

# Explicit MPC for the constrained zonotope case with low-rank matrix updates <sup>★</sup>

Stefan S. Mihai <sup>a</sup>, Florin Stoican <sup>a</sup>, Martin Monnigmann <sup>b</sup>, Bogdan D. Ciubotaru <sup>a</sup>,

<sup>a</sup>*Department of Automatic Control and Systems Engineering,  
Nat. Univ. of Science and Technology Politehnica Bucharest, 060042, Bucharest, Romania.*

<sup>b</sup>*Automatic Control and System Theory, Ruhr University, Bochum, 44801 Bochum, Germany.*

---

## Abstract

Solving the explicit Model Predictive Control (MPC) problem requires enumerating all critical regions and their associated feedback laws, a task that scales exponentially with the system dimension and the prediction horizon, as well. When the problem's constraints are boxes or zonotopes, the feasible domain admits a compact constrained-zonotope representation. Building on this insight, we exploit the geometric properties of the equivalent constrained-zonotope reformulation to accelerate the computation of the explicit solution. Specifically, we formulate the multi-parametric problem in the lifted generator space and solve it using second-order optimality conditions, employ low-rank matrix updates to reduce computation time, and introduce an analytic enumeration of candidate active sets that yields the explicit solution in tree form.

*Key words:* Explicit model predictive control; Set-based computing; Linear systems; Constrained zonotopes.

---

## 1 Introduction

Fundamentally, Model Predictive Control (MPC) is a feedback control strategy that employs a dynamic model to predict the evolution of the system states and to compute an optimal sequence of constrained control actions over a finite prediction horizon [12]. A prominent line of research focuses on its *explicit formulation*, which exploits the multi-parametric structure of the underlying optimization problem [4,2,16,17,23]. The resulting control law is a piecewise affine function with polyhedral support, partitioning the state space into bounded convex polyhedral regions known as *critical regions*.

The main challenge in explicit MPC is the exponential growth of the critical regions' number with the prediction horizon, making their enumeration computationally demanding. Hence, research has focused on faster implementations, many based on enumerating candidate active constraint sets [16,1,15]. Noteworthy, these sets are linked with the face lattice of the polyhedral feasible domain, which can be exploited in the process of solution finding [20,14].

As the system dimension and number of constraints grow, handling polyhedra quickly becomes intractable, motivating alternative formulations. Thanks to their resilience to the “curse of dimensionality,” constrained zonotopes [19] have been recently used in applications such as reachability analysis [3], projection [22], and fault diagnosis [24]. They can approximate any convex set arbitrarily well [19, Theorem 1] and are closed under intersection, Minkowski addition, and affine transformations.

We propose a novel explicit MPC approach that exploits the geometric properties of the equivalent constrained-zonotope reformulation, extending our previous work [21]. First, the multi-parametric problem is formulated in the lifted generator space and solved using second-order optimality conditions. Second, computation is accelerated via low-rank matrix updates, thus reducing the time required to compute each critical region. Finally, we introduce an analytic enumeration of candidate active sets that naturally yields the explicit solution in tree form.

The paper is organized as follows. Section 2 introduces the required definitions. Section 3 presents the constrained-zonotope reformulation of the MPC problem. Section 4 outlines the computation of the explicit solution, and Section 5 describes the associated algorithms. Section 6 provides numerical results, followed by conclusions in Section 7.

---

<sup>★</sup> Corresponding author: Ștefan S. Mihai.

*Email addresses:* [sergiu.stefan.mihai@upb.ro](mailto:sergiu.stefan.mihai@upb.ro) (Stefan S. Mihai), [florin.stoican@upb.ro](mailto:florin.stoican@upb.ro) (Florin Stoican), [martin.moennigmann@rub.de](mailto:martin.moennigmann@rub.de) (Martin Monnigmann), [bogdan.ciubotaru@upb.ro](mailto:bogdan.ciubotaru@upb.ro) (Bogdan D. Ciubotaru).

## Notation

|                    |                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------|
| $\mathbb{A}$       | a finite, sorted set of natural numbers                                                             |
| $V_{\mathbb{A}}$   | the sub-matrix formed from the rows of matrix $V$ indexed by the elements in $\mathbb{A}$           |
| $V_{(\mathbb{A})}$ | a matrix computed in relation to $\mathbb{A}$                                                       |
| $V^+$              | the Moore–Penrose generalized inverse of $V$                                                        |
| $v^{(k)}$          | the $k$ -th column of the matrix $V$                                                                |
| $\Lambda(V)$       | the spectrum of $V$ , i.e., $\{\lambda \in \mathbb{C} : \det(\lambda I - V) = 0\}$                  |
| $e_k$              | the $k$ -th vector of the canonical basis of $\mathbb{R}^n$                                         |
| $n_{\bullet}$      | the number of elements in $\bullet$ (e.g., $n_{\mathbb{A}} =  \mathbb{A} $ )                        |
| $\otimes$          | the Kronecker product                                                                               |
| $x_{i j}$          | the predicted value of variable $x$ at time $i \geq j$ , based on information available at time $j$ |
| $\circ$            | Hadamard product                                                                                    |

## 2 Context and framework

We summarize the standard definitions of polyhedral and constrained zonotopic sets, which will be used later to describe the explicit MPC formulation. Additional details can be found in [9,19,18].

**Definition 1 (Polyhedral set, polytope)** *The matrix pair  $(A \in \mathbb{R}^{d_H \times n}, b \in \mathbb{R}^{d_H})$  gives the half-space representation of a polyhedral set*

$$P(A, b) = \{x \in \mathbb{R}^n : A_i x \leq b_i, \forall i \in \{1, \dots, d_H\}\}, \quad (1)$$

where  $b_i, A_i$  denote the  $i$ -th element of vector  $b$ , and the  $i$ -th row of matrix  $A$ , respectively. A bounded polyhedral set is called a polytope.

A bounded face  $F$  of the polytope  $P$ , defined as in Def. 1, admits a half-space representation obtained by separating the active and inactive constraints, indexed by the sets  $\mathbb{A}$  and  $\mathbb{I}$ , respectively, where  $\mathbb{A} \cup \mathbb{I} = \{1, \dots, d_H\}$  and  $\mathbb{A} \cap \mathbb{I} = \emptyset$ . The face  $F$  is then explicitly given by

$$F(\mathbb{A}) = \{x \in \mathbb{R}^n : A_i x = b_i, \forall i \in \mathbb{A}, \\ A_j x \leq b_j, \forall j \in \mathbb{I}\}. \quad (2)$$

The collection of all faces forms the face lattice of the polytope, which is a partially ordered set (*poset*).

Zonotopes are symmetric polytopes that can be represented as a Minkowski sum of line segments, [9, Chap. 4.8]. Although they exhibit strong numerical robustness in high dimensions, they are not closed under set intersection. Constrained zonotopes (CZs) extend zonotopes to overcome this limitation, [19].

**Definition 2 (Constrained zonotope)** *Vectors  $c \in \mathbb{R}^n$ ,  $\theta \in \mathbb{R}^{n_c}$ , and matrices  $G \in \mathbb{R}^{n \times n_g}$ ,  $F \in \mathbb{R}^{n_c \times n_g}$ , define the constrained zonotope*

$$Z = \langle c, G, F, \theta \rangle \\ = \{x \in \mathbb{R}^n : x = c + G\xi, \|\xi\|_{\infty} \leq 1, F\xi = \theta\}. \quad (3)$$

Constrained zonotopes possess several useful properties [19], three of which are of particular relevance in this work. First, they are closed under affine transformations:

$$r + RZ_1 = \langle r + Rc_1, RG_1, F_1, \theta_1 \rangle. \quad (4)$$

Second, CZs are closed under Minkowski sum:

$$Z_1 \oplus Z_2 = \left\langle c_1 + c_2, \begin{bmatrix} G_1 & G_2 \end{bmatrix}, \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix}, \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \right\rangle. \quad (5)$$

Third, CZs are closed under set intersection:

$$Z_1 \cap Z_2 = \left\langle c_1, \begin{bmatrix} G_1 & 0 \end{bmatrix}, \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix}, \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \right\rangle, \quad (6)$$

where  $Z_1 := \langle c_1, G_1, F_1, \theta_1 \rangle$  and  $Z_2 := \langle c_2, G_2, F_2, \theta_2 \rangle$ .

**Remark 3** *Operations (5) and (6) can be computationally demanding, as they increase the number of generators and constraints. Redundancies may be mitigated using techniques similar to those in [19,18].*

**Example 4** *Consider a zonotope in  $\mathbb{R}^2$  described by the center  $c = [0.15 \ 0.25]^\top$  and generator matrix  $G = [g_1 \ g_2 \ g_3]$ , where  $g_1 = [-0.75 \ 0]^\top$ ,  $g_2 = [0 \ 0.5]^\top$ , and  $g_3 = [1 \ 0.25]^\top$ . Then, let the plane  $F\xi = \theta$ , where  $F = [0.5 \ -2 \ 0.25]$  and  $\theta = 1$ , define the equality constraint; incorporating this constraint as described in (3) results in the constrained zonotope.*

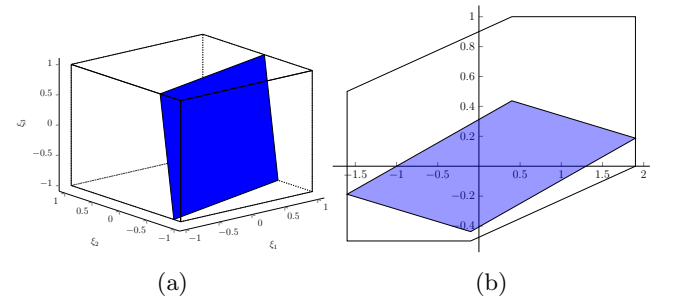


Fig. 1. (a) Domain of  $\xi \in \mathbb{R}^{n_g}$ ,  $n_g = 3$  and hyperplane defined by  $F\xi = \theta$ ; (b) Zonotope (black contour) and constrained zonotope (blue) in  $\mathbb{R}^n$ ,  $n = 2$ .

Both the zonotope  $\langle c, G, \emptyset, \emptyset \rangle$  and the constrained zonotope  $\langle c, G, F, \theta \rangle$  are illustrated in Fig. 1b. The key difference lies in the admissible values of  $\xi$  used to generate points of the form  $c + G\xi$ . While the zonotope maps the entire hypercube  $\|\xi\|_{\infty} \leq 1$ , the constrained zonotope restricts  $\xi$  to the intersection of this hypercube with a hyperplane, as shown in Fig. 1a.

### 3 Constrained zonotope formulation

Consider a discrete-time, linear time-invariant system characterized by the state-space representation

$$x_{k+1} = A_d x_k + B_d u_k, \quad (7)$$

where  $x_k \in \mathbb{R}^n$  and  $u_k \in \mathbb{R}^m$  denote the state and input vectors at time step  $k$  and  $A_d \in \mathbb{R}^{n \times n}$ ,  $B_d \in \mathbb{R}^{n \times m}$  are the state and input matrices. State and input constraints are represented as set membership inclusions  $x_k \in \mathcal{X} \subset \mathbb{R}^n$ ,  $u_k \in \mathcal{U} \subset \mathbb{R}^m$ , where  $\mathcal{X}$  and  $\mathcal{U}$  are polytopes. To control such systems, while ensuring constraint satisfaction and cost minimization over a finite prediction horizon of length  $N$ , a well-established approach is MPC<sup>1</sup> [12], respectively

$$\arg \min_{\substack{u_{0|k}, \dots, u_{N-1|k} \\ x_{0|k}, \dots, x_{N|k}}} x_{N|k}^\top S x_{N|k} + \sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}) \quad (8a)$$

$$\text{s.t. } x_{i+1|k} = A_d x_{i|k} + B_d u_{i|k}, \quad (8b)$$

$$x_{i|k} \in \mathcal{X}, \quad u_{i|k} \in \mathcal{U}, \quad (8c)$$

$$x_{N|k} \in \mathcal{T}. \quad (8d)$$

The stage cost  $\ell(x_{i|k}, u_{i|k})$  is defined as  $\ell(x_{i|k}, u_{i|k}) = x_{i|k}^\top Q x_{i|k} + u_{i|k}^\top R u_{i|k}$ , where  $Q \succeq 0$  and  $R \succ 0$  penalize deviations in the state and control input, respectively. The terminal weighting matrix  $S = S^\top \succ 0$ , together with the polytopic terminal set  $\mathcal{T} \subset \mathbb{R}^n$ , ensures recursive feasibility and asymptotic stability [12]. To close the loop, at simulation step  $k$ , the initial predicted state is set to the current state,  $x_{0|k} = x_k$ , and the control input applied in (7) is  $u_k = u_{0|k}$ .

Reformulating (8) into the equivalent multi-parametric quadratic program (mpQP) by using the substitution  $x_{i|k} = A^i x_0 + \sum_{j=0}^{i-1} A^{i-1-j} B u_{j|k}$  (see [7]) leads to

$$\arg \min_{\mathbf{u}_{[0:N-1]}} \frac{1}{2} \mathbf{u}_{[0:N-1]}^\top \tilde{Q} \mathbf{u}_{[0:N-1]} + x_0^\top \tilde{H} \mathbf{u}_{[0:N-1]} \quad (9a)$$

$$\text{s.t. } A \mathbf{u}_{[0:N-1]} \leq E x_0 + b, \quad (9b)$$

where  $\tilde{Q} \in \mathbb{R}^{Nm \times Nm}$ ,  $\tilde{H} \in \mathbb{R}^{n \times Nm}$ ,  $A \in \mathbb{R}^{q \times Nm}$ ,  $b \in \mathbb{R}^q$ , and  $E \in \mathbb{R}^{q \times n}$  are obtained from (8) through standard matrix manipulations. Denoting  $\mathbf{u}_{[0:N-1]} = [u_{0|k}^\top \ u_{1|k}^\top \ \dots \ u_{N-1|k}^\top]^\top$ . For further use,  $\mathbf{x}_{[0:N-1]} = [x_{0|k}^\top \ x_{1|k}^\top \ \dots \ x_{N-1|k}^\top]^\top$ , and note the relations

$$\mathbf{x}_{[0:N-1]} = \tilde{A}_{[0:N-1]} x_0 + \tilde{B}_{[0:N-1]} \mathbf{u}_{[0:N-1]}, \quad (10a)$$

$$x_N = \tilde{A}_N x_0 + \tilde{B}_N \mathbf{u}_{[0:N-1]}, \quad (10b)$$

<sup>1</sup> To simplify notation, we use the shorthand  $x_{i|k} = x_{k+i|k}$  and  $u_{i|k} = u_{k+i|k}$  throughout the paper. Whenever clear from context, we denote  $x_0 = x_{0|k}$ .

where

$$\tilde{A}_{[0:N-1]} = \begin{bmatrix} I & A_d^\top & A_d^2^\top & \dots & A_d^{N-1}^\top \end{bmatrix}^\top, \quad (11a)$$

$$\tilde{A}_N = A_d^N, \quad (11b)$$

$$\tilde{B}_{[0:N-1]} = \sum_{i=2}^N \sum_{j=1}^{i-1} (e_i e_j^\top) \otimes (A_d^{i-j-1} B_d), \quad (11c)$$

$$\tilde{B}_N = \begin{bmatrix} A_d^{N-1} B_d & A_d^{N-2} B_d & \dots & B_d \end{bmatrix}. \quad (11d)$$

By revisiting the state and input constraints in (8c) and the terminal constraint (8d), and incorporating the shorthand notation from (10), we observe that the feasible domain of (8) can be rewritten as  $\tilde{\mathcal{X}} \cap \tilde{\mathcal{T}} \cap \tilde{\mathcal{U}}$ , with

$$\tilde{\mathcal{X}} = \left\{ \tilde{A}_{[0:N-1]} x_0 + \tilde{B}_{[0:N-1]} \mathbf{u}_{[0:N-1]} \in \mathcal{X} \times \dots \times \mathcal{X} \right\},$$

$$\tilde{\mathcal{U}} = \mathcal{U} \times \dots \times \mathcal{U}, \quad (12)$$

$$\tilde{\mathcal{T}} = \left\{ \tilde{A}_N x_0 + \tilde{B}_N \mathbf{u}_{[0:N-1]} \in \mathcal{T} \right\}.$$

Starting from the initial constrained zonotope representation of the constraint sets  $\mathcal{X}$ ,  $\mathcal{U}$  and  $\mathcal{T}$ , we derive a constrained zonotope description of the feasible domain associated with (8), which is then used to efficiently derive the explicit MPC solution. For later use, we denote the equivalent polyhedral representation  $\tilde{\mathcal{X}} \cap \tilde{\mathcal{U}} \cap \tilde{\mathcal{T}} = \{x \in \mathbb{R}^{Nm} : A \mathbf{u}_{[0:N-1]} \leq b + E x_0\}$ , with  $A \in \mathbb{R}^{q \times Nm}$ ,  $b \in \mathbb{R}^q$  and  $F \in \mathbb{R}^{q \times n}$ , appropriately constructed.

Consider the case where the state and input constraints are represented in zonotopic form as  $\mathcal{X} = \langle c_{\mathcal{X}}, G_{\mathcal{X}} \rangle$ ,  $\mathcal{U} = \langle c_{\mathcal{U}}, G_{\mathcal{U}} \rangle$ .

Typically, the terminal set  $\mathcal{T}$  is defined as the result of a set recurrence  $\Omega_{k+1} = (A_d + B_d K)^{-1} \Omega_k \cap \Omega_0$ , where  $\Omega_0 = \mathcal{X} \cap \mathcal{U}$ , for a predefined stabilizing static gain  $K$ . Since this terminal set can be conveniently represented as a constrained zonotope [10], we express it as  $\mathcal{T} = \langle c_{\mathcal{T}}, G_{\mathcal{T}}, F_{\mathcal{T}}, \theta_{\mathcal{T}} \rangle$ . For reference, the generator matrices  $G_{\{\mathcal{X}, \mathcal{U}, \mathcal{T}\}}$  belong to  $\mathbb{R}^{\{n, m, n\} \times \{g_{\mathcal{X}}, g_{\mathcal{U}}, g_{\mathcal{T}}\}}$ , while the constraint parameters  $(F_{\mathcal{T}}, \theta_{\mathcal{T}})$  are elements of  $\mathbb{R}^{c_{\mathcal{T}} \times g_{\mathcal{T}}} \times \mathbb{R}^{c_{\mathcal{T}}}$ .

The notation allows us to express (12) in the form of a constrained zonotope. Enforcing the stage constraints  $x_k \in \mathcal{X}$  and control inputs  $u_k \in \mathcal{U}$ , for  $k = 0 : N-1$ , along with the terminal constraint  $x_N \in \mathcal{T}$ , gives

$$\tilde{A}_{[0:N-1]} x_0 + \tilde{B}_{[0:N-1]} \mathbf{u}_{[0:N-1]} \in \langle \mathbf{1}_N \otimes c_{\mathcal{X}}, I_N \otimes G_{\mathcal{X}} \rangle, \quad (13a)$$

$$\mathbf{u}_{[0:N-1]} \in \langle \mathbf{1}_N \otimes c_{\mathcal{U}}, I_N \otimes G_{\mathcal{U}} \rangle, \quad (13b)$$

$$\tilde{A}_N x_0 + \tilde{B}_N \mathbf{u}_{[0:N-1]} \in \langle c_{\mathcal{T}}, G_{\mathcal{T}}, F_{\mathcal{T}}, \theta_{\mathcal{T}} \rangle. \quad (13c)$$

**Lemma 5** Introducing  $\xi \in \mathbb{R}^{N(g_X + g_U) + G_T}$  and applying substitution  $\mathbf{u}_{[0:N-1]} = c_D + G_D \xi$  in (9) leads to the equivalent formulation

$$\arg \min_{\xi} \quad \frac{1}{2} \xi^\top G_D^\top \tilde{Q} G_D \xi + c_D^\top \tilde{Q} G_D \xi + x_0^\top \tilde{H} G_D \xi + x_0^\top \tilde{H} c_D + \frac{1}{2} c_D^\top \tilde{Q} c_D \quad (14a)$$

$$s.t. \quad F_D \xi - \theta_{1,D} - \theta_{2,D} x_0 = \mathbf{0}_{\bar{D}}, \quad (14b)$$

$$Y \xi - \mathbf{1}_{2\bar{D}} \leq \mathbf{0}_{2\bar{D}}, \quad (14c)$$

where  $Y = [I_{\bar{D}} \ -I_{\bar{D}}]^\top$  and

$$c_D = \mathbf{1}_N \otimes c_U, \quad (15)$$

$$G_D = \begin{bmatrix} I_N \otimes G_U & \mathbf{0}_{Nn \times (Ng_X + g_T)} \end{bmatrix},$$

$$F_D = \begin{bmatrix} \tilde{B}_{[0:N-1]}(I_N \otimes G_U) & -I_N \otimes G_X & \mathbf{0}_{Nn \times g_T} \\ \tilde{B}_N(I_N \otimes G_U) & \mathbf{0}_{n \times Ng_X} & -G_T \\ \mathbf{0}_{c_T \times Ng_U} & \mathbf{0}_{c_T \times Ng_X} & F_T \end{bmatrix},$$

$$\theta_{1,D} = \begin{bmatrix} \mathbf{1}_N \otimes c_X \\ c_T \\ \theta_T \end{bmatrix} - \begin{bmatrix} \tilde{B}_{[0:N]} \\ \mathbf{0}_{c_T} \end{bmatrix} (\mathbf{1}_N \otimes c_U), \theta_{2,D} = \begin{bmatrix} \tilde{A}_{[0:N]} \\ \mathbf{0}_{c_T \times n} \end{bmatrix}.$$

The term  $\bar{D} = N(g_X + g_U) + g_T$  denotes the number of generators and  $\bar{n}_c = (N+1)n + c_T$  the number of equalities.  $\tilde{A}_{[0:N]}$  and  $\tilde{B}_{[0:N]}$  are obtained by vertically concatenating  $(\tilde{A}_{[0:N-1]}, \tilde{A}_N)$  and  $(\tilde{B}_{[0:N-1]}, \tilde{B}_N)$ , respectively.

**PROOF.** Substituting (13b) into (13a), following the approach in (6), gives the constrained zonotope bounding the control inputs sequence  $\mathbf{u}_{[0:N-1]}$  as

$$\left\langle \mathbf{1}_N \otimes c_U, \begin{bmatrix} I_N \otimes G_U & \mathbf{0}_{Nm \times Ng_X} \end{bmatrix}, \begin{bmatrix} \tilde{B}_{[0:N-1]}(I_N \otimes G_U) & -I_N \otimes G_X \end{bmatrix}, \mathbf{1}_N \otimes c_X - \tilde{B}_{[0:N-1]}(\mathbf{1}_N \otimes c_U) - \tilde{A}_{[0:N-1]}x_0 \right\rangle.$$

Including (13c) in the previous set and performing a minor regrouping of the terms to highlight  $x_0$  yields

$$\left\{ \mathbf{u}_{[0:N-1]} = c_D + G_D \xi, F_D \xi = \theta_{1,D} + \theta_{2,D} x_0, \|\xi\|_\infty \leq 1 \right\}, \quad (16)$$

with  $c_D, G_D, F_D, \theta_{1,D}$  and  $\theta_{2,D}$  defined as in (15).

Making the change of variable  $\mathbf{u}_{[0:N-1]} = c_D + G_D \xi$  allows to reformulate (9) into (14), thus concluding the proof.  $\square$

## 4 Solution computation and improvements

As per (16), the decision variable  $\xi$  lies in the intersection between a fixed hypercube and an affine subspace, parameterized in  $x_0$ . Hence, we focus on the structural simplicity of (14), leveraging its constrained zonotope representation, to compute efficiently the explicit solution.

### 4.1 The critical region and the associated affine law

Although the reformulated problem (14) features a quadratic cost with strictly positive definite weighting matrix  $\tilde{Q}$ , the Hessian  $G_D^\top \tilde{Q} G_D$  is not strictly positive definite. Since  $G_D \in \mathbb{R}^{Nm \times \bar{D}}$ , its rank is at most  $Nm < \bar{D}$ , implying rank deficiency. As a result, the Karush–Kuhn–Tucker (KKT) conditions remain necessary but are no longer sufficient. To recover the optimality, second-order conditions must be included to account for the null space of  $G_D^\top \tilde{Q} G_D$ .

**Proposition 6** For a candidate set of active inequalities  $\mathbb{A} \subset \{1, \dots, 2\bar{D}\}$ , the optimal solution minimizing (14) leads to the affine law

$$\mathbf{u}_{(\mathbb{A}),[0:N-1]}^*(x_0) = G_D K_{(\mathbb{A})}^{-1} \kappa_{2,(\mathbb{A})} x_0 + \left( c_D + G_D K_{(\mathbb{A})}^{-1} \kappa_{1,(\mathbb{A})} \right), \quad (17)$$

and its associated (possibly empty) critical region

$$CR_{(\mathbb{A})} = \left\{ \begin{bmatrix} Y_{\mathbb{I}} K_{(\mathbb{A})}^{-1} \kappa_{2,(\mathbb{A})} \\ -S_{\mathbb{A},2} \end{bmatrix} x_0 + \begin{bmatrix} Y_{\mathbb{I}} K_{(\mathbb{A})}^{-1} \kappa_{1,(\mathbb{A})} \\ -s_{\mathbb{A},2} \end{bmatrix} \leq \begin{bmatrix} \mathbf{1}_{2\bar{D}-n_{\mathbb{A}}} \\ \mathbf{0}_{n_{\mathbb{A}}} \end{bmatrix} \right\}, \quad (18)$$

over which it is active, with the notation

$$K_{(\mathbb{A})} = \begin{bmatrix} Z_{(\mathbb{A})}^\top G_D^\top \tilde{Q} G_D \\ F_D \\ Y_{\mathbb{A}} \end{bmatrix}, \quad (19a)$$

$$\kappa_{1,(\mathbb{A})} = \begin{bmatrix} -Z_{(\mathbb{A})}^\top G_D^\top \tilde{Q} c_D \\ \theta_{1,D} \\ \mathbf{1}_{n_{\mathbb{A}}} \end{bmatrix}, \kappa_{2,(\mathbb{A})} = \begin{bmatrix} -Z_{(\mathbb{A})}^\top G_D^\top \tilde{H}^\top \\ \theta_{2,D} \\ \mathbf{0}_{n_{\mathbb{A}}} \end{bmatrix}, \quad (19b)$$

$$S_{(\mathbb{A})} = -T_{(\mathbb{A})}^+ G_D^\top (\tilde{Q} G_D K_{(\mathbb{A})}^{-1} \kappa_{2,(\mathbb{A})} + \tilde{H}^\top), \quad (19c)$$

$$s_{(\mathbb{A})} = -T_{(\mathbb{A})}^+ G_D^\top \tilde{Q} G_D K_{(\mathbb{A})}^{-1} \kappa_{1,(\mathbb{A})} c_D, \quad (19d)$$

$$T_{(\mathbb{A})} = \begin{bmatrix} F_D^\top & Y_{\mathbb{A}}^\top \end{bmatrix}, Z_{(\mathbb{A})} = \text{null} \left( T_{(\mathbb{A})}^\top \right), \quad (19e)$$

where  $\mathbb{I} = \{1, \dots, 2\bar{D}\} \setminus \mathbb{A}$ ,  $n_{\mathbb{A}}$  denotes the cardinality of index set  $\mathbb{A}$ , and  $S_{\mathbb{A},2}$ ,  $s_{\mathbb{A},2}$  are the sub-matrices gathering the last  $n_{\mathbb{A}}$  rows from  $S_{(\mathbb{A})}$ ,  $s_{(\mathbb{A})}$ , corresponding to  $\mu_{\mathbb{A}}$ .

**PROOF.** As per [6, Prop. 1.30],  $\xi^*$  is a strict local minimum of (14) for the given set of active indices  $\mathbb{A}$  iff there exist the Lagrangian multipliers  $\lambda^*$ ,  $\mu^*$  such that

$$G_{\mathcal{D}}^{\top} \tilde{Q} G_{\mathcal{D}} \xi^* + G_{\mathcal{D}}^{\top} \tilde{Q} c_{\mathcal{D}} + G_{\mathcal{D}}^{\top} \tilde{H}^{\top} x_0 + F_{\mathcal{D}}^{\top} \lambda^* + Y_{\mathbb{A}}^{\top} \mu_{\mathbb{A}}^* = \mathbf{0}_{\bar{D}}, \quad (20a)$$

$$F_{\mathcal{D}} \xi^* - \theta_{1,\mathcal{D}} - \theta_{2,\mathcal{D}} x_0 = \mathbf{0}_{\bar{n}_c}, \quad (20b)$$

$$Y \xi^* - \mathbf{1}_{2\bar{D}} \leq \mathbf{0}_{2\bar{D}}, \quad (20c)$$

$$\mu_{\mathbb{A}}^* > \mathbf{0}_{n_{\mathbb{A}}}, \mu_{\mathbb{I}}^* = \mathbf{0}_{2\bar{D}-n_{\mathbb{A}}}, \quad (20d)$$

$$\mu^* \circ (Y \xi^* - \mathbf{1}_{2\bar{D}}) = \mathbf{0}_{2\bar{D}}, \quad (20e)$$

and, for every  $z \neq 0$  which satisfies  $F_{\mathcal{D}} z = 0$ ,  $Y_{\mathbb{A}} z = 0$ , we have that

$$z^{\top} G_{\mathcal{D}}^{\top} \tilde{Q} G_{\mathcal{D}} z > 0. \quad (21)$$

Since, by construction,  $\tilde{Q} \succ 0$ , a necessary and sufficient condition for (21) to hold is that  $z$  is in the null spaces of  $F_{\mathcal{D}}$  and  $Y_{\mathbb{A}}$  but not in the null space of  $G_{\mathcal{D}}$ , respectively

$$z \in \ker F_{\mathcal{D}} \cap \ker Y_{\mathbb{A}}, z \notin \ker G_{\mathcal{D}}. \quad (22)$$

By construction, the columns of matrix  $Z_{(\mathbb{A})}$  from (19e) describe a basis of the subspace (22). Left-multiplying with  $Z_{(\mathbb{A})}^{\top}$  in (20a) and reordering such that the equalities appear first leads to

$$Z_{(\mathbb{A})}^{\top} (G_{\mathcal{D}}^{\top} \tilde{Q} G_{\mathcal{D}} \xi^* + G_{\mathcal{D}}^{\top} \tilde{Q} c_{\mathcal{D}} + G_{\mathcal{D}}^{\top} \tilde{H}^{\top} x_0) = \mathbf{0}_{\bar{D}-\bar{n}_c-n_{\mathbb{A}}}, \quad (23a)$$

$$F_{\mathcal{D}} \xi^* - \theta_{1,\mathcal{D}} - \theta_{2,\mathcal{D}} x_0 = \mathbf{0}_{\bar{n}_c}, \quad (23b)$$

$$Y_{\mathbb{A}} \xi^* - \mathbf{1}_{n_{\mathbb{A}}} = \mathbf{0}_{n_{\mathbb{A}}}, \quad (23c)$$

$$\mu_{\mathbb{I}}^* = \mathbf{0}_{2\bar{D}-n_{\mathbb{A}}}, \quad (23d)$$

$$Y_{\mathbb{I}} \xi^* - \mathbf{1}_{2\bar{D}-n_{\mathbb{A}}} \leq \mathbf{0}_{2\bar{D}-n_{\mathbb{A}}}, \quad (23e)$$

$$\mu_{\mathbb{A}}^* > \mathbf{0}_{n_{\mathbb{A}}}. \quad (23f)$$

After rearranging<sup>2</sup> (23a)–(23c) to isolate  $\xi^*$  and substituting it into (20b), one obtains, using the notation of (19), both the primal and dual optimal solutions

$$K_{(\mathbb{A})} \xi^* = \kappa_{2,(\mathbb{A})} x_0 + \kappa_{1,(\mathbb{A})}, \begin{bmatrix} \lambda^* \\ \mu_{\mathbb{A}}^* \end{bmatrix} = S_{(\mathbb{A})} x_0 + s_{(\mathbb{A})}. \quad (24)$$

Mapping  $\mathbf{u}_{[0:N-1]}^*(x_0) = c_{\mathcal{D}} + G_{\mathcal{D}} \xi^*(x_0)$  yields the optimal affine law (17), while substituting (24) into (23e)–(23f) provides the description of the critical region, leading to (18) and concluding the proof.  $\square$

<sup>2</sup> The invertibility of  $K_{(\mathbb{A})} \in \mathbb{R}^{\bar{D} \times \bar{D}}$  is assumed to hold and it is one of the test conditions appearing later in Algorithm 1.

When  $\text{null}(T_{(\mathbb{A})}) = \{\mathbf{0}_{\bar{D}}\}$ , the matrix  $Z_{(\mathbb{A})} \in \mathbb{R}^{\bar{D} \times (\bar{D}-\bar{n}_c-n_{\mathbb{A}})}$  is empty and Prop. 6 reduces to the following result.

**Corollary 7** *Let  $\mathbb{A}$  be a candidate active set such that  $\bar{D} = \bar{n}_c + n_{\mathbb{A}}$ . Then, the KKT conditions are necessary and sufficient to recover from (14) the affine law (17) and the associated critical region (18), provided that the matrices  $K_{(\mathbb{A})}$ ,  $\kappa_{1,(\mathbb{A})}$ , and  $\kappa_{2,(\mathbb{A})}$  (cf. (19a)–(19b)) are redefined by retaining only their last  $\bar{n}_c + n_{\mathbb{A}}$  rows.*

**PROOF.** Applying the KKT conditions to (14) leads to

$$\begin{bmatrix} G_{\mathcal{D}}^{\top} \tilde{Q} G_{\mathcal{D}} & F_{\mathcal{D}}^{\top} & Y_{\mathbb{A}}^{\top} \\ F_{\mathcal{D}} & \mathbf{0} & \mathbf{0} \\ Y_{\mathbb{A}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \xi^* \\ \lambda^* \\ \mu_{\mathbb{A}}^* \end{bmatrix} + \begin{bmatrix} G_{\mathcal{D}}^{\top} \tilde{H}^{\top} \\ -\theta_{2,\mathcal{D}} \\ \mathbf{0}_{n_{\mathbb{A}}} \end{bmatrix} x_0 + \begin{bmatrix} G_{\mathcal{D}}^{\top} \tilde{Q} c_{\mathcal{D}} \\ -\theta_{1,\mathcal{D}} \\ -\mathbf{1}_{n_{\mathbb{A}}} \end{bmatrix} = \mathbf{0}_{\bar{D}+\bar{n}_c+n_{\mathbb{A}}}. \quad (25)$$

Solving for the primal solution  $\xi^*$  and the dual solutions  $\lambda^*$ ,  $\mu_{\mathbb{A}}^*$  shows, by inspection, that they coincide with the form (24) if and only if  $K_{(\mathbb{A})}$ ,  $\kappa_{1,(\mathbb{A})}$ , and  $\kappa_{2,(\mathbb{A})}$  (cf. (19a)–(19b)) are redefined to retain only their last  $\bar{n}_c + n_{\mathbb{A}}$  rows. Carrying the remaining steps in Prop. 6 to arrive at (17)–(18) concludes the proof.  $\square$

#### 4.2 Iterative updates

In (19), the matrices  $Z_{(\mathbb{A})}$ ,  $T_{(\mathbb{A})}^+$ , and  $K_{(\mathbb{A})}^{-1}$  need not be recomputed from scratch when the active set  $\mathbb{A}$  is updated. We next analyze the effect of an update  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$  on each of these matrices.

**Lemma 8** *Whenever  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$  holds, we have*

$$Z_{(\bar{\mathbb{A}})} = Z_{(\mathbb{A})} \text{null}(Y_i Z_{(\mathbb{A})}). \quad (26)$$

**PROOF.** From  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$ , we have

$$Z_{(\bar{\mathbb{A}})} = \text{null} \left( \begin{bmatrix} F_{\mathcal{D}}^{\top} & Y_{\bar{\mathbb{A}}}^{\top} \end{bmatrix}^{\top} \right) = \text{null} \left( \begin{bmatrix} F_{\mathcal{D}}^{\top} & Y_{\mathbb{A}}^{\top} & Y_i^{\top} \end{bmatrix}^{\top} \right).$$

Since  $Z_{(\mathbb{A})} = \text{null} \left( \begin{bmatrix} F_{\mathcal{D}}^{\top} & Y_{\mathbb{A}}^{\top} \end{bmatrix}^{\top} \right)$ , any  $z \in Z_{(\bar{\mathbb{A}})}$  satisfies: i)  $F_{\mathcal{D}} z = \mathbf{0}$ , ii)  $Y_{\mathbb{A}} z = \mathbf{0}$ , iii)  $Y_i z = \mathbf{0}$ . From (i) and (ii), it follows that  $z \in Z_{(\mathbb{A})}$ , so there exists  $v$  with  $z = Z_{(\mathbb{A})} v$ . Substituting into (iii) yields the additional condition  $Y_i Z_{(\mathbb{A})} v = \mathbf{0}$ , i.e.,  $v \in \text{null}(Y_i Z_{(\mathbb{A})})$ . Hence,  $Z_{(\bar{\mathbb{A}})} = Z_{(\mathbb{A})} \text{null}(Y_i Z_{(\mathbb{A})})$ , as in (26).  $\square$

$$Z_{(\bar{\mathbb{A}})}^\top = \begin{bmatrix} I_{k_1} & \begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix} & \\ & & I_{k_2} \end{bmatrix} \cdot \begin{bmatrix} Z_{(\bar{\mathbb{A}}),1}^\top & \\ & \begin{smallmatrix} Z_{(\bar{\mathbb{A}}),j}^\top \\ Z_{(\bar{\mathbb{A}}),2}^\top \end{smallmatrix} & \\ & & F_D \\ & & Y_{\bar{\mathbb{A}},1} \\ & & Y_{\bar{\mathbb{A}},2} \end{bmatrix}; \quad K_{(\bar{\mathbb{A}})} = \begin{bmatrix} Z_{(\bar{\mathbb{A}}),1}^\top & \\ & \begin{smallmatrix} Z_{(\bar{\mathbb{A}}),j}^\top \\ Z_{(\bar{\mathbb{A}}),2}^\top \end{smallmatrix} & \\ & & F_D \\ & & Y_{\bar{\mathbb{A}},1} \\ & & Y_{\bar{\mathbb{A}},2} \end{bmatrix} \cdot G_D^\top \tilde{Q} G_D \xrightarrow{\bar{\mathbb{A}}' = \bar{\mathbb{A}} \cup \{i\}} K_{(\bar{\mathbb{A}}')} = \begin{bmatrix} Z_{(\bar{\mathbb{A}}),1}^\top + v_1 Z_{(\bar{\mathbb{A}}),j}^\top & \\ v_2 Z_{(\bar{\mathbb{A}}),j}^\top + Z_{(\bar{\mathbb{A}}),2}^\top & \\ & F_D \\ & Y_{\bar{\mathbb{A}},1} \\ & \begin{smallmatrix} Y_i \\ Y_{\bar{\mathbb{A}},2} \end{smallmatrix} \end{bmatrix} \cdot G_D^\top \tilde{Q} G_D$$

Fig. 2. Illustration of the update of matrices  $Z_{(\bar{\mathbb{A}})}$ ,  $K_{(\bar{\mathbb{A}})}$ .

While  $\text{null}(Y_i Z_{(\bar{\mathbb{A}})})$  can be computed by standard methods (e.g., QR factorization), updating  $K_{(\bar{\mathbb{A}})}^{-1}$  is more efficient when a sparse representation is employed. The next result provides such a representation.

**Lemma 9** *Let  $z \in \mathbb{R}^n$  be a nonzero row vector and choose an index  $j$  with  $z_j \neq 0$ . Define the  $n \times (n-1)$  matrix  $V$  whose columns are*

$$V_{ik} = \begin{cases} 1, & i = \sigma(k), \\ -\frac{z_{\sigma(k)}}{z_j}, & i = j, \\ 0, & \text{otherwise,} \end{cases}$$

where  $\sigma(k) = k$ , for  $k < j$ , and  $\sigma(k) = k+1$ , for  $k \geq j$ . Then, the columns of  $V$  form a basis of  $\text{null}(z)$ .

**PROOF.** Let  $v^{(k)} = V e_k$ . By the definition of  $V$ ,  $v_{\sigma(k)}^{(k)} = 1$ ,  $v_j^{(k)} = -\frac{z_{\sigma(k)}}{z_j}$ , and  $v_i^{(k)} = 0$ . Hence,  $z v^{(k)} = z_{\sigma(k)} + z_j \left(-\frac{z_{\sigma(k)}}{z_j}\right) = 0$ , so every column  $v^{(k)} \in \text{null}(z)$ . Thus,  $\text{im}(V) \subseteq \text{null}(z)$ . Deleting the  $j$ -th row of  $V$  results in a matrix whose columns are vectors of the canonical base of  $\mathbb{R}^{n-1}$ , which are linearly independent. Since  $z \neq 0$ ,  $\text{rank}(z) = 1$ , and, by rank-nullity,  $\dim(\text{null}(z)) = n-1$ . We have  $n-1$  independent vectors in  $\text{null}(z)$ , so they form a basis of  $\text{null}(z)$ .  $\square$

We have now the prerequisite to compute the update of  $K_{(\bar{\mathbb{A}})}$  and of its inverse,  $K_{(\bar{\mathbb{A}})}^{-1}$ .

**Proposition 10** *Whenever  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$  holds, we have*

$$K_{(\bar{\mathbb{A}})}^{-1} = [K_{(\mathbb{A})}^{-1} - K_{(\mathbb{A})}^{-1} U (I_2 + W K_{(\mathbb{A})}^{-1} U)^{-1} W K_{(\mathbb{A})}^{-1}] P(j, \bar{n}_c + i)^\top, \quad (27)$$

where

$$U = \begin{bmatrix} \vdots & v_1 \\ e_j & -1 \\ \vdots & v_2 \end{bmatrix}, \quad W = \begin{bmatrix} Y_i \\ Z_{(\mathbb{A}),j}^\top G_D^\top \tilde{Q} G_D \end{bmatrix}, \quad (28)$$

and with  $v = \begin{bmatrix} v_1^\top & v_2^\top \end{bmatrix}^\top$ , the  $j$ -th row from matrix  $V$ , chosen such that the null matrix update

$$Z_{(\bar{\mathbb{A}})} = Z_{(\mathbb{A})} V = Z_{(\mathbb{A})} \begin{bmatrix} I_{k_1} & \mathbf{0} \\ -\frac{v_1^\top}{z_j} & -\frac{v_2^\top}{z_j} \\ \mathbf{0} & I_{k_2} \end{bmatrix} \quad (29)$$

holds;  $P(j, \bar{n}_c + i)$  extracts the  $j$ -th row and inserts it at the  $(\bar{n}_c + i)$ -th position, shifting all subsequent rows, as needed.

**PROOF.** Computing  $V$  as in Lemma 9 for  $z = Y_i Z_{(\mathbb{A})}$  and applying Lemma 8 yields the expression of  $Z_{(\bar{\mathbb{A}})}$  in (29). The update from  $K_{(\mathbb{A})}$  to  $K_{(\bar{\mathbb{A}})}$  is twofold: (i) the  $j$ -th row is eliminated by a linear combination of the rows of  $Z_{(\mathbb{A})}^\top$  with weightings given by  $v$ , and (ii) a new row  $Y_i$  is inserted at the position  $\bar{n}_c + i$ . This can be modeled by first updating the  $j$ -th row and then moving it to the position  $\bar{n}_c + i$ . With the notation of (28), the update reads

$$K_{(\bar{\mathbb{A}})} = P(j, \bar{n}_c + i) (K_{(\mathbb{A})} + U W). \quad (30)$$

Then, noting that  $P(j, \bar{n}_c + i)^{-1} = P(j, \bar{n}_c + i)^\top$  and applying the Woodbury matrix identity gives (27), which concludes the proof.  $\square$

For clarity, Fig. 2 illustrates the updates  $Z_{(\mathbb{A})} \rightarrow Z_{(\bar{\mathbb{A}})}$  and  $K_{(\mathbb{A})} \rightarrow K_{(\bar{\mathbb{A}})}$ . Patterned rectangles mark the rows modified during the update, while indices 1, 2 on matrices  $Z_{(\mathbb{A})}^\top$ ,  $Y_{\bar{\mathbb{A}}}$ , and the vector  $v$  denote the row blocks between which a single row is deleted or inserted.

**Remark 11** *The invertibility of  $K_{(\bar{\mathbb{A}})}$  is determined by the factor  $(I_2 + W K_{(\mathbb{A})}^{-1} U)$  in (27). Since this matrix lies in  $\mathbb{R}^{2 \times 2}$ , the singularity test is both efficient and numerically reliable.*

**Remark 12** Whenever  $\bar{\mathbb{A}} = \mathbb{A} \cup i$  holds, we have [5]

$$T_{(\bar{\mathbb{A}})}^+ = S_k^\top \begin{bmatrix} T_{(\mathbb{A})}^+ - db^\top \\ b^\top \end{bmatrix}, \quad (31)$$

where  $d = T_{(\mathbb{A})}^+ Y_i^\top$ ,  $c = (Y_i^\top - T_{(\mathbb{A})} d)$ , and  $b$  is a vector

$$b = \begin{cases} \frac{c}{c^\top c}, & \sqrt{c^\top c} > \varepsilon, \\ \frac{T_{(\mathbb{A})}^+ d}{1 + d^\top d}, & \text{otherwise.} \end{cases}$$

The term  $S_k \in \mathbb{R}^{(n+1) \times (n+1)}$  is a column permutation that moves the last column to position  $k$ , i.e.,  $S_k = \left( \sum_{j=1}^{k-1} e_j e_j^\top + e_{n+1} e_k^\top + \sum_{j=k}^n e_j e_{j+1}^\top \right)$ . We take  $k = n_{F_D} + |\{\ell \in \bar{\mathbb{A}} : \ell \leq i\}|$ ; lastly,  $\varepsilon$  is a tolerance.

As shown in [13], the active sets defining non-empty critical regions are embedded in the face lattice of the polyhedral feasible domain of the MPC problem. Here, owing to the simpler structure, we employ a lifted formulation in  $\mathbb{R}^D$  with variable  $\xi$ . According to [11, Prop. 18], there exists a surjective relation whereby each face of the feasible domain corresponds to a face in the lifted space (though not all faces of  $\xi$  map to faces of  $u$ ). We exploit this fact to propose a fast necessary check for the non-emptiness of a candidate set  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$ .

**Proposition 13** Let  $\mathbb{A}$  be an active set such that  $CR_{(\mathbb{A})}$  is non-empty. Define the corresponding active set of polyhedral constraints  $\mathbb{A}_{\text{red}} \subset \{1, \dots, q\}$  as

$$\mathbb{A}_{\text{red}} = \{j : A_j \mathbf{u}_{[0:N-1]}_{(\mathbb{A})} = b_j + F_j x_0\}. \quad (32)$$

Then, for a candidate set  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$ , the following implication holds

$$(\exists j \text{ s.t. } \bar{\mathbb{A}}_{\text{red}} = \mathbb{A}_{\text{red}} \cup \{j\}) \implies (CR_{(\bar{\mathbb{A}})} = \emptyset). \quad (33)$$

**PROOF.** For a candidate active set  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$ , we compute  $\bar{\mathbb{A}}_{\text{red}}$  as in (32). If there is no index  $j$  such that  $\bar{\mathbb{A}}_{\text{red}} = \mathbb{A}_{\text{red}} \cup \{j\}$ , the candidate solution does not lie on a child face of the polyhedral representation. Since, under mild assumptions<sup>3</sup>, the explicit solution is continuous and corresponds to a single index update in the active set, implication (33) follows, concluding the proof.  $\square$

<sup>3</sup> We assume LICQ, which can be ensured via slight perturbations of the original constraints.

## 5 Explicit MPC algorithm

We introduce Algorithm 1 to compute the solution of explicit MPC, adapting to the particular structure and properties of constrained zonotopes. The algorithm follows similar reasoning to that introduced in the previous work for the general polyhedral case (see [13]).

---

### Algorithm 1 Explicit MPC solution in the CZ case

---

**Require:** The cost matrices  $\tilde{Q}$  and  $\tilde{H}$ ; the constrained zonotope matrices in (23); the matrix  $Y$  of the hypercube representation in (14c).

**Ensure:** The graph  $\mathcal{L}'$  containing the solution

```

1: Initialize  $\mathcal{L}'$  with a node containing  $\emptyset$ , denoted as  $n_\emptyset$ 
2: Initialize a queue  $\mathcal{Q}$  with  $\emptyset$ 
3: while  $\mathcal{Q}$  is not empty do
4:   Extract active set  $\mathbb{H}$  from the queue  $\mathcal{Q}$ 
5:   Compute  $\mathbb{S}_{(\mathbb{H})}$  as in (34)
6:   Initialize  $\mathcal{G} \leftarrow \{\emptyset\}$ 
7:   for each  $i$  in  $\mathbb{S}_{(\mathbb{H})}$  do
8:     Let  $h' \leftarrow \mathbb{H}$ 
9:     Make  $h'(i) \leftarrow 0$ ;  $h'(i + D) \leftarrow 1$ 
10:     $\mathcal{G} \leftarrow \{\mathcal{G}, h'\}$ 
11:    Make  $h'(i) \leftarrow 1$ ;  $h'(i + D) \leftarrow 0$ 
12:     $\mathcal{G} \leftarrow \{\mathcal{G}, h'\}$ 
13:   Let  $n^{\mathbb{H}}$  be a node for  $\mathbb{H}$  in  $\mathcal{L}'$ 
14:   for each candidate set  $\mathbb{A} \in \mathcal{G}$  do
15:     Find or create the node  $n^{\mathbb{A}}$  for  $\mathbb{A}$  in  $\mathcal{L}'$ 
16:     if  $n^{\mathbb{A}}$  is newly created then
17:       Compute the critical region using Alg. 2
18:        $(L_{(\mathbb{A})}, \ell_{(\mathbb{A})}, \text{flag}) \leftarrow \text{getCriticalRegion}(\mathbb{A})$ 
19:       if  $\text{flag}$  is false then
20:         go to Step 14 [NUMERICAL ISSUES]
21:       if implication (33) does not hold then
22:         go to Step 14 [QUICK CHECK]
23:       if the polyhedral set  $P(L_{(\mathbb{A})}, \ell_{(\mathbb{A})})$  is empty then
24:         go to Step 14 [CHEBISHEV RADIUS]
25:       Add the active set  $\mathbb{A}$  to the queue  $\mathcal{Q}$ 
26:       Add the edge  $(n^{\mathbb{H}}, n^{\mathbb{A}})$  to the graph  $\mathcal{L}'$ 
27:       Add the edge  $(n^{\mathbb{H}}, n^{\mathbb{A}})$  to  $\mathcal{L}'$  if not already added
28: return the solution graph  $\mathcal{L}'$ 

```

---

Alg. 1 enumerates nodes from the face lattice of the feasible domain (16), each corresponding to a non-empty critical region. The resulting structure is a tree in which every node is defined by its active set, together with the associated primal/dual solution and critical region (plus auxiliary data required for internal updates, e.g.,  $Z_{(\mathbb{A})}$ ,  $K_{(\mathbb{A})}^{-1}$ , and  $T_{(\mathbb{A})}^+$ ). The edges are indexed by the constraints which become active in the transition from parent to child.

Although the procedure is conceptually simple, its efficiency depends critically on three aspects: i) generation of candidate active sets for a given parent node; ii) reuse of previously computed information in determining the critical region; and, iii) a hierarchy of emptiness checks

and stopping conditions to avoid unnecessary computations. These aspects are discussed in detail next.

First, since active sets are in bijection with the faces of the feasible domain, each potential child of a parent node ( $\mathbb{H}$  from step 4) is either a neighboring face of a higher order or the empty set. In the polyhedral case, this enumeration requires facet–vertex operations [13], which are computationally burdensome. In contrast, the constrained zonotope description of (16) enables explicit enumeration in steps 7–12.

Assume that the inequalities  $|\xi| \leq \mathbf{1}_{\bar{D}}$  in (16) are ordered such that the  $i$ -th inequality corresponds to  $\xi_i \leq 1$  and the  $(i+D)$ -th to  $-\xi_i \leq 1$ . We then extract the indices for which neither<sup>4</sup> constraint in the pair  $(i, i+D)$  is active

$$\mathbb{S}_{(\mathbb{A})} = \{i \in \{1, \dots, D\} : a_i \vee a_{i+D} = 0, i \in \mathbb{A}\}. \quad (34)$$

Here, the active set  $\mathbb{A} = \{a_1, a_2, \dots, a_{2D}\}$  is encoded with logical values, as  $a_i \in \{0, 1\}$ , where  $a_i = 1$  if the  $i$ -th constraint is active. The exclusive OR is defined as

$$x \vee y = \begin{cases} 1, & x \neq y, \\ 0, & x = y. \end{cases} \quad (35)$$

Enumerating over each  $i \in \mathbb{S}_{(\mathbb{A})}$  and activating either the  $i$ -th or  $(i+D)$ -th constraint, as done in steps 7–12, generates all child facets of the face described by  $\mathbb{A}$ .

Secondly, Algorithm 2 details the computation of the critical region in the constrained-zonotope case. For brevity, we only highlight the improvements over [21], namely the iterative construction of the matrices  $Z_{(\bar{\mathbb{A}})}$ ,  $K_{(\bar{\mathbb{A}})}$ , and  $T_{(\bar{\mathbb{A}})}^+$  (steps 4, 13 and 8).

Thirdly, we clarify the stopping conditions in Algorithm 1. Since the full emptiness test is computationally expensive, we adopt a hierarchical approach. If Algorithm 2 returns **false** (e.g., due to numerical issues or lack of an inverse), we stop early at step 19. If the necessary condition in Prop. 13 fails at step 21, we also terminate. Only at step 23 we do perform the full emptiness test. Among the available strategies, our practical implementation computes the Chebyshev radius [8] of the candidate region and applies a threshold-based rule to discard regions with small radius.

## 6 Numerical simulation

This section presents a comparison of the algorithms introduced in the previous sections. For clarity, we label the results as follows: Algorithm 1 is labeled **CZ**, and

<sup>4</sup> Both constraints cannot be simultaneously active, since this would require  $\xi_i = 1$  and  $\xi_i = -1$ .

---

**Algorithm 2** (`getCriticalRegion`) Compute the critical region with CZ using iterative updates

---

**Require:** The current active set  $\bar{\mathbb{A}}$  of the form  $\bar{\mathbb{A}} = \mathbb{A} \cup \{i\}$ , a tolerance  $\varepsilon$

**Ensure:** Critical region  $(L_{(\bar{\mathbb{A}})}, \ell_{(\bar{\mathbb{A}})})$  and a flag *success*

- 1: Let  $D \leftarrow$  number of columns in  $G_D$
- 2: Initialize  $L_{(\bar{\mathbb{A}})} \leftarrow \emptyset$ ,  $\ell_{(\bar{\mathbb{A}})} \leftarrow \emptyset$ , *success*  $\leftarrow$  **true**
- 3: Retrieve  $Z_{(\mathbb{A})}$ ,  $K_{(\mathbb{A})}^{-1}$ , and  $T_{(\mathbb{A})}^+$  from memory
- 4: Compute  $Z_{(\bar{\mathbb{A}})} = Z_{(\mathbb{A})} \text{null}(Y_i Z_{(\mathbb{A})})$  as in Lemma 8
- 5: **if**  $Z_{(\bar{\mathbb{A}})}^\top G_D^\top Q G_D Z_{(\bar{\mathbb{A}})}$  is not positive definite **then**
- 6:     *success*  $\leftarrow$  **false** [NUMERICAL INFEASIBILITY]
- 7:     **return**  $(L_{(\bar{\mathbb{A}})}, \ell_{(\bar{\mathbb{A}})}, \text{success})$
- 8: Let  $k \leftarrow \mathbb{A} \vee \bar{\mathbb{A}}$  and compute  $T_{(\bar{\mathbb{A}})}^+$  as in (31)
- 9: Determine matrices  $U$ ,  $V$  as in (28)
- 10: **if**  $\min_{\lambda \in \Lambda(I_2 + V K_{(\bar{\mathbb{A}})}^{-1} U)} |\lambda| \leq \varepsilon$  **then**
- 11:     *success*  $\leftarrow$  **false** [SINGULAR MATRIX DETECTED]
- 12:     **return**  $(L_{(\bar{\mathbb{A}})}, \ell_{(\bar{\mathbb{A}})}, \text{success})$
- 13: Compute  $K_{(\bar{\mathbb{A}})}^{-1}$  as in Prop. 10
- 14: Store  $Z_{(\bar{\mathbb{A}})}$ ,  $K_{(\bar{\mathbb{A}})}^{-1}$ , and  $T_{(\bar{\mathbb{A}})}^+$  in memory
- 15: Compute  $S_{(\bar{\mathbb{A}})}$  and  $s_{(\bar{\mathbb{A}})}$  with (19c), (19d)
- 16: Extract  $L_{(\bar{\mathbb{A}})}$  and  $\ell_{(\bar{\mathbb{A}})}$  using (18)
- 17: **return**  $(L_{(\bar{\mathbb{A}})}, \ell_{(\bar{\mathbb{A}})}, \text{success})$

---

the improvements described in Section 4.2 are labeled **CZ (Iterative)** and **CZ (Iterative | Quick)**. We test the methods on an example with the state vector in  $\mathbb{R}^4$  given as

$$A = \begin{bmatrix} 0.84 & 0.02 & 0.00 & -0.02 \\ 0.01 & 0.88 & -0.03 & -0.02 \\ 0.04 & -0.08 & 0.86 & -0.02 \\ -0.09 & -0.02 & -0.02 & 0.88 \end{bmatrix}, B = \begin{bmatrix} -0.10 & 0.03 \\ -0.07 & -0.07 \\ -0.27 & 0.13 \\ 0.01 & -0.00 \end{bmatrix}. \quad (36)$$

The associated cost matrices are  $S = Q = I_4$  and  $R = 0.1 \cdot I_2$ , with state and input constraints  $|x_k| \leq 10 \cdot \mathbf{1}_n$  and  $|u_k| \leq \mathbf{1}_m$ . All algorithms are run in MATLAB R2024b on a laptop with AMD Ryzen 7 and 16 GB of RAM, on the Windows 11 operating system.

Applying Alg. 1 to the dynamics (36) with prediction horizon  $N = 4$  yields the tree structure shown in Fig. 3. Each edge is labeled with the index of the constraint that becomes active, so the active set of any node can be obtained by tracing a path from the root (empty set) to that node (two examples are highlighted). Although multiple paths may exist depending on the order in which indices are appended, the result is a tree, since Alg. 1 terminates the search once a link is established.

Figure 4 illustrates the computation time of each variant of the algorithm with respect to the prediction hori-

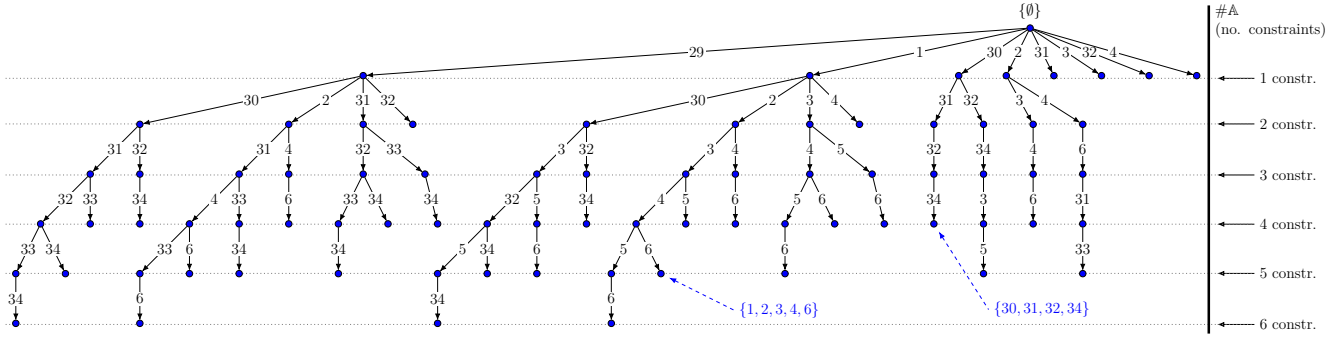


Fig. 3. Tree structure obtained from Alg. 1, applied to dynamics (36), for prediction horizon  $N = 4$ .

zon. We compare our algorithm with the state-of-the-art solvers, namely YALMIP and MPT3. Several trends are clear. First, for the short to mid horizons (where  $N \leq 8$ ), both iterative variants of our algorithm are consistently faster than YALMIP and MPT3. The iterative quick variant is the best performer. Second, the crossover region (where  $N \approx 10$ ) reveals that the performance converges. The iterative quick variant is roughly on par with the state-of-the-art solvers, whereas the iterative non-quick becomes modestly slower. Third, for long horizons, we notice that YALMIP and MPT3 scale more favorably and overtake our methods. The baseline CZ (non-iterative) degrades the fastest. The iterative quick variant remains the best of our three, but it is slower than YALMIP and MPT3.

Considering the scaling behavior, all curves are approximately straight lines on a log-time axis, indicating exponential growth with  $N$ . Among our variants, the iterative quick method reduces the slope relative to the plain iterative one, while the non-iterative CZ has the steepest slope. Conversely, YALMIP and MPT3 exhibit nearly identical scaling across the entire range. Notably, our code is implemented in MATLAB without using compiled code (e.g., through mex calls). However, we expect that significant reductions in computation time can be obtained with a C++ compiled version of the code.

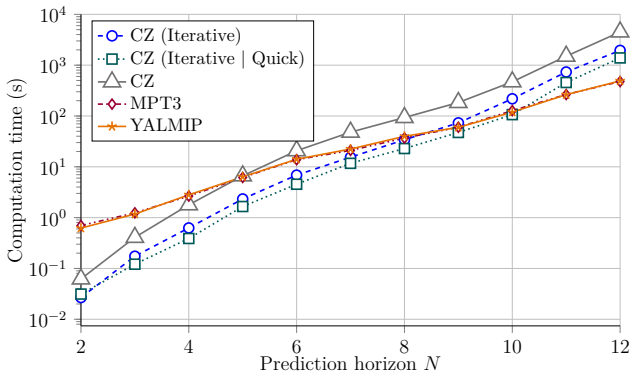


Fig. 4. Computation time vs. prediction horizon  $N$ .

It is equally important to analyze the time performance

while considering the number of critical regions discovered by the algorithms, since generating more solutions leads to an increase in computation time. Differences in the number of regions arise from tolerances, mainly involved in deciding whether a polyhedral set is empty or not. From Figure 5, we conclude that our algorithm and its variants produce the same number of critical regions when compared to the state-of-the-art solvers, except for the case  $N = 12$ , where YALMIP and MPT3 produce slightly more regions in this particular example. This is a clear indicator that our methods work as intended and are comparable to the state-of-the-art solvers.

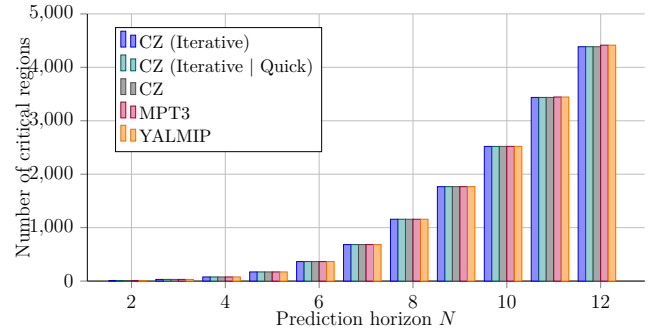


Fig. 5. Number of critical regions vs. prediction horizon  $N$ .

Lastly, Figure 6 provides insights about the behavior of our methods. We have chosen to illustrate only the iterative quick variant, since it considers all stopping criteria discussed in Alg. 2.

The figure shows at which points the algorithm exits: when it encounters a numerical tolerance violation (blue), when the quick check is guaranteed (teal), when the region is empty (yellow), or when a new critical region is discovered (purple). As it can be observed, the quick exit condition makes a significant contribution, which justifies the increased performance of this variant. The numerical tolerance cases are also significant; these are cases in which we either have numerical infeasibility (see Step 6 in Algorithm 2) or a singular matrix has been detected (see Step 11 of the same algorithm).

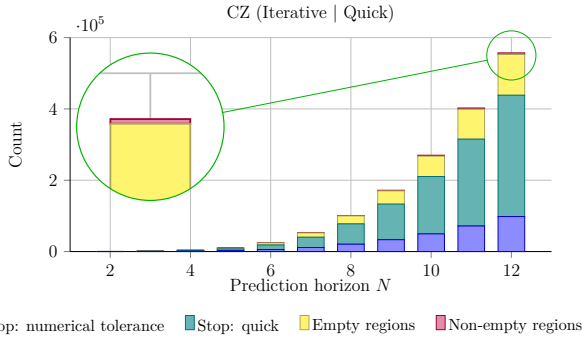


Fig. 6. Count of the exit conditions vs. prediction horizon  $N$  for the iterative quick variant.

## 7 Conclusions

This paper presented a reinterpretation of explicit MPC in which the feasible domain is represented as a constrained zonotope. The multi-parametric problem was formulated in the lifted generator space and solved using second-order optimality conditions. Computational efficiency was enhanced through low-rank matrix updates and an analytic enumeration of candidate active sets naturally yielded the explicit solution in tree form. Numerical results demonstrated improved performance over conventional polyhedral formulations both in computation time and scalability.

The code used to generate the results is available at: [www.gitlab.com/msstefan/constrained-zonotope-empc/-/tree/paper-1-iterative-updates](https://www.gitlab.com/msstefan/constrained-zonotope-empc/-/tree/paper-1-iterative-updates).

## References

- [1] Parisa Ahmadi-Moshkenani, Tor Arne Johansen, and Sorin Olaru. Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions. *IEEE Transactions on Automatic Control*, 63(10):3221–3231, 2018.
- [2] Alessandro Alessio and Alberto Bemporad. A survey on explicit model predictive control. In *Nonlinear model predictive control*, pages 345–369. Springer, 2009.
- [3] Matthias Althoff, Olaf Stursberg, and Martin Buss. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear analysis: hybrid systems*, 4(2):233–249, 2010.
- [4] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
- [5] Adi Ben-Israel and Thomas N.E. Greville. *Generalized inverses: Theory and applications*. Springer, 2nd edition, 2003.
- [6] Dimitri P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.
- [7] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [8] Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [9] Komei Fukuda. Polyhedral computation. 2020. Publisher: Department of Mathematics, Institute of Theoretical Computer Science ETH Zurich.
- [10] Bogdan Gheorghe, Daniel-Mihail Ioan, Florin Stoican, and Ionela Prodan. Computing the maximal positive invariant set for the constrained zonotopic case. *IEEE Control Systems Letters*, 8:1481–1486, 2024.
- [11] Colin N. Jones, Eric C. Kerrigan, and Jan M. Maciejowski. Equality set projection: A new algorithm for the projection of polytopes in halfspace representation. 2004.
- [12] William S. Levine, Lars Grüne, Rafal Goebel, Saša V. Rakovic, Ali Mesbah, Ilya Kolmanovsky, Stefano Di Cairano, Douglas A. Allan, James B. Rawlings, and Martin A. Seif. Handbook of model predictive control. 2018.
- [13] Ștefan S. Mihai, Florin Stoican, and Bogdan D. Ciubotaru. Explicit MPC solution using Hasse diagrams: Construction, storage and retrieval. In *International Conference on Difference Equations and Applications*, pages 353–369. Springer, 2022.
- [14] Ștefan S. Mihai, Florin Stoican, and Bogdan D. Ciubotaru. On the link between explicit MPC and the face lattice of the lifted feasible domain. *IFAC-PapersOnLine*, 55(16):308–313, 2022.
- [15] Martin Mönnigmann. On the structure of the set of active sets in constrained linear quadratic regulation. *Automatica*, 106:61–69, 2019.
- [16] Richard Oberdieck, Nikolaos A. Dangelakis, Maria M. Papathanasiou, Ioana Nascu, and Efstratios N. Pistikopoulos. Pop-parametric optimization toolbox. *Industrial & Eng. Chemistry Research*, 55(33):8979–8991, 2016. Publisher: ACS Publications.
- [17] Juraj Oravec and Martin Klaučo. Real-time tunable approximated explicit MPC. *Automatica*, 142:110315, 2022.
- [18] Vignesh Raghuraman and Justin P. Koeln. Set operations and order reductions for constrained zonotopes. *Automatica*, 139:110204, 2022.
- [19] Joseph K. Scott, Davide M. Raimondo, Giuseppe Roberto Marcegaglia, and Richard D. Braatz. Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69:126–136, 2016.
- [20] Maria M. Seron, Graham C. Goodwin, and José A. De Doná. Characterisation of receding horizon control for constrained linear systems. *Asian Journal of Control*, 5(2):271–286, 2003.
- [21] Florin Stoican, Ștefan S. Mihai, Martin Mönnigmann, and Bogdan D. Ciubotaru. Computing the explicit MPC solution in the constrained zonotope case. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 1055–1060. IEEE, 2024.
- [22] Abraham P. Vinod, Avishai Weiss, and Stefano Di Cairano. Projection-free computation of robust controllable sets with constrained zonotopes. *Automatica*, 175:112211, 2025.
- [23] Songlin Yang, Sorin Olaru, and Pedro Rodriguez-Ayerbe. Lifting-based PWA control: A global solution to handle disturbances in the state space partitions. *Automatica*, 182:112529, 2025.
- [24] Zhao Zhang, Xiao He, and Donghua Zhou. Active fault diagnosis for LPV systems based on constrained zonotopes. *IEEE Transactions on Automatic Control*, 69(11):7893–7900, 2024.