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 and fast to execute. In addition, the new approach exploits QP duality to identify and construct critical regions and to handle degeneracy issues.

Index Terms—Model predictive control (MPC), multiparametric programming, nonnegative least squares (NNLS), quadratic programming (QP).

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 model of the controlled system and quadratic penalties on tracking errors and actuation efforts; in this case, time-invariant model of the controlled system and quadratic programming (QP).

In contrast, explicit MPC was proposed in [14], [15] to replace the on-line QP solution by an off-line precomputed piecewise affine function of the state and control variables (inputs, outputs, and disturbances). In its off-line version, the control law is precomputed off-line as a piecewise affine (PWA) function of 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], alternating 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 $\infty$-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. The approach was further extended in [21] to solve mixed-integer quadratic programs.

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.

Manuscript received April 18, 2014; revised October 9, 2014; accepted February 26, 2015. Date of publication March 30, 2015; date of current version October 26, 2015. Recommended by Associate Editor J. Imura.

The author is with the IMT Institute for Advanced Studies, 55100 Lucca, Italy (e-mail: alberto.bemporad@imtilucca.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2015.2417851
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 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.

**Notation:** Let \( \mathbb{R}^n \) denote the set of real vectors of dimension \( n \) and the set of natural numbers, respectively. Let \( I \subset \mathbb{N} \) be a finite set of indices and denote by \( \#I \) its cardinality. For a vector \( a \in \mathbb{R}^n \), \( a_i \) denotes the \( i \)th entry of \( a \), and \( a \) the subvector obtained by collecting the entries \( a_i \) for all \( i \in I \). The Euclidean norm of \( a \), the condition \( a \) is equivalent to \( a_i > 0 \), \( \forall i \in 1, \ldots, n \) (and similarly for \( a \), \( \leq \), \( < \) inequalities), \( \|
abla f(a)\| \) is the diagonal matrix whose \( i\), \( j \)th element is \( a_i \), and \( B(a, c) = \{ x \in \mathbb{R}^n : \|x - a\|_2 \leq c \} \) the Euclidean ball centered in \( a \) of radius \( c \). Given two vectors \( a, b \in \mathbb{R}^n \), \( a \perp b \) means the complementarity condition \( a_i b_i = 0 \). For a matrix \( A \in \mathbb{R}^{n \times m} \), \( A^\top \) denotes its transpose, \( A \) the submatrix of \( A \) obtained by collecting the rows \( A_i \) for all \( i \in I \), \( A \) the submatrix of \( A \) obtained by collecting the rows and columns of \( A \) indexed by \( i \in I \) and \( j \in J \), respectively, \( A \) the vector whose \( i \)th entry is \( (A_i)_{j} \), and \( A_{#} \in \mathbb{R}^{m \times n} \) a pseudoinverse matrix of \( A \), namely \( A A_{#} A = A \), \( (A A_{#})_{#} = A_{#} \), \( A = (A_{#} A)^{#} \) (if \( A \) is full column rank, \( A_{#} = (A^{\top} A)^{-1} A^{\top} \)). 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 \( >, \leq \), \( < \) denote positive semidefiniteness, negative definiteness, negative semidefiniteness, respectively). \( I_{kn} \) 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:

\[
V^*(x) = \min_{z} \ell_N(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \tag{1a}
\]

**s.t.** \( x_{k+1} = Ax_k + Bu_k \) \( \tag{1b} \)

\[
C_x x_k + C_u u_k \leq c_0 \tag{1c}
\]

\[ k = 0, \ldots, N - 1 \]

\[
C_N x_N \leq c_N \tag{1d}
\]

\[ x_0 = 0 \tag{1e} \]

where \( N \) is the prediction horizon, \( u_k \in \mathbb{R}^{m} \) 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 \( A \in \mathbb{R}^{m \times m} \) and \( B \in \mathbb{R}^{m \times n} \).

\[
\ell(x, u) = \frac{1}{2} x' Q_x x + u'R_u u \tag{2a}
\]

\[
\ell_N(x) = \frac{1}{2} x' P_x x \tag{2b}
\]

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 \( C_x \in \mathbb{R}^{m \times n_c}, C_u \in \mathbb{R}^{m \times n_u} \), \( c_0 \in \mathbb{R}^{n_u} \), let \( n_N \) be the number of terminal constraints, namely \( C_N \in \mathbb{R}^{m \times n_N}, c_N \in \mathbb{R}^{m \times n_N} \), and let \( n_q \triangleq n_n + n_N \) 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)

\[
V^*(x) = \min_{z} \frac{1}{2} z' Q z + (F x + c)' z + \frac{1}{2} x' Y x \tag{3a}
\]

**s.t.** \( G z \leq W + S x \tag{3b} \)

where \( z \triangleq [u_0 \ldots u_{N-1}'] \in \mathbb{R}^{n} \), \( n \triangleq n_u N \), is the optimization vector, \( Q = Q' \in \mathbb{R}^{n \times n} \) 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 \( G \in \mathbb{R}^{n \times n}, S \in \mathbb{R}^{n \times m}, W \in \mathbb{R}^{m} \) define the constraints imposed in (1) in a compact form. Because of the assumptions made on the weights \( Q_x, R_x, P_x, P_x' \), matrix \( Q > 0 \) and \( F Y' \geq 0 \). 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)

\[
\Phi^*(x) = \min_{y} \frac{1}{2} y' H y + (D x + d)' y \tag{5a}\]

**s.t.** \( y \geq 0 \tag{5b}\)

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

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

**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

\[
X \triangleq \{ x \in \mathbb{R}^m : E x \leq e_x \} \tag{7}\]

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]:

\[
\text{BEPMORAD: MPQP ALGORITHM WITH POLYHEDRAL COMPUTATIONS BASED ON NNLS}
\]

2893
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 $z^* : X_f \rightarrow \mathbb{R}^n$ is piecewise affine and continuous over $X_f$;

iii) If in addition matrix $\begin{bmatrix} Q & E' \\ F & Y \end{bmatrix}$ is symmetric and positive semidefinite, the value function $V^* : X_f \rightarrow \mathbb{R}$ 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

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

An example of such a partition is reported in Fig. 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) online 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:

$$\min_z \left\{ \sum_{k=0}^{N-1} \frac{1}{2} (y_k - r_k)'Q_y(y_k - r_k) + \frac{1}{2} \Delta u_k' R \Delta u_k \\ + (u_k - u^*_k)' R_u (u_k - u^*_k)' + \rho \epsilon^2 \right\} \quad (9a)$$

s.t. $x_{k+1} = Ax_k + Bu_k + By_k$ \quad (9b)

$y_k = Cx_k + Du_k + Dy_k$ \quad (9c)

$u_k = u_{k-1} + \Delta u_k, \ k = 0, \ldots, N - 1$ \quad (9d)

$u_{-1} = u_1$ \quad (9e)

$\Delta u_k = 0, \ k = N_u, \ldots, N - 1$ \quad (9f)

$u^k_{\min} \leq u_k \leq u^k_{\max}, \ k = 0, \ldots, N_u - 1$ \quad (9g)

$V^k_{\min} - \epsilon V^k_{\min} \leq y_k \leq V^k_{\max} + \epsilon V^k_{\max}$ \quad (9h)

where $R_{\Delta u} = R_{\Delta u} > 0$, $Q_y = Q_y' \geq 0$, $R_u = R_u \geq 0$, $x_0$ is the current state, $v_k$ is a vector of measured disturbances, $y_k \in \mathbb{R}^{n_y}$ is the output vector, $r_k \in \mathbb{R}^{n_y}$ its corresponding reference to be tracked, $\Delta v_k$ the vector of input increments, $u_{-1}$ is the command input applied during the previous sampling interval, $u^k_{\min}, u^k_{\max}, \Delta u^k_{\min}, \Delta u^k_{\max}$ are bounds, and $N_u, N_v, N_c$ are, respectively, the prediction, control, and constraint horizons. The extra variable $\epsilon$ is introduced to softly output constraints via the relaxation vectors $V^k_{\min}, V^k_{\max} > 0$ of $\mathbb{R}^{n_y}$ and penalized by the (usually large) weight $\rho \epsilon$ 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 ($r_k \equiv r_0, \forall k \in \{0, \ldots, N - 1\}$), no measured disturbance, fixed upper and lower bounds, the explicit solution is a continuous piecewise affine function of the parameter vector $[x_0' r_0' u_{-1}']'$. 

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

$$r^* = \min_r \| Av - b \|^2_2 \quad \text{s.t.} \quad v \geq 0, \quad (10)$$

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].
Algorithm 1 NNLS solver [19, p.161]

**Input:** Matrices $A$, $b$.

1) $\mathcal{P} \leftarrow \emptyset$, $v \leftarrow 0$;
2) $w \leftarrow A^t(Av - b)$;
3) if $w \geq 0$ or $\mathcal{P} = \{1, \ldots, m\}$ then go to Step 11;
4) $i \leftarrow \arg \min_{i \in \{1, \ldots, m\}} w_i$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{i\}$;
5) $y_p \leftarrow \arg \min_{y \in \mathcal{P}} \|\begin{pmatrix} (A^t)^t \end{pmatrix}y - b\|_2^2$, $y \in \{1, \ldots, m\}, p \leftarrow 0$;
6) if $y_p \geq 0$ then $v \leftarrow y$ and go to Step 2;
7) $j \leftarrow \arg \min_{i \notin \mathcal{P}}: w_i \leq 0 \left\{ \begin{pmatrix} \bar{v}_{i} \end{pmatrix} \right\}$;
8) $v \leftarrow v + \frac{1}{y_j} (y - v)$;
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 $\mathcal{P} \neq \emptyset$ of active constraints, see, e.g., [22, 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 LDLT 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

$$r^* = \min_{v} \|Av - b\|_2^2$$

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 (PNLNS) problem

$$r^* = \min_{v,u} \|Av + Bu - c\|_2^2$$

s.t. $v \geq 0$, $u$ free

$$= \min_{v,u} \|Av - \tilde{b}\|_2^2$$

s.t. $v \geq 0$, $u$ free

(12)

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

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

$$v^* \in \arg \min_{v} \|Av - \tilde{b}\|_2^2$$

s.t. $v \geq 0$

(13)

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

Proof: Let $(v^*, u^*)$ be a solution of the PNLNS problem (12), and let $u(v) \in \arg \min_{u} \|Av + Bu - c\|_2^2 = -B^#(Av - c)$ for some pseudoinverse $B^#$ of $B$. By definition $\|Av^* + Bu^* - c\|_2^2 \leq \|Av^* + Bu - c\|_2^2$ for all $u \in \mathbb{R}^p$, we have in particular $\|Av^* + Bu^* - c\|_2^2 \leq \|Av^* + Bu - c\|_2^2$, and hence $(v^*, u^*)$ is also an optimal solution of (12). Then, the search for optimal NNLS 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 PNLNS 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 PNLNS problem

$$(v^*, u^*) \in \arg \min_{v,u} \|Iv + Eu - e\|_2^2$$

s.t. $v \geq 0$, $u$ free

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 $Ev = e_i$ 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

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

Then $\bar{P}_i \neq \emptyset$ if and only if the PNLNS problem

$$(v^*, u^*) \in \arg \min_{v,u} \|C_i v + Eu - e\|_2^2$$

s.t. $v \geq 0$, $u$ free

(15)

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

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

$$\tilde{u} \triangleq \begin{bmatrix} e_1 - E_{11} \tilde{u} \\ \vdots \\ e_{i-1} - E_{i-1,1} \tilde{u} \\ e_{i+1} - E_{i+1,1} \tilde{u} \\ \vdots \\ e_m - E_{m,1} \tilde{u} \end{bmatrix} \geq 0$$

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

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

$$u^* \in \arg \min_{u} \|Eu + E_0\|_2^2$$

(16)

where $E_0 \in \mathbb{R}^m$ is a vector whose $i$th entry $(E_0)_i = \|E_i\|_2$. The cone $C$ is full-dimensional if and only if $Eu^* + E_0 = 0$. 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 \( Eu^* + E_i = 0 \) and consider a generic vector \( u \in B(u^*, 1) \), that is \( u = u^* + w \) for some \( w \in B(0, 1) \). Then \( E_i u = E_iu^* + E_iw = -\|E_i\|_2 + E_iw \leq -\|E_i\|_2 + \|E_i\|_2 = 0 \), \( \forall i = 1, \ldots , n \), 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 \( d = Eu^* < 0 \). Let \( u \in \mathbb{R}^n \) such that \( u_i = -(\|E_i\|_2/d_i)\bar{u}_i \). Then, \( (Eu + E_i)\bar{u} = -\|E_i\|_2 + \|E_i\|_2 = 0 \), that is \( Eu + E_i = 0 \), and hence \( \|Eu^* + E_i\|_2 = 0 \) 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 \( P = \{ u \in \mathbb{R}^n : E_i u \leq e_j , j = 1, \ldots , i-1 , i+1 , \ldots , m \} \).

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

\[
\begin{align*}
E^i = \begin{bmatrix} E_1 \\ \vdots \\ E_{i-1} \\ -E_i \\ E_{i+1} \\ \vdots \\ E_m \end{bmatrix}, & e^i = \begin{bmatrix} e_1 \\ \vdots \\ e_{i-1} \\ -e_i \\ e_{i+1} \\ \vdots \\ e_m \end{bmatrix} \\
P^i = \{ u \in \mathbb{R}^n : E^i u \leq e^i \}, & \tilde{P}^i = P \cap \{ u \in \mathbb{R}^n : E_i u = e_i \} \\
\end{align*}
\]

for \( i = 1, \ldots , m \). Clearly, \( \tilde{P}^i = P^i \cap \{ u \in \mathbb{R}^n : E_i u = e_i \} \) and \( P^i = P \cap P^i \).

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

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

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

\[
\begin{align*}
\min_{v,u} & \quad \| I + \begin{bmatrix} e^i & e^i \end{bmatrix} v - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}^2 \\
\text{s.t.} & \quad v \geq 0 , u \text{ free} \quad \text{(18)}
\end{align*}
\]

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

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

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

\[
\begin{align*}
\min_{v} & \quad \| E^i v - e^i \|_2^2 \\
\text{s.t.} & \quad v \geq 0 \quad \text{(19)}
\end{align*}
\]

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

Proof: By Lemma 5, strong-redundancy of the \( i \)th inequality \( E_i u \leq e_i \) is equivalent to emptiness of \( P^i = \{ u \in \mathbb{R}^n : E^i u \leq e^i \} \). 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 \( (E^i)' v = 0 \), \( v \geq 0 \), \( (e^i)' v = -1 \). By Park's Lemma [23, p. 201], this is equivalent to infeasibility of \( E^i u \leq e^i \). Vice versa, if \( E^i u \leq e^i \) is infeasible then a vector \( \bar{v} \) exists such that \( (E^i)' \bar{v} = 0 \), \( \bar{v} \geq 0 \), \( (e^i)' \bar{v} < 0 \). By defining \( v = -(1/(e^i)' \bar{v}) \bar{v} \), we get

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

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

<table>
<thead>
<tr>
<th>m</th>
<th>Th. 2</th>
<th>Th. 3</th>
<th>Th. 4</th>
<th>LP (35)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.0081</td>
<td>0.0039</td>
<td>0.0006</td>
<td>0.0046</td>
</tr>
<tr>
<td>4</td>
<td>0.0351</td>
<td>0.0301</td>
<td>0.0019</td>
<td>0.0013</td>
</tr>
<tr>
<td>6</td>
<td>0.1297</td>
<td>0.1233</td>
<td>0.0038</td>
<td>0.0019</td>
</tr>
<tr>
<td>8</td>
<td>0.4046</td>
<td>0.3976</td>
<td>0.0071</td>
<td>0.0034</td>
</tr>
<tr>
<td>10</td>
<td>1.0185</td>
<td>1.0082</td>
<td>0.0111</td>
<td>0.0055</td>
</tr>
<tr>
<td>12</td>
<td>2.0665</td>
<td>2.1030</td>
<td>0.0178</td>
<td>0.0095</td>
</tr>
<tr>
<td>14</td>
<td>4.5279</td>
<td>4.5546</td>
<td>0.0263</td>
<td>0.1426</td>
</tr>
<tr>
<td>16</td>
<td>7.9159</td>
<td>7.9133</td>
<td>0.0357</td>
<td>0.1959</td>
</tr>
</tbody>
</table>

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

Proof: If Problem (15) has a zero residual, by Lemma 3, $\bar{P} \neq \emptyset$, which implies $P^* \neq \emptyset$, and hence, by Lemma 5, the inequality $E_iu \leq e_i$ is not strongly redundant. Vice versa, if $E_iu \leq e_i$ is not strongly redundant then $P^* \neq \emptyset$. Take two arbitrary vectors $u_1 \in P^*, u_2 \in P \setminus P^*$. Clearly, $E_iu_h \leq e_i, \forall j \in \{1, \ldots, m\} \setminus \{i\}, h = 1, 2,$ and $E_iu_1 = e_i + v_1, E_iu_2 = e_i - v_2$ for some scalars $v_1, v_2 \geq 0$. Let $\alpha = \min\{v_2/(v_1 + v_2)\}$ and set $u = \alpha u_1 + (1 - \alpha)u_2$. Then $E_iu = \alpha E_iu_1 + (1 - \alpha)E_iu_2 = \alpha e_i + \alpha v_1 + (1 - \alpha)v_2 = \alpha e_i - \alpha v_2 + v_2 = e_i$, which implies $u \in \bar{P}$. Therefore, by Lemma 3, Problem (15) has a zero residual.\qed

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 $(m, m)$, 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 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 and 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 $P = \{u \in \mathbb{R}^n : Eu \leq e\}$, $E \in \mathbb{R}^{m \times n}$, $e \in \mathbb{R}^m$, be a nonempty polyhedron, let $u_0 \in P$, and let $\gamma_i = E_iu_0$, $i = 1, \ldots, m$. Let $I$ denote the set of indices $i$ such that $\gamma_i = 0$ (active inequalities at $u_0$). The following statements are true:

i) If $I = \emptyset$, then $P$ is full-dimensional. Moreover, $B(u, \epsilon) \subset P$ for all $\epsilon \in \mathbb{R}$ such that $0 < \epsilon \leq \min_{i=1,\ldots,m} |\gamma_i|/\|E_i\|_2$.

ii) If $I \neq \emptyset$, let $u^*$ be a solution of the LS problem (16) with $E$ replaced by $E_I$ and $E_1$ by $(E_I)_1$. Then $P$ is full-dimensional if and only if the residual $E_Iu^* + (E_I)_1 = 0$.

In this case, by defining

$$\alpha_\gamma = \min_{i \in I : E_iu > 0} \frac{\gamma_i}{\|E_i\|_2}$$

where $\alpha_\gamma = +\infty$ if $E_iu^* \leq 0$ for all $i = 1, \ldots, m$, and

$$\epsilon_\gamma = \min_{i \in \{1, \ldots, m\}} \frac{|e_i - E_iu_0 - \alpha_\gamma E_iu^*|}{\|E_i\|}$$

it holds that $\alpha_\gamma > 0, 0 < \epsilon_\gamma \leq \alpha_\gamma$, and $B(u_0 + \alpha_\gamma \epsilon_\gamma, \epsilon_\gamma) \subset P$ for all $\epsilon_\gamma \epsilon \gamma$ such that $0 < \epsilon \leq \epsilon_\gamma$.

Proof: Part (i): If $I = \emptyset$, then $E_iu < e_i$ and hence $\epsilon \geq \min_{i=1,\ldots,m} |\gamma_i|/\|E_i\|_2$. Moreover, for any $w \in B(u_0, \epsilon)$ we have $E_iw - e_i = -\gamma_i + E_i(w - u_0) \leq -\gamma_i + \epsilon\|E_i\|_2$. Hence, if $E_iw = 0$ then $E_iw - e_i \leq -\gamma_i$, while if $E_iw \neq 0$ then $E_iw - e_i \leq \|E_i\|_2(\gamma_i/\|E_i\|_2)$. This proves that $B(u_0, \epsilon) \subset P, \forall \epsilon \leq \epsilon_\gamma$.

Part (ii): Let $I \neq \emptyset$. Assume that $P$ is full-dimensional and by contradiction that the residual $E_Iu^* + (E_I)_1 \neq 0$. By Lemma 4 the cone $C = \{u \in \mathbb{R}^n : E_Iu < -u \leq 0\}$ is not full-dimensional, and therefore the set $C = \{u \in C \cap B(u_0, \epsilon) \}$ is not full-dimensional. Consider any $u \in P$. Then $E_Iu^* + (E_I)_1 \neq 0$, which implies $E_Iu < u \leq 0$ and hence $u - u_0 \in C$, 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 $E_Iu^* + (E_I)_1 = 0$. We want to prove that $P$ is full-dimensional. If no $i \notin I$ exists such that $E_iu^* > 0$ then $\gamma_i = +\infty > 0$. Otherwise, $\gamma_i > 0$ by definition, being the minimum of ratios of positive scalars.

Consider a generic $\alpha_\gamma 0 < \alpha < \alpha_\gamma$ and the quantity $\sigma_i = (\gamma_i - \alpha E_iu^*)/\|E_i\|_2$. For $i \in I, \sigma_i = \alpha_\gamma > 0$. For $i \notin I$, if $E_iu^* \leq 0$ then $\sigma_i = \gamma_i/\|E_i\|_2 > 0$; otherwise, if $E_iu^* > 0$, then $\sigma_i = (E_iu^* - \alpha E_iu^*)/\|E_i\|_2(\|E_iu^* - \alpha E_iu^*\|/\|E_i\|_2 (\sigma_i - \alpha_\gamma) > 0$.

Hence, $\epsilon_\gamma = \min_{i \in \{1, \ldots, m\}} \sigma_i > 0$ and moreover $\epsilon_\gamma = \min_{i \in \{1, \ldots, m\}} (\gamma_i - \alpha E_iu^*)/\|E_i\|_2 - \alpha E_iu^*/\|E_i\|_2 \geq 0$. Let $\epsilon < \epsilon_\gamma$, let $u_n = u_0 + \alpha \epsilon_\gamma$, and let $w \in B(u_0, \epsilon)$. For $i \in I$, for all $w \in B(u_0, \epsilon)$ we have $E_iw - e_i = E_iu_0 - e_i + \alpha E_iu^* - (E_iu - u_0) \leq -\gamma_i + \alpha E_iu^* + \epsilon\|E_i\|_2 \leq \|E_i\|_2(\gamma_i - \alpha E_iu^*)/\|E_i\|_2 + \epsilon) \leq 0$ for all $\epsilon \leq \epsilon_\gamma$. Hence, $B(u_0, \epsilon) \subset P$, for all $\epsilon \leq \epsilon_\gamma$, and for all $\alpha < \alpha_\gamma$, and therefore $P$ is full-dimensional.\qed

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 $B(u_0, \epsilon)$ included in $P$, that is the Chebychev center $u_c$ and radius $\epsilon_c$ of $P$, or if $P$ contains a ball of a given radius $\rho$ that is larger than the radius determined by Theorem 5. Algorithm 2 shows how to compute $u_c, \epsilon_c$ via NNLS and bisecion. At each iteration, the interval $[r_{\min}, r_{\max}]$ is halved, which ensures a good convergence of the algorithm.
Algorithm 2 Chebychev center and radius via PNNLS and bisection

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

1. \( r_{\min} \leftarrow \min_{e_i}(\|e_i - E_i u_0\|/\|E_i\|_2), r_{\max} \leftarrow +\infty, u_c \leftarrow u_0; \)
2. \textbf{while } \( r_{\max} - r_{\min} \geq \delta \)
3. \quad 2.1 \( r \leftarrow (r_{\min} + r_{\max})/2; \)
4. \quad 2.2 solve the PNNLS problem
5. \quad \( (v^*, u^*) \in \arg \min_{v,u} \|Iv + Eu - (e^i - r E_i)\|_2^2 \)
6. \quad \text{s.t.} \quad v \geq 0, u \text{ free}; \quad (22) \)
7. \quad 2.3 \text{if } \|v^* + Eu^* - (e^i - r E_i)\|_2 \leq \delta \text{ then } r_{\min} \leftarrow r,
8. \quad \else \( u_c \leftarrow u^* \text{ else } r_{\max} \leftarrow r; \) \)
9. \quad 3. \( c_e \leftarrow r_{\min}; \) \textbf{end.}

\textbf{Output: } \( B(u_c, c_e) = \) largest ball (within precision \( \delta \)) in \( P \).

Remark 1: By Lemma (2), the PNNLS problem (22) solved at Step 2.2 determines if the set of inequalities \( Eu + r E_i \leq e \) 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 \( \rho \), Algorithm 2 reduces to simply executing Step 2.2 for \( r = \rho \).

Remark 2: Algorithm 2 can be easily modified to 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 \( \|Eu + C_i v - (e^i - r E_i)\|_2^2 \), where \( E_i \) is obtained from \( E_i \) by zeroing its \( i \)th entry.

Remark 3: The PNNLS solver for (12) can be easily modified to solve the Linear Program (LP) \( \min \{ w : Gu \geq W \} \) \( f^T u, \)
\( G \in \mathbb{R}^{p \times n}, W \in \mathbb{R}^q, f \in \mathbb{R}^p, \) by considering the dual LP problem \( \max \{ y : G^T f - W^T y \} \) \( f^T u \), \( G \in \mathbb{R}^{q \times n}, W \in \mathbb{R}^q, f \in \mathbb{R}^q, \)\( G^T f - W^T y \) \( f^T u \) and imposing the optimality conditions \( G^T y = 0, G u + s = W, s \geq 0, y \geq 0, \) and the complementarity condition \( y^T (G^T W - W) = 0 \), where the latter is equivalent to the condition \( f^T u + W^T y = 0 \) (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 \( A = \begin{bmatrix} W^T & 0 \\ 0 & I \end{bmatrix}, B = \begin{bmatrix} f^T \\ G^T \end{bmatrix}, c = \begin{bmatrix} 0 \\ W \end{bmatrix} - f \) . Then, this approach can be used as an alternative to Algorithm 2 to finding the Chebychev radius of the polyhedron \( P \) via the LP \( \max \{ u : Eu \leq e \} \) \( e \).

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 \( 1/\|E_i\|_2 E_i \) and each component \( e_i \) by \( 1/\|E_i\|_2 e_i \). 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 [24, Algorithm 4.1] and to remove redundancy in Fourier-Motzkin elimination methods [25] 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 hence, 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 \( X \subset \mathbb{R}^m \) 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)
\[
\begin{align*}
\begin{cases}
  w &= Hy + Dx + d \\
  0 &\leq y \perp w \geq 0.
\end{cases}
\end{align*}
\]

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

Definition 5: A combination \( I \subseteq \{1, \ldots, q\} \) 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 \( y_I \geq 0 \) and \( w_I = 0 \).

Definition 6: Given an optimal combination \( I \subseteq \{1, \ldots, q\} \), the corresponding critical region \( CR_I \) 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 \( y_I \geq 0 \) and \( w_I = 0 \). When solving an mpQP problem, we are interested in collecting all full-dimensional critical regions \( CR_I \) (in minimal H-representation) such that \( CR_I \cap X \neq \emptyset \).

Given a vector \( x \in \mathbb{R}^n \), an optimal combination \( I \), and a pair \( (y,w) \) satisfying (23a), by (6) the corresponding optimal solution is
\[
\begin{align*}
z(x) &= -Q^{-1}(G^T y_I(x) + F x + c).
\end{align*}
\]

By (23a) and recalling the definitions of \( H, D, d \) in (5) we get that \( w(x) = G Q^{-1} G y_I(x) + (G Q^{-1} F + S)x + G Q^{-1} c + W = -G z + S x + W \) is the slack vector, so that \( w(x) \geq 0 \) is the condition of primal feasibility. Therefore, \( w_I = 0 \) is equivalent to \( G z = S x + W_I \), so that \( I \) determines an optimal active set for the primal QP problem (3).

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

Lemma 6 (Critical Region): Let \( I \subseteq \{1, \ldots, q\} \) be an optimal combination, and assume LICQ is satisfied for \( I \). Then \( I \) is optimal for all vectors \( x \) contained in the critical region \( CR_I \)
\[
CR_I = \{ x \in \mathbb{R}^m : y_I(x) \geq 0, w_I(x) \geq 0 \}
\]
where $J = \{1, \ldots, q\} \setminus I$, and

$$y_I(x) = -H_{II}^{-1}(D_Ix + d_I) \quad (25b)$$

$$w_J(x) = H_{JII}y_I(x) + D_Jx + d_J \quad (25c)$$

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

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

Note that in case $I = \emptyset$ 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 $I$, 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 $I \subseteq \{1, \ldots, q\}$ be an optimal combination, $n_I \triangleq \#I$, and let $r < n_I$ the rank of $G_I$ (LICQ condition violated). Let $G_IE_I = Q_IR_I$ be a QR decomposition of $G_I$, where $Q_I \in \mathbb{R}^{n_I \times n_I}$, $Q_I^TQ_I = I$, $Q_I = [q_1' \ q_2']$, $q_1 \in \mathbb{R}^{n_I \times n_I}$, $q_2 \in \mathbb{R}^{(n_I-r) \times n_I}$, $R_I \in \mathbb{R}^{n_I \times n_I}$ is such that $R_I = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$, 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 $E_I$ is a permutation matrix.

i) If $Q_2[D_Ix + d_I] \neq 0 \quad (26)$

then $CR_I$ is not a full-dimensional critical region.

ii) Let $M_I = -((R_1 R_2)[E_I^{-1}]^{-1}(E_I^{-1})'[R_1 R_2]')^{-1}Q_I$. Then any vector $y(x)$ such that $y_J(x) = 0$, $J = \{1, \ldots, q\} \setminus I$, and $y_I(x) = Q_I M_I[D_Ix + d_I]s_2$ is a dual solution of (5) for any $s_2 \in \mathbb{R}^{n_I-r}$ such that $y_I(x) > 0$, and $z(x) = -Q^{-1}(Fx + c + (E_I^{-1})'[R_1 R_2]M_I[D_Ix + d_I]) \quad (27)$

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

$$P = \left\{ x \in \mathbb{R}^{m+n_I-r} : Q_I \left[ M_I[D_Ix + d_I] \right] s_2 \geq 0 \right\}. \quad (28)$$

**Proof:** Let $\tilde{Q}_I = E_I^{-1}Q_I^{-1}(E_I^{-1})'$, so that $H_{II} = G_IE_IE_I^TQ_I^T\tilde{Q}_I$. By (23a), $u_I = 0$ leads to $H_{II}y_I + D_Ix + d_I = 0$, and hence, by setting $s_2 = Q_Iy_I = s_1 s_2$, $s_1 \in \mathbb{R}^r$, $s_2 \in \mathbb{R}^{n_I-r}$, we get $R_1 \tilde{Q}_I R_1's_1 + Q_I^T[D_Ix + d_I] = 0$. The latter leads to

$$\begin{align*}
[R_1 & R_2][\tilde{Q}_I[R_1 R_2]'s_1 + Q_I[D_Ix + d_I] = 0 \quad (29a) \\
Q_2[D_Ix + d_I] = 0. \quad (29b)
\end{align*}$$

(i) If condition (26) is satisfied, then $Q_2D_I \neq 0$ or $Q_2d_I \neq 0$, or both. If $Q_2D_I \neq 0$, clearly the set of vectors $x$ satisfying (29b) is not full-dimensional, and therefore $CR_I$ is not full-dimensional, because (23a) can only be satisfied on an affine subspace of dimension smaller than $m$. Otherwise, if $Q_2D_I = 0$ and $Q_2d_I \neq 0$, then (29b) has no solution, and $CR_I$ is empty.

(ii) Since $Q_I$ and $R_1$ are nonsingular, matrix $[R_1 R_2][Q_I[R_1 R_2]']$ is invertible, which leads to $s_1(x) = M_I(D_Ix + d_I)$, and hence $y_I(x) = Q_I \left[ M_I[D_Ix + d_I] \right] s_2$.

Then $G'_I y_I(x) = (Q_2 R_I E_I^{-1})'Q_2 s_I = (E_I^{-1})'[R_1 0] s_I(x) = (E_I^{-1})'[R_1 R_2]' s_I(x)$. Replacing $G'_I y_I(x)$ 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 $CR_I = \{ x : \exists s_2 \text{ such that } x s_2 \in P \}$ can be computed by projecting $P$ onto $\mathbb{R}^m$ to get rid of the $n_I - r$ components of $s_2$.

The need for polyhedral projections was reported in [9, Sect. 4.1.1], in which the condition $Q_2S_I = 0$ was also recognized as a condition for full-dimensionality of $CR_I$. A result related to Theorem 6 was shown in [10, Th. 3].

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

$$\text{rank}[G_I - S_I - W_I] > r \quad (30)$$

implies that $CR_I$ is not a full-dimensional critical region.

**Proof:** Since $\tilde{r} \triangleq \text{rank}[G_I - S_I - W_I] = \text{rank}[G_IE_I - S_I E_I - W_I E_I] = \text{rank}[R_I - Q_I S_I E_I - Q'_I W_I E_I]$, then

$$\tilde{r} = \text{rank} \begin{bmatrix} R_1 & R_2 & -Q_I S_I E_I & -Q'_I W_I E_I \end{bmatrix} \quad (31)$$

Since $Q_2[D_I d_I] = Q_2G_2Q_2^{-1}[F \ c] + Q_2[S_I W_I]$ and $Q_2G_2 = [0 \ I][Q'_I Q_I R_I = [0 \ I]\begin{bmatrix} R_1 & R_2 \ 0 & 0 \end{bmatrix} = 0$, then $Q_2D_I = Q_2S_I$ and $Q_2d_I = Q_2W_I$. As $R_I$ is upper triangular, det $R_1 \neq 0$, and $E_I$ is nonsingular, the condition in (26) is satisfied iff $\tilde{r} > r$. 

**Lemma 7** is a slight generalization of the result of [26, Th. 3]. This states that when the LICQ condition is violated ($n_I > r$), then the condition $\tilde{r} = n_I > r$, is an indicator that $CR_I$ is not full-dimensional, while Lemma (7) only requires $\tilde{r} > r$ (clearly, $\tilde{r} \leq n_I$).

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 $I_0$ and its corresponding $CR_{I_0}$, finding all neighboring regions $CR_I$ of $CR_{I_0}$, corresponding to new combinations $I$, and then proceeds recursively until no optimal combination $I$ exists such that $CR_I \cap X \neq \emptyset$.

We focus first on determining an initial optimal combination $I_0$. In case the polyhedron $CR_{I_0} = \{ x \in \mathbb{R}^r : G^{-1}(Q^{-1}(Fx + c)) \leq W + Sx, E_I x \leq c_x \}$ is nonempty, $I_0 = \emptyset$ is a valid initial combination. Otherwise, the following Lemma 8 provides a general method to determine $I_0$. 

---

**Remark:** The proof of (26a) is based on the fact that $Q_2D_I \neq 0$ and $Q_2d_I \neq 0$, and the general method to determine $I_0$ is based on the fact that $Q_2D_I \neq 0$ or $Q_2d_I \neq 0$, or both. If $Q_2D_I \neq 0$, clearly the set of vectors $x$ satisfying (29b) is not full-dimensional, and therefore $CR_I$ is not full-dimensional, because (23a) can only be satisfied on an affine subspace of dimension smaller than $m$. Otherwise, if $Q_2D_I = 0$ and $Q_2d_I \neq 0$, then (29b) has no solution, and $CR_I$ is empty.

---

**Remark:** The proof of (26b) is based on the fact that $Q_2D_I \neq 0$ and $Q_2d_I \neq 0$, and the general method to determine $I_0$ is based on the fact that $Q_2D_I \neq 0$ or $Q_2d_I \neq 0$, or both. If $Q_2D_I \neq 0$, clearly the set of vectors $x$ satisfying (29b) is not full-dimensional, and therefore $CR_I$ is not full-dimensional, because (23a) can only be satisfied on an affine subspace of dimension smaller than $m$. Otherwise, if $Q_2D_I = 0$ and $Q_2d_I \neq 0$, then (29b) has no solution, and $CR_I$ is empty.
Lemma 8: Consider problem (3) with \( Q = Q' > 0 \) and let \( X = \{ x \in \mathbb{R}^m : E_i x \leq e_x \} \subseteq \mathbb{R}^m \). Consider the QP problem

\[
\min_{x} \frac{1}{2} z^T Q z + (F x + c)^T z + \frac{1}{2} (x - \bar{x})^T B (x - \bar{x})
\]

s.t. \( G x - S x \leq W \)

\[ E_i x \leq e_x \]

(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 \( \begin{bmatrix} z_0 \\ x_0 \end{bmatrix} \) of (32) the combination \( I_0 \) such that \( G_{I_0} z_0 = S_{I_0} x_0 + W_{I_0} \) is an optimal combination for active constraints for problem (3).

Proof: The proof is rather simple. If (32) is infeasible, then no vector \( x \in X \) exists such that \( G x \leq W + S x \), and therefore \( X \cap X_f = \emptyset \). If (32) admits a solution \( \begin{bmatrix} z_0 \\ x_0 \end{bmatrix} \), 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 \( \begin{bmatrix} z_1 \\ x_0 \end{bmatrix} \) satisfies the constraints in (3) and has a lower cost. Hence, \( I_0 \) 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 \( I_0 \) has been determined, the corresponding critical region \( CR_{I_0} \) 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 \( CR_{I_0} \), we recall the following theorem [10, Theorem 2]:

Theorem 7 (Neighboring Regions): Consider problem (3) with \( Q = Q' > 0 \), let \( \mathcal{I} \) an arbitrary region of constraints and \( CR_\mathcal{I} = \{ x \in \mathbb{R}^m : E F x \leq E \} \) 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., \( H^{-1}_{CR_\mathcal{I}} [D x] \neq 0 \) in (25b). Let \( CR_{\mathcal{I}'} \) be a full-dimensional critical region neighboring \( CR_\mathcal{I} \), let \( P_1 = \{ x \in \mathbb{R}^m : E F x = e^T ) \} = CR_{\mathcal{I}'} \cap CR_{\mathcal{I}} \) the common facet between \( CR_{\mathcal{I}'} \) and \( CR_{\mathcal{I}} \), and assume that the LICQ property is satisfied on \( P_1 \). Then the optimal combination \( I_1 \) associated with \( CR_{I_1} \) is related to \( I \) as follows:

1) If \( P_1 \) corresponds to a constraint of type \( w_k(x) \geq 0 \) in (25), then \( I_1 = I \cup \{ k \} \).

2) If \( P_1 \) corresponds to a constraint of type \( y_h(x) \geq 0 \) in (25), then \( I_1 = I \setminus \{ h \} \).

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], [27]. This approach is based on the following idea. For each full-dimensional facet \( P_j = \{ x \in \mathbb{R}^m : E_j x = e_j^T \} \) of \( CR_\mathcal{I} \), we determine the Chebychev center \( x_j \) of \( P_j \) as described in Remark 2, set \( \bar{x}_j = x_j + \epsilon E_j^T \), 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 \( I_j \) 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 \( I \) 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 \( I_i \) contained in \( \mathcal{R} = \{ I_1, \ldots, I_M \} \), where \( M = \# \mathcal{R} \) is the cardinality of \( \mathcal{R} \).

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 \( \{ [z \ x] : G x \leq W + S x, x \in X \} \) (Section III-B);
3) solve (32); if infeasible, \( X_f \cap X = \emptyset \); go to step 18;
4) compute \( I_0 \) (Lemma 8); set \( \mathcal{U} \leftarrow \{ I_0 \}; \)
5) while \( \mathcal{U} \neq \emptyset \) do:
6) get an element \( I \in \mathcal{U} \); set \( \mathcal{U} \leftarrow \mathcal{U} \setminus \{ I \}; \)
7) if \( I \in \mathcal{E} \) set \( \mathcal{E} \leftarrow \mathcal{E} \cup \{ I \}; \)
8) compute \( CR_{\mathcal{I}} \cap X \) (Lemma 6 or Theorem 6);
9) if \( CR_{\mathcal{I}} \cap X \) is full-dimensional (Section III-C) or \( I = I_0 \);
10) remove redundant inequalities in \( CR_{\mathcal{I}} \cap X \) (Section III-B);
11) for all facets \( f \) of \( CR_{\mathcal{I}} \):
12) find the neighboring combination \( I_j \) (Theorem 7 or Remark 5);
13) if \( I_j \) exists, set \( \mathcal{U} \leftarrow \mathcal{U} \cup \{ I_j \}; \)
14) if \( CR_{\mathcal{I}} \cap X \) is full-dimensional:
15) store \( CR_{\mathcal{I}} \cap X = \{ x \in \mathbb{R}^m : E F x \leq e^T \} \)
16) set \( \mathcal{R} \leftarrow \mathcal{R} \cup \{ I \}; \)
17) compute \( z(x) = K^2 x + h^2 \) as in (24) and (25b), or as in (27);
18) end.

Output: Multiparametric solution \( \{ K^2, h^2, E^i, e^i \} \forall \mathcal{I} \in \mathcal{R} \).

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 solution \( \{ K^2, h^2, E^i, e^i \} \), and by the worst-case execution time required to compute \( K^2 x + h^2 \) 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 [26], [28]–[30].

An upper-bound to \( M \) is \( 2^n \), which is the number of all possible combinations of active constraints. In practice \( M \) is much smaller than \( 2^n \), 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 \( n_a \) component of the multiparametric solution \( \{x, i(x)\} \) 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:

\[
i(x) \in \arg \min_{j=1,\ldots,n_i} \{ \max_j \{ E_j^i x - e_j^i \} \} \quad (33a)
\]

\[
u(x) = K^i(x) x + h^i(x) \quad (33b)
\]

where \( n_i = \dim (E^i) \) is the number of inequalities defining the \( i \)th critical region \( C R^i \). 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 \( E^i x = e_j^i \); second, to compute suboptimal explicit control laws by extrapolation, as suggested in the following section. In fact, \( u(x) = u^*(x) \) in case \( x \in C R^i \), for some \( i \), otherwise (33) extrapolates the affine law \( K^i(x) x + h^i(x) \) outside region \( C R^i \), where \( i(x) \) identifies the region whose constraints are least violated by vector \( x \). Clearly, in this case the satisfaction of constraints (1c) and(1d) is no longer guaranteed.

Lemma 9: Given a set of quadruples \( \{ K^u, h^i, E^i, e^i \} \), \( i=1,\ldots,M \), 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

\[
\min_{\epsilon, \delta, z} \sum_{i=1}^M z_i \\
\text{s.t.} \quad \epsilon_j \geq E_j^i x - e_j^i, \quad \forall j = 1, \ldots, n_i \\
\epsilon_i \leq \epsilon_j + 2M(1 - \delta_j), \quad \forall j = 1, \ldots, n_i, \ j \neq i \\
z_i \leq \epsilon_i + M(1 - \delta_i) \\
z_i \geq \epsilon_i - M(1 - \delta_i) \\
\epsilon_i \leq M \delta_i \\
z_i \geq -M \delta_i \\
\forall i = 1, \ldots, M \\
\sum_{i=1}^M \delta_i = 1 \quad (34)
\]

for \( x \in X \), where \( M \) is an upper-bound on \( \pm (E_j^i x - e_j^i) \) for all \( x \in X \), \( i \in \{1, \ldots, M\} \), \( j \in \{1, \ldots, n_i\} \). Problem (34) determines a solution \( \delta(x) \) such that \( \delta_j(x) = 1 \) whenever \( E_j^i x - e_j^i \leq 0 \), \( \forall j = 1, \ldots, n_i \). As the solution \( \delta(x) \) of (34) is a piecewise constant function of \( x \) [31], the scalar product

\[
u(x) = \left[ \begin{array}{c} K^i x + h^i \\
\vdots \\
K^M x + h^M \end{array} \right] \left[ \begin{array}{c} \delta_1(x) \\
\vdots \\
\delta_M(x) \end{array} \right]
\]

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 \( K^u x + h^u \) of the multiparametric solution \( z(x) \) are the same, either by direct recognition of convexity and computation of unions of pairs of sets [24, Algorithm 4.1], or by optimal merging algorithms [32] to get a minimal number \( M \) of partitions.

Suboptimal solutions can be obtained by removing critical regions \( C R^u \) 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] and [33]. The idea is to run closed-loop MPC simulations based on on-line QP and only store the critical regions \( C R^u \) corresponding to the most visited combinations \( I \). 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 \( C R^i \), we first compare the approaches of Theorems 2, 3, and 4 based on NLLS and the common approach of labelling an inequality constraint \( E_i x \leq e_i \) of a polyhedron \( P = \{ x \in \mathbb{R}^m : E x \leq e \} \) as non-redundant if the LP problem

\[
\epsilon_i = \max_x E_i x - e_i \\
\text{s.t.} \quad E_i x \leq e_j, \ j \neq i \quad (35)
\]

returns \( \epsilon_i > 0 \). The LP (35) is solved by GLPK [34], while NLLS 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 10 \( m \) 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 NLLS 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 \( P \neq \emptyset \).

**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 [35]. In [19] the authors prove that the norm of \( |A_{v} - b| \) 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., [35] and the references therein included, and [36], [37]. In particular, for the NNLS algorithm a few ideas to deal with finite precision arithmetics are described in [19, p. 164–165].

### B. Performance of the mpQP Solver

We compare Algorithm 3 with the mpQP solvers of the Hybrid Toolbox [38] and of the Multi Parametric Toolbox [39]. 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 [38] and [39] 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 [40]. With reference to the MPC formulation (9) we have \( A \in \mathbb{R}^{4 \times 4} \), \( B_{u} \in \mathbb{R}^{4} \), \( B_{v} = \emptyset \), \( B \in \mathbb{R}^{2 \times 4} \), \( D_{u} \in \mathbb{R}^{2} \), \( D_{v} = \emptyset \), \( u_{\text{min}} = -220 \), \( u_{\text{max}} = 220 \), \( \Delta y_{\text{min}} = -\infty \), \( \Delta y_{\text{max}} = +\infty \), \( y_{\text{min}} = [-\infty, -78.54] \), \( y_{\text{max}} = [78.54, +\infty] \), \( V_{\text{min}} = [1] \), \( V_{\text{max}} = [1] \), \( N = N_{c} = 7 \), \( N_{u} = 2 \), \( \rho_{u} = 1000 \), \( Q_{y} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \), \( R_{\Delta u} = 0.0025 \), \( R_{u} = 0 \). 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 \( X = \{ x \in \mathbb{R}^{5} : -1000 \leq x_{1,4} \leq 1000, -5 \leq x_{5,6} \leq 5, -221 \leq x_{6} \leq 221 \} \) in 5.3296 s using Algorithm 3 (it takes 6.0470 s with the Hybrid Toolbox [38] and 6.2476 s with the Multi-Parametric Toolbox [39]). A section in the \((x_{1}, x_{2})\)-space of the six-dimensional partition, which consists of 215 regions, is reported in Fig. 1.

### VII. Conclusion

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.) [41].

**References**


---

**Table II**

<table>
<thead>
<tr>
<th>( q )</th>
<th>( m )</th>
<th>Hybrid Thx. [38]</th>
<th>MPT [39]</th>
<th>mpQP NLNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>0.0174</td>
<td>0.0256</td>
<td>0.0026</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.0263</td>
<td>0.0356</td>
<td>0.0038</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.0432</td>
<td>0.0559</td>
<td>0.0061</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.0650</td>
<td>0.0850</td>
<td>0.0097</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0.0827</td>
<td>0.1105</td>
<td>0.0126</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0.0347</td>
<td>0.0396</td>
<td>0.0050</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>0.0583</td>
<td>0.0680</td>
<td>0.0092</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>0.0916</td>
<td>0.0999</td>
<td>0.0140</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>0.1869</td>
<td>0.2147</td>
<td>0.0322</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>0.3177</td>
<td>0.3611</td>
<td>0.0586</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>0.0398</td>
<td>0.0387</td>
<td>0.0054</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>0.1121</td>
<td>0.1158</td>
<td>0.0191</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>0.2067</td>
<td>0.2001</td>
<td>0.0352</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>0.6180</td>
<td>0.6428</td>
<td>0.1151</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>1.2453</td>
<td>1.3601</td>
<td>0.2426</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>0.1029</td>
<td>0.0763</td>
<td>0.0152</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>0.3698</td>
<td>0.2905</td>
<td>0.0588</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>0.9069</td>
<td>0.7100</td>
<td>0.1617</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>2.2978</td>
<td>1.9761</td>
<td>0.4395</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>6.1220</td>
<td>6.2518</td>
<td>1.2853</td>
</tr>
</tbody>
</table>