mathcal{I}$ determines an optimal *active set* for the primal QP problem (3).

Definition 7 ([25], Def. 2): For an active set $\mathcal{I} \subseteq \{1, \ldots, q\}$, we say that the linear independence constraint qualification (LICQ) holds for $\mathcal{I}$ if the set of active constraint gradients are linearly independent, i.e., $G_\mathcal{I}$ has full row-rank.

Lemma 6 (Critical region): Let $\mathcal{I} \in \{1, \ldots, q\}$ be an optimal combination, and assume LICQ is satisfied for $\mathcal{I}$. Then $\mathcal{I}$ is optimal for all vectors x contained in the *critical region* $CR_\mathcal{I}$

$$CR_\mathcal{I} = \{x \in \mathbb{R}^m : \ y_\mathcal{I}(x) \geq 0, \ w_\mathcal{J}(x) \geq 0\} \tag{25a}$$

where $\mathcal{J} = \{1, \ldots, q\} \setminus \mathcal{I}$, and

$$y_\mathcal{I}(x) = -H_{\mathcal{I}\mathcal{I}}^{-1}(D_\mathcal{I} x + d_\mathcal{I}) \tag{25b}$$
$$w_\mathcal{J}(x) = H_{\mathcal{J}\mathcal{I}} y_\mathcal{I}(x) + D_\mathcal{J} x + d_\mathcal{J} \tag{25c}$$

are the dual optimizer and slack vector corresponding to x, respectively.

Proof: Since $Q > 0$ and $G_\mathcal{I}$ is full row rank, then $H_{\mathcal{I}\mathcal{I}} > 0$ and is invertible. The proof follows immediately by the KKT conditions (23) for the dual QP problem (5). ∎

Note that in case $\mathcal{I} = \emptyset$ we have $y(x) = 0$, $w(x) = Dx + d$, $z(x) = -Q^{-1}(Fx + c)$.

In case the LICQ condition is violated for a given $\mathcal{I}$, we refer to a condition of *(primal) degeneracy*, in which the dual solution y may not be uniquely defined as a function of x. The following result allows one to rule out critical regions that are not full-dimensional.

Theorem 6 (Degeneracy): Let $\mathcal{I} \subseteq \{1, \ldots, q\}$ be an optimal combination, $n_\mathcal{I} \triangleq \#\mathcal{I}$, and let $r < n_\mathcal{I}$ the rank of $G_\mathcal{I}$ (LICQ condition violated). Let $G_\mathcal{I} E_\mathcal{I} = Q_\mathcal{I} R_\mathcal{I}$ be a QR decomposition of $G_\mathcal{I}$, where $Q_\mathcal{I} \in \mathbb{R}^{n_\mathcal{I} \times n_\mathcal{I}}$, $Q'_\mathcal{I} Q_\mathcal{I} = I$, $Q_\mathcal{I} = [Q'_1 \ Q'_2]$, $Q_1 \in \mathbb{R}^{r \times n_\mathcal{I}}$, $Q_2 \in \mathbb{R}^{(n_\mathcal{I} - r) \times n_\mathcal{I}}$, $R_\mathcal{I} \in \mathbb{R}^{n_\mathcal{I} \times q}$ is such that $R_\mathcal{I} = \left[\begin{smallmatrix} R_1 & R_2 \\ 0 & 0 \end{smallmatrix}\right]$, with $R_1 \in \mathbb{R}^{r \times r}$ upper triangular and $\det R_1 \neq 0$, $R_2 \in \mathbb{R}^{r \times (q-r)}$, and $E_\mathcal{I}$ is a permutation matrix.

i) If

$$Q_2 \left[\begin{matrix} D_\mathcal{I} & d_\mathcal{I} \end{matrix} \right] \neq 0 \tag{26}$$

then $CR_\mathcal{I}$ is not a full-dimensional critical region.

ii) Let $M_\mathcal{I} = -\left([R_1 \ R_2] E_\mathcal{I}^{-1} Q^{-1} (E_\mathcal{I}^{-1})' [R_1 \ R_2]'\right)^{-1} Q_1$. Then any vector $y(x)$ such that $y_\mathcal{J}(x) = 0$, $\mathcal{J} = \{1, \ldots, q\} \setminus \mathcal{I}$, and $y_\mathcal{I}(x) = Q_\mathcal{I} \left[\begin{smallmatrix} M_\mathcal{I}(D_\mathcal{I} x + d_\mathcal{I}) \\ s_2 \end{smallmatrix}\right]$ is a dual solution of (5) for any $s_2 \in \mathbb{R}^{n_\mathcal{I} - r}$ such that $y_\mathcal{I}(x) \geq 0$, and

$$z(x) = -Q^{-1}(Fx + c + (E_\mathcal{I}^{-1})' [R_1 \ R_2]' M_\mathcal{I}(D_\mathcal{I} x + d_\mathcal{I})) \tag{27}$$

is the primal solution of (3). Moreover $CR_\mathcal{I}$ is the projection onto $\mathbb{R}^n$ of the set

$$P = \left\{ \left[\begin{smallmatrix} x \\ s_2 \end{smallmatrix}\right] \in \mathbb{R}^{m + n_\mathcal{I} - r} : \ Q_\mathcal{I} \left[\begin{smallmatrix} M_\mathcal{I}(D_\mathcal{I} x + d_\mathcal{I}) \\ s_2 \end{smallmatrix}\right] \geq 0 \right\}. \tag{28}$$

Proof: Let $\tilde{Q}_\mathcal{I} = E_\mathcal{I}^{-1} Q^{-1} (E_\mathcal{I}^{-1})'$, so that $H_{\mathcal{I}\mathcal{I}} = G_\mathcal{I} E_\mathcal{I} \tilde{Q}_\mathcal{I} E'_\mathcal{I} G'_I$. By (23a), $w_\mathcal{I} = 0$ leads to $H_{\mathcal{I}\mathcal{I}} y_\mathcal{I} + D_\mathcal{I} x + d_\mathcal{I} = 0$, and hence, by setting $s_\mathcal{I} \triangleq Q'_\mathcal{I} y_\mathcal{I} = \left[\begin{smallmatrix} s_1 \\ s_2 \end{smallmatrix}\right]$, $s_1 \in \mathbb{R}^r$, $s_2 \in \mathbb{R}^{n_\mathcal{I} - r}$, we get $R_\mathcal{I} \tilde{Q}_\mathcal{I} R'_\mathcal{I} s_\mathcal{I} + Q'_\mathcal{I}(D_\mathcal{I} x + d_\mathcal{I}) = 0$. The latter leads to

$$\left[\begin{matrix} R_1 & R_2 \end{matrix} \right] \tilde{Q}_\mathcal{I} \left[\begin{matrix} R_1 & R_2 \end{matrix} \right]' s_1 + Q_1(D_\mathcal{I} x + d_\mathcal{I}) = 0 \tag{29a}$$
$$Q_2(D_\mathcal{I} x + d_\mathcal{I}) = 0. \tag{29b}$$

(i) If condition (26) is satisfied, then $Q_2 D_\mathcal{I} \neq 0$ or $Q_2 d_\mathcal{I} \neq 0$, or both. If $Q_2 D_\mathcal{I} \neq 0$, clearly the set of vectors x satisfying (29b) is not full-dimensional, and therefore $CR_\mathcal{I}$ is not full-dimensional, because (23a) can only be satisfied on an affine subspace of dimension smaller than m. Otherwise, if $Q_2 D_\mathcal{I} = 0$ and $Q_2 d_\mathcal{I} \neq 0$, then (29b) has no solution, and $CR_\mathcal{I}$ is empty.

(ii) Since $\tilde{Q}_\mathcal{I}$ and R_1 are nonsingular, matrix $\left[\begin{matrix} R_1 & R_2 \end{matrix} \right] \tilde{Q}_\mathcal{I} \left[\begin{matrix} R_1 & R_2 \end{matrix} \right]'$ is invertible, which leads to $s_1(x) = M_\mathcal{I}(D_\mathcal{I} x + d_\mathcal{I})$, and hence $y_\mathcal{I}(x) = Q_\mathcal{I} \left[\begin{smallmatrix} M_\mathcal{I}(D_\mathcal{I} x + d_\mathcal{I}) \\ s_2 \end{smallmatrix}\right]$. Then $G'_\mathcal{I} y_\mathcal{I}(x) = (Q_\mathcal{I} R_\mathcal{I} E_\mathcal{I}^{-1})' Q_\mathcal{I} s_\mathcal{I} = (E_\mathcal{I}^{-1})' \left[\begin{smallmatrix} R'_1 & 0 \\ R'_2 & 0 \end{smallmatrix}\right] s_\mathcal{I}(x) = (E_\mathcal{I}^{-1})' \left[\begin{smallmatrix} R_1 & R_2 \end{smallmatrix}\right]' s_1(x)$. Replacing $G'_\mathcal{I} y_\mathcal{I}(x)$ in (24) leads to (27). The set P in (28) is obtained by simply imposing primal feasibility of $z(x)$ and dual feasibility of $y(x)$. The critical region $CR_\mathcal{I} = \{x : \ \exists s_2 \text{ such that } \left[\begin{smallmatrix} x \\ s_2 \end{smallmatrix}\right] \in P\}$ can be computed by projecting P onto $\mathbb{R}^m$ to get rid of the $n_\mathcal{I} - r$ components of s_2. ∎

The need for polyhedral projections was reported in [9, Sect. 4.1.1], in which the condition $Q_2 S_{\mathcal{I}} = 0$ was also recognized as a condition for full-dimensionality of $CR_{\mathcal{I}}$. A result related to Theorem 6 was shown in [10, Th. 3].

Lemma 7: Under the same hypotheses of Theorem 6, the condition

$$\mathrm{rank}[G_{\mathcal{I}} \ \ -S_{\mathcal{I}} \ \ -W_{\mathcal{I}}] > r \tag{30}$$

implies that $CR_{\mathcal{I}}$ is not a full-dimensional critical region.

Proof: Since $\bar{r} \triangleq \mathrm{rank}[G_{\mathcal{I}} \ -S_{\mathcal{I}} \ -W_{\mathcal{I}}] = \mathrm{rank}[G_{\mathcal{I}}E_{\mathcal{I}} \ -S_{\mathcal{I}}E_{\mathcal{I}} \ -W_{\mathcal{I}}E_{\mathcal{I}}] = \mathrm{rank}[R_{\mathcal{I}} \ -Q'_{\mathcal{I}}S_{\mathcal{I}}E_{\mathcal{I}} \ -Q'_{\mathcal{I}}W_{\mathcal{I}}E_{\mathcal{I}}]$, then

$$\bar{r} = \mathrm{rank}\begin{bmatrix} R_1 & R_2 & -Q_1 S_{\mathcal{I}} E_{\mathcal{I}} & -Q_1 W_{\mathcal{I}} E_{\mathcal{I}} \\ 0 & 0 & -Q_2 S_{\mathcal{I}} E_{\mathcal{I}} & -Q_2 W_{\mathcal{I}} E_{\mathcal{I}} \end{bmatrix} \tag{31}$$

Since $Q_2[D_{\mathcal{I}} \ d_{\mathcal{I}}] = Q_2 G_{\mathcal{I}} Q^{-1}[F \ c] + Q_2[S_{\mathcal{I}} \ W_{\mathcal{I}}]$ and $Q_2 G_{\mathcal{I}} = [0 \ I]Q'_{\mathcal{I}} Q_{\mathcal{I}} R_{\mathcal{I}} = [0 \ I]\left[\begin{smallmatrix} R_1 & R_2 \\ 0 & 0 \end{smallmatrix}\right] = 0$, then $Q_2 D_{\mathcal{I}} = Q_2 S_{\mathcal{I}}$ and $Q_2 d_{\mathcal{I}} = Q_2 W_{\mathcal{I}}$. As R_1 is upper triangular, $\det R_1 \neq 0$, and $E_{\mathcal{I}}$ is nonsingular, the condition in (26) is satisfied iff $\bar{r} > r$. ■

Lemma 7 is a slight generalization of the result of [25, Th. 3]. This states that when the LICQ condition is violated ($n_{\mathcal{I}} > r$), then the condition $\bar{r} = n_{\mathcal{I}} > r$, is an indicator that $CR_{\mathcal{I}}$ is not full-dimensional, while Lemma (7) only requires $\bar{r} > r$ (clearly, $\bar{r} \leq n_{\mathcal{I}}$).

As most existing algorithms proposed in the literature, the mpQP algorithm described in this paper determines all critical regions that intersect the given set $X \subseteq \mathbb{R}^n$ of interest defined in (7), by first determining an initial optimal combination $\mathcal{I}_0$ and its corresponding $CR_{\mathcal{I}_0}$, finding all neighboring regions $CR_{\mathcal{I}}$ of $CR_{\mathcal{I}_0}$ corresponding to new combinations $\mathcal{I}$, and then proceeds recursively until no optimal combination $\mathcal{I}$ exists such that $CR_{\mathcal{I}} \cap X \neq \emptyset$.

We focus first on determining an initial optimal combination $\mathcal{I}_0$. In case the polyhedron $CR_{\emptyset} = \{x \in \mathbb{R}^m : G(-Q^{-1}(Fx+c)) \leq W + Sx, \ E_x x \leq e_x\}$ is nonempty, $\mathcal{I}_0 = \emptyset$ is a valid initial combination. Otherwise, the following Lemma 8 provides a general method to determine $\mathcal{I}_0$.

Lemma 8: Consider problem (3) with $Q = Q' > 0$ and let $X = \{x \in \mathbb{R}^m : E_x x \leq e_x\} \subseteq \mathbb{R}^m$. Consider the QP problem

$$\begin{array}{ll} \min_{\left[\begin{smallmatrix} z \\ x \end{smallmatrix}\right]} & \frac{1}{2}z'Qz + (Fx+c)'z + \frac{1}{2}(x-\bar{x})'B(x-\bar{x}) \\ \text{s.t.} & Gz - Sx \leq W \\ & E_x x \leq e_x \end{array} \tag{32}$$

where $B \in \mathbb{R}^{m \times m}$, $B = B' \geq 0$, $\bar{x} \in \mathbb{R}^m$ are an arbitrary matrix and vector, respectively. If (32) is infeasible, then $X_f \cap X = \emptyset$, where X_f is the set of parameter vectors x for which the QP problem (3) admits a solution. Otherwise, for any solution $\left[\begin{smallmatrix} z_0 \\ x_0 \end{smallmatrix}\right]$ of (32) the combination $\mathcal{I}_0$ such that $G_{\mathcal{I}_0} z_0 = S_{\mathcal{I}_0} x_0 + W_{\mathcal{I}_0}$ is an optimal combination of active constraints for problem (3).

Proof: The proof is rather simple. If (32) is infeasible, then no vector $x \in X$ exists such that $Gz \leq W + Sx$, and therefore $X \cap X_f = \emptyset$. If (32) admits a solution $\left[\begin{smallmatrix} z_0 \\ x_0 \end{smallmatrix}\right]$, then z_0 is also the unique solution of (3) for $x = x_0$, otherwise a different vector z_1 solving (3) would exist such that $\left[\begin{smallmatrix} z_1 \\ x_0 \end{smallmatrix}\right]$ satisfies the constraints in (3) and has a lower cost. Hence, $\mathcal{I}_0$ identifies an optimal combination for $x = x_0$. ■

A possible choice for B, $\bar{x}$ in (32) is $B = Y$, in case matrix Y in (3) is symmetric and positive semidefinite (which is always the case for mpQP problems deriving from (1), (9)), and $\bar{x}$ as the Chebychev center of X (which in case X is a box is simply the semi-sum of the upper and lower bounds defining X).

Once a starting optimal combination $\mathcal{I}_0$ has been determined, the corresponding critical region $CR_{\mathcal{I}_0}$ can be determined in accordance with Lemma 6 or Theorem 6, followed by the removal of redundant inequalities, as described in Section III-B.

In order to explore the parameter space outside $CR_{\mathcal{I}_0}$, we recall the following theorem (Theorem 2 of [10]):

Theorem 7 (Neighboring regions): Consider problem (3) with $Q = Q' > 0$, let $\mathcal{I}$ an optimal combination of constraints and $CR_{\mathcal{I}} = \{x \in \mathbb{R}^m : E^{\mathcal{I}} \leq e^{\mathcal{I}}\}$ the corresponding critical region in minimal H-representation. Assume no constraint is weakly active at optimality for all $x \in CR_{\mathcal{I}}$, i.e., $H^{-1}_{\mathcal{I}\mathcal{I}}[D_{\mathcal{I}} \ d_{\mathcal{I}}] \neq 0$ in (25b). Let $CR_{\mathcal{I}_j}$ be a full-dimensional critical region neighboring $CR_{\mathcal{I}}$, let $\tilde{P}_j = \{x \in \mathbb{R}^m : E^{\mathcal{I}}_j x = e^{\mathcal{I}}_j\} = CR_{\mathcal{I}} \cap CR_{\mathcal{I}_j}$ the common facet between $CR_{\mathcal{I}}$ and $CR_{\mathcal{I}_j}$, and assume that the LICQ property is satisfied on $\tilde{P}_j$. Then the optimal combination $\mathcal{I}_j$ associated with $CR_{\mathcal{I}_j}$ is related to $\mathcal{I}$ as follows:

$I)$ If $\tilde{P}_j$ corresponds to a constraint of type $w_k(x) \geq 0$ in (25), then $\mathcal{I}_j = \mathcal{I} \cup \{k\}$;

$II)$ If $\tilde{P}_j$ corresponds to a constraint of type $y_h(x) \geq 0$ in (25), then $\mathcal{I}_j = \mathcal{I} \setminus \{h\}$.

Remark 5: When the hypotheses of Theorem 7 are not met, the most efficient methods proposed in the literature to identify neighboring critical regions exploit the so-called "facet-to-facet" property of the multiparametric solution [11], [26]. This approach is based on the following idea. For each full-dimensional facet $\tilde{P}_j = \{x \in \mathbb{R}^n : E^{\mathcal{I}}_j x = e^{\mathcal{I}}_j\}$ of $CR_{\mathcal{I}}$, we determine the Chebychev center x_j of $\tilde{P}_j$ as described in Remark 2, set $\bar{x}_j = x_j + \epsilon E^{\mathcal{I}}_j$, where $\epsilon > 0$ is a small tolerance, and then solve the QP (3) for $x = \bar{x}_j$ in order to determine a new combination $\mathcal{I}_j$ of active constraints. Ways to cope with the (very rare) case of lack of the facet-to-facet property were proposed in [11, Sect. IV] and in [12, Algo. 3]. An alternative method that avoids dealing with this issue is the one proposed in [13], based on looking at (23) as a multiparametric linear complimentarity problem.

Based on the results of the previous sections, the overall mpQP algorithm is formalized in Algorithm 3. At line 6 of Algorithm 3 the way $\mathcal{I}$ is picked from $\mathcal{U}$ determines the exploration strategy: selecting the most recently introduced element corresponds to depth first, selecting the oldest element corresponds to breadth first. With a slight abuse of notation, in the sequel we will denote by $\{K^i, \ h^i, E^i, e^i\}$ the solution corresponding to the i-th combination $\mathcal{I}_i$ contained in $\mathcal{R} = \{\mathcal{I}_1, \ldots, \mathcal{I}_M\}$, where $M = \#\mathcal{R}$ is the cardinality of $\mathcal{R}$.

V. Complexity of the Solution

The complexity of the solution is given by the number M of critical regions and gains that form the explicit solution (8), dictating the amount of memory to store the parametric

Algorithm 3 Multiparametric QP algorithm

Input: Q, F, c, G, W, S, X.

1) **set** $\mathcal{E} \leftarrow \emptyset$, $\mathcal{U} \leftarrow \emptyset$, $\mathcal{R} \leftarrow \emptyset$;
2) **remove** redundant inequalities in the set $\{[\begin{smallmatrix} z \\ x \end{smallmatrix}] : Gz \leq W + Sx,\ x \in X\}$ (Section III-B);
3) **solve** (32); if infeasible, $X_f \cap X = \emptyset$; **go to** step 18;
4) **compute** $\mathcal{I}_0$ (Lemma 8); **set** $\mathcal{U} \leftarrow \{\mathcal{I}_0\}$;
5) **while** $\mathcal{U} \neq \emptyset$ **do**:
6) **get** an element $\mathcal{I} \in \mathcal{U}$; **set** $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathcal{I}\}$;
7) **if** $\mathcal{I} \notin \mathcal{E}$:
8) **set** $\mathcal{E} \leftarrow \mathcal{E} \cup \{\mathcal{I}\}$;
9) **compute** $CR_{\mathcal{I}} \cap X$ (Lemma 6 or Theorem 6);
10) **if** $CR_{\mathcal{I}} \cap X$ is full-dimensional (Section III-C) **or** $\mathcal{I} = \mathcal{I}_0$:
11) **remove** redundant inequalities in $CR_{\mathcal{I}} \cap X$ (Section III-B);
12) **for** all facets f_j of $CR_{\mathcal{I}}$:
13) **find** the neighboring combination $\mathcal{I}_j$ (Theorem 7 or Remark 5); **if** $\mathcal{I}_j$ exists, **set** $\mathcal{U} \leftarrow \mathcal{U} \cup \{\mathcal{I}_j\}$;
14) **if** $CR_{\mathcal{I}} \cap X$ is full-dimensional:
15) **store** $CR_{\mathcal{I}} \cap X = \{x \in \mathbb{R}^m : E^{\mathcal{I}}x \leq e^{\mathcal{I}}\}$
16) **set** $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{I}\}$;
17) **compute** $z(x) = K^{\mathcal{I}}x + h^{\mathcal{I}}$ as in (24) and (25b), or as in (27);
18) **end**.

Output: Multiparametric solution $\{K^{\mathcal{I}},\ h^{\mathcal{I}}, E^{\mathcal{I}}, e^{\mathcal{I}}\}_{\mathcal{I} \in \mathcal{R}}$.

solution (K^i, h^i, E^i, e^i, $i = 1, \ldots, M$), and by the worst-case execution time required to compute $K^i x + h^i$ once the so-called "point-location" problem of identifying the index i of the critical region containing the current state x is solved (which usually takes most of the time). A few methods have been proposed to solve the point-location problem more efficiently than searching linearly through the list of regions, see for example the methods proposed in [25], [27]–[29].

An upper-bound to M is 2^q, which is the number of all possible combinations of active constraints. In practice M is much smaller than 2^q, as most combinations are never active at optimality for any vector $x \in X$ (for example, lower and upper limits on an actuation signal cannot be active at the same time, unless they coincide). Moreover, one can attempt joining regions in which the first n_u component of the multiparametric solution $z(x)$ are the same (see Section V-B below). Nonetheless, the complexity of the explicit MPC law typically grows exponentially with the number q of constraints. The number m of parameters is less critical and mainly affects the number of elements to be stored in memory. The number n of free variables also affects the number M of regions, mainly because they are usually upper and lower bounded.

A. Robust implementation of the PWA law

The explicit solution (8) can be evaluated as follows:

$$i(x) \in \arg\min_{i=1,\ldots,M} \left\{ \max_{j=1\ldots,n_i} \left\{E^i_j x - e^i_j\right\} \right\} \quad (33a)$$

$$u(x) = K^{i(x)}x + h^{i(x)}, \quad (33b)$$

where $n_i = \dim(E^i)$ is the number of inequalities defining the i-th critical region $CR_{\mathcal{I}_i}$. The advantage of (33) over (8) is that it allows defining a solution $u(x)$ also in case x does not belong to any of the regions. This is particularly useful for two reasons: First, to avoid that the solution is undefined because of gaps introduced by numerical errors in defining the hyperplanes $E^i x = e^i$; second, to compute suboptimal explicit control laws by extrapolation, as suggested in the following section. In fact, $u(x) = u^*(x)$ in case $x \in CR_{\mathcal{I}_i}$ for some i, otherwise (33) extrapolates the affine law $K^{i(x)}x + h^{i(x)}$ outside region $CR_{\mathcal{I}_{i(x)}}$, where $i(x)$ identifies the region whose constraints are least violated by vector x. Clearly, in this case the satisfaction of constraints (1c)–(1d) is no longer guaranteed.

Lemma 9: Given a set of quadruples $\{K^u, h^i, E^i, e^i\}_{i=1}^M$, $e^i \in \mathbb{R}^{n_i}$, covering a polytope $X \subset \mathbb{R}^n$, the control law (33) is a (possibly discontinuous) PWA function.

Proof: Let $\delta(x) : \mathbb{R}^m \to \{0,1\}^M$ be a solution of the multiparametric MILP problem

$$\begin{array}{ll} \min_{\epsilon,\delta,z} & \sum_{i=1}^M z_i \\ \text{s.t.} & \epsilon_i \geq E^i_j x - e^i_j,\ \forall j = 1, \ldots, n_i \\ & \epsilon_i \leq \epsilon_j + 2\mathcal{M}(1-\delta_i),\ \forall j = 1, \ldots, n_i, j \neq i \\ & z_i \leq \epsilon_i + \mathcal{M}(1-\delta_i) \\ & z_i \geq \epsilon_i - \mathcal{M}(1-\delta_i) \\ & z_i \leq \mathcal{M}\delta_i \\ & z_i \geq -\mathcal{M}\delta_i \\ & \quad \forall i = 1, \ldots, M \\ & \sum_{i=1}^M \delta_i = 1 \end{array} \quad (34)$$

for $x \in X$, where $\mathcal{M}$ is an upper-bound on $\pm(E^i_j x - e^i_j)$ for all $x \in X$, $i \in \{1, \ldots, M\}$, $j \in \{1, \ldots, n_i\}$. Problem (34) determines a solution $\delta_{i(x)}(x) = 1$, $\delta_j(x) = 0$, $\forall j \in \{1, \ldots, M\} \setminus \{i(x)\}$, such that the index $i(x)$ solves (33a). As the solution $\delta(x)$ of (34) is a piecewise constant function of x [30], the scalar product

$$u(x) = \begin{bmatrix} K^1 x + h^1 \\ \vdots \\ K^M x + h^M \end{bmatrix}' \begin{bmatrix} \delta_1(x) \\ \vdots \\ \delta_M(x) \end{bmatrix}$$

of an affine and a piecewise constant function is a piecewise affine function of x. ∎

B. Explicit MPC with a reduced number of regions

A first approach to reduce the number M of regions that maintains the solution exact is to attempt joining critical regions in which the first n_u components $K^i x + h^i$ of the multiparametric solution $z(x)$ are the same, either by direct recognition of convexity and computation of unions of pairs of sets [23, Algorithm 4.1], or by optimal merging algorithms [31] to get a minimal number M of partitions.

Suboptimal solutions can be obtained by removing critical regions $CR_{\mathcal{I}}$ that are almost flat, such as regions whose Chebychev radius is smaller than a given positive threshold, that can be recognized as described in Remark 1.

In alternative, or in addition, suboptimal solutions can be obtained by partial enumeration as suggested in [14, Sect. 3.1]

TABLE I
COMPUTATION OF MINIMAL H-REPRESENTATIONS. CPU TIME (S) ON RANDOM POLYHEDRA IN $\mathbb{R}^m$ DEFINED BY $10m$ INEQUALITIES (RESULTS AVERAGED ON 20 NONEMPTY POLYHEDRA PER VALUE OF m)

m	Th. 2	Th. 3	Th. 4	LP (35)
2	0.0051	0.0035	0.0006	0.0046
4	0.0351	0.0301	0.0019	0.0103
6	0.1297	0.1233	0.0038	0.0193
8	0.4046	0.3976	0.0071	0.0340
10	1.0185	1.0082	0.0111	0.0554
12	2.0665	2.1030	0.0178	0.0955
14	4.5279	4.5546	0.0263	0.1426
16	7.9159	7.9133	0.0357	0.1959

and [32]. The idea is to run closed-loop MPC simulations based on on-line QP and only store the critical regions $CR_\mathcal{I}$ corresponding to the most visited combinations $\mathcal{I}$. In any case, by Lemma 9 the corresponding (suboptimal) explicit MPC law defined by (33) would be a PWA control law.

VI. NUMERICAL RESULTS

A. Removal of redundant constraints

As most of the CPU time is spent in removal of redundant constraints to get minimal H-representations of critical regions $CR_\mathcal{I}$, we first compare the approaches of Theorems 2, 3, 4 based on NLLS and the common approach of labelling an inequality constraint $E_i x \leq e_i$ of a polyhedron $P = \{x \in \mathbb{R}^m : \ Ex \leq e\}$ as non-redundant if the LP problem

$$\begin{array}{rcll} \epsilon_i & = & \max_x & E_i x - e_i \\ & & \text{s.t.} & E_j x \leq e_j, \ j \neq i \end{array} \tag{35}$$

returns $\epsilon_i > 0$. The LP (35) is solved by GLPK [33], while NNLS problems are coded in Embedded MATLAB and compiled. The results are shown in Table I on random nonempty polyhedra in $\mathbb{R}^m$ defined by $10m$ inequalities (CPU time is averaged on 20 polyhedra per value of m). It is apparent that the method of Theorem 4 outperforms the other methods. The basic Algorithm 1 is used to solve all NNLS problems, by solving Step 5 via MATLAB's built-in QR factorization method to solve linear systems in the least-squares sense and without any attempt to warm-starting the algorithm from $\mathcal{P} \neq \emptyset$.

Remark 6: In general, possible sequences of non-improving steps may occur in active-set methods for linearly constrained optimization, including the simplex method used for LP, as analyzed in [34]. In [19] the authors prove that the norm of $\|Av - b\|$ of the residual is strictly decreasing during the iterations of Algorithm 1, so that cycling cannot occur in infinite precision. Although none of the extensive tests (run in double precision arithmetics) revealed such an issue, cycling remains a theoretical possibility and anti-cycling procedures were proposed in the literature, see e.g. [34] and the references therein included, and [35], [36]. In particular, for the NNLS algorithm a few ideas to deal with finite precision arithmetics are described in [19, p. 164–165].

TABLE II
COMPARISON OF MPQP ALGORITHMS. CPU TIME (S) IS AVERAGED ON 20 RANDOM MPQP PROBLEMS FOR EACH PAIR (q, m)

q	m	Hybrid Tbx. [37]	MPT [38]	mpQP NNLS
4	2	0.0174	0.0256	0.0026
4	3	0.0263	0.0356	0.0038
4	4	0.0432	0.0559	0.0061
4	5	0.0650	0.0850	0.0097
4	6	0.0827	0.1105	0.0126
8	2	0.0347	0.0396	0.0050
8	3	0.0583	0.0680	0.0092
8	4	0.0916	0.0999	0.0140
8	5	0.1869	0.2147	0.0322
8	6	0.3177	0.3611	0.0586
12	2	0.0398	0.0387	0.0054
12	3	0.1121	0.1158	0.0191
12	4	0.2067	0.2001	0.0352
12	5	0.6180	0.6428	0.1151
12	6	1.2453	1.3601	0.2426
20	2	0.1029	0.0763	0.0152
20	3	0.3698	0.2905	0.0588
20	4	0.9069	0.7100	0.1617
20	5	2.2978	1.9761	0.4395
20	6	6.1220	6.2518	1.2853

B. Performance of the mpQP solver

We compare Algorithm 3 with the mpQP solvers of the Hybrid Toolbox [37] and of the Multi Parametric Toolbox [38]. Algorithm 3 is implemented in MATLAB code, and all polyhedral computation functions, the NNLS Algorithm 1, and the QP algorithm [5] in compiled Embedded MATLAB code. The approach of Theorem 4 is used for computing minimal H-representations of polyhedra. Results are obtained on a Macbook Pro 2.6 GHz Inter Core i5 with 8 Gb RAM running MATLAB R2014a.

Table II reports the results obtained by solving random mpQP problems generated for different values of q (number of constraints), m (dimension of parameter vector), and $n = 2m$ (number of optimization variables). CPU time is averaged on 20 random mpQP's for each pair (q, m). Clearly, the results obtained with the tools of [37] and [38] strongly depend on the linear and quadratic programming solvers used to solve the multiparametric problem, here they were both configured with default options.

Finally, we test the mpQP algorithm on the DC servo example of [39]. With reference to the MPC formulation (9) we have $\mathcal{A} \in \mathbb{R}^{4\times 4}$, $\mathcal{B}_u \in \mathbb{R}^4$, $\mathcal{B}_v = \emptyset$, $\mathcal{C} \in \mathbb{R}^{2\times 4}$, $\mathcal{D}_u \in \mathbb{R}^2$, $\mathcal{D}_v = \emptyset$, $u_{\min} = -220$, $u_{\max} = 220$, $\Delta u_{\min} = -\infty$, $\Delta u_{\max} = +\infty$, $y_{\min} = \left[\begin{smallmatrix} -\infty \\ -78.54 \end{smallmatrix}\right]$, $y_{\max} = \left[\begin{smallmatrix} +\infty \\ 78.54 \end{smallmatrix}\right]$, $V_{\min} = V_{\max} = \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$, $N = N_c = 7$, $N_u = 2$, $\rho_\epsilon = 1000$, $Q_y = \left[\begin{smallmatrix} 100 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, $R_{\Delta u} = 0.0025$, $R_u = 0$. The resulting mpQP problem has $m = 6$ parameters (4 states, 1 reference for y_1, 1 previous input), $n = 3$ variables, $q = 18$ constraints, and is solved for $X = \{x \in \mathbb{R}^6 : \ -1000 \leq x_{1-4} \leq 1000, \ -5 \leq x_5 \leq 5, \ -221 \leq x_6 \leq 221\}$ in 5.3296 s using Algorithm 3 (it takes 6.0470 s with the Hybrid Toolbox [37] and 6.2476 s with the Multi-Parametric Toolbox [38]). A section in the (x_1, x_2)-space of the six-dimensional partition, which consists of 215 regions, is reported in Figure 1.

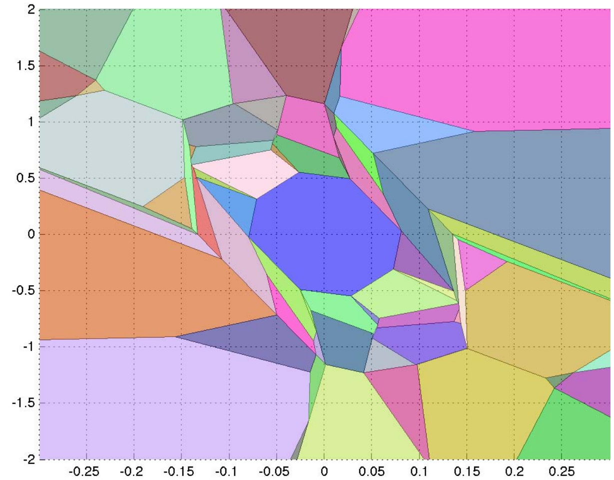

Fig. 1. DC motor example – Section of the partition for $x_{3-6} = 0$

VII. Conclusions

In this paper we have proposed a new mpQP algorithm to convert an MPC design into an equivalent explicit form. Contrarily to other methods proposed in the literature, the algorithm relies on a NNLS solver to perform all the operations on polyhedra that are necessary to solve the mpQP problem. In addition, the algorithm exploits the multiparametric dual QP to create critical regions and handle degeneracy. The overall mpQP method described in this paper is the core engine for explicit MPC design in the Model Predictive Control Toolbox for MATLAB 5.0 (The Mathworks, Inc.) [40].

References

[1] D. Mayne and J. Rawlings, *Model Predictive Control: Theory and Design*. Madison,WI: Nob Hill Publishing, LCC, 2009.

[2] A. Bemporad, "Model-based predictive control design: New trends and tools," in *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, 2006, pp. 6678–6683.

[3] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2011, in press (version updated: Mar 7, 2014).

[4] N. Ricker, "Use of quadratic programming for constrained internal model control," *Ind. Eng. Chem. Process Des. Dev.*, vol. 24, no. 4, pp. 925–936, 1985.

[5] C. Schmid and L. Biegler, "Quadratic programming methods for reduced Hessian SQP," *Computers & Chemical Engineering*, vol. 18, no. 9, pp. 817–832, 1994.

[6] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Contr. Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[8] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Trans. Automatic Control*, vol. 59, no. 1, pp. 18–33, 2014.

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

[10] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, Mar. 2003.

[11] J. Spjøtvold, E. Kerrigan, C. Jones, P. Tøndel, and T. A. Johansen, "On the facet-to-facet property of solutions to convex parametric quadratic programs," *Automatica*, vol. 42, no. 12, pp. 2209–2214, 2006.

[12] P. Patrinos and H. Sarimveis, "A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings," *Automatica*, vol. 46, no. 9, pp. 1405–1418, 2010.

[13] C. Jones and M. Morari, "Multiparametric linear complementarity problems," in *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, 2006, pp. 5687–5692.

[14] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear Model Predictive Control: Towards New Challenging Applications*, ser. Lecture Notes in Control and Information Sciences, D. R. L. Magni, F. Allgower, Ed., vol. 384. Berlin Heidelberg: Springer-Verlag, 2009, pp. 345–369.

[15] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming — The explicit solution," *IEEE Trans. Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[16] ——, "Min-max control of constrained uncertain discrete-time linear systems," *IEEE Trans. Automatic Control*, vol. 48, no. 9, pp. 1600–1606, 2003.

[17] ——, "Piecewise linear optimal controllers for hybrid systems," in *Proc. American Contr. Conf.*, Chicago, IL, June 2000, pp. 1190–1194.

[18] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, no. 10, pp. 1709–1721, Oct. 2005.

[19] C. Lawson and R. Hanson, *Solving least squares problems*. SIAM, 1974, vol. 161.

[20] A. Bemporad, "A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control," 2014, submitted for publication.

[21] A. Bemporad and M. Paggi, "Optimization algorithms for the solution of the frictionless normal contact between rough surfaces," 2014, submitted for publication.

[22] R. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[23] A. Bemporad, K. Fukuda, and F. Torrisi, "Convexity recognition of the union of polyhedra," *Computational Geometry: Theory and Applications*, vol. 18, pp. 141–154, 2001.

[24] T. Huynh, C. Lassez, and J.-L. Lassez, "Practical issues on the projection of polyhedral sets," *Annals of mathematics and artificial intelligence*, vol. 6, no. 4, pp. 295–315, 1992.

[25] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, May 2003.

[26] M. Baotić, "An efficient algorithm for multi-parametric quadratic programming," Automatic Control Institute, ETH, Zurich, Switzerland, Tech. Rep. AUT02-05, 2002.

[27] F. Christophersen, M. Kvasnica, C. Jones, and M. Morari, "Efficient evaluation of piecewise control laws defined over a large number of polyhedra," in *Proc. European Control Conf.*, Kos, Greece, 2007, pp. 2360–2367.

[28] Y. Wang, C. Jones, and J. Maciejowski, "Efficient point location via subdivision walking with application to explicit MPC," in *Proc. European Control Conf.*, Kos, Greece, 2007, pp. 447–453.

[29] M. Herceg, S. Mariethoz, and M. Morari, "Evaluation of piecewise affine control law via graph traversal," in *Proc. European Control Conf.*, Zürich, Switzerland, 2013, pp. 3083–3088.

[30] V. Dua and E. Pistikopoulos, "An algorithm for the solution of multiparametric mixed integer linear programming problems," *Annals of Operations Research*, vol. 1, pp. 123–139, Dec. 2000.

[31] T. Geyer, F. Torrisi, and M. Morari, "Optimal complexity reduction of polyhedral piecewise affine systems," *Automatica*, vol. 44, pp. 1728–1740, 2008.

[32] G. Pannocchia, J. Rawlings, and S. Wright, "Fast, large-scale model predictive control by partial enumeration," *Automatica*, vol. 43, pp. 852–860, 2007.

[33] A. Makhorin, *GLPK (GNU Linear Programming Kit) User's Guide*, 2004. [Online]. Available: http://www.gnu.org/software/glpk/glpk.html

[34] P. Gill, W. Murray, M. Saunders, and M. Wright, "A practical anticycling procedure for linearly constrained optimization," *Mathematical Programming*, vol. 45, pp. 437–474, 1989.

[35] R. Fletcher, "Degeneracy in the presence of roundoff errors," *Linear Algebra and its Applications*, vol. 106, pp. 149–183, 1988.

[36] P. Stark and R. Parker, "Bounded-variable least-squares: An algorithm and applications," *Computational Statistics*, vol. 10, pp. 129–141, 1995.

[37] A. Bemporad, *Hybrid Toolbox – User's Guide*, Dec. 2003, http://cse.lab.imtlucca.it/∼bemporad/hybrid/toolbox.

[38] M. Kvasnica, P. Grieder, and M. Baotić, *Multi Parametric Toolbox (MPT)*, 2006, http://control.ee.ethz.ch/∼mpt/.

[39] A. Bemporad and E. Mosca, "Fulfilling hard constraints in uncertain linear systems by reference managing," *Automatica*, vol. 34, no. 4, pp. 451–461, 1998.

[40] A. Bemporad, M. Morari, and N. Ricker, *Model Predictive Control Toolbox for MATLAB 5.0*. The Mathworks, Inc., 2014, http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/.

Alberto Bemporad Alberto Bemporad received his master's degree in Electrical Engineering in 1993 and his Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. He spent the 1996/97 academic year at the Center for Robotics and Automation, Department of Systems Science & Mathematics, Washington University, St. Louis, as a visiting researcher. In 1997-1999 he held a postdoctoral position at the Automatic Control Laboratory, ETH Zurich, Switzerland, where he collaborated as a senior researcher in 2000-2002. In 1999-2009 he was with the Department of Information Engineering of the University of Siena, Italy, becoming an associate professor in 2005. In 2010-2011 he was with the Department of Mechanical and Structural Engineering of the University of Trento, Italy. In 2011 he joined the IMT Institute for Advanced Studies Lucca, Italy as a full professor, where he also became the director in 2012. In 2011 he cofounded ODYS S.r.l., a spinoff company of IMT Lucca. He has published around 300 papers in the areas of model predictive control, hybrid systems, automotive control, multiparametric optimization, computational geometry, robotics, and finance. He is author or coauthor of various MATLAB toolboxes for model predictive control design, including the Model Predictive Control Toolbox (The Mathworks, Inc.). He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001-2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society in 2002-2010. He received the IFAC High-Impact Paper Award for the 2011-14 triennial. He has been an IEEE Fellow since 2010.