2023 24th International Conference on Process Control (PC)
June 6–9, 2023, Štrbské Pleso, Slovakia

Exploiting symmetries in tree-based combinatorial calculation of explicit linear MPC solutions

1st Ruth Mitze
Dep. of Mechanical Engineering
Ruhr-Universität Bochum
Bochum, Germany
ruth.mitze@rub.de

2nd Martin Mönnigmann
Dep. of Mechanical Engineering
Ruhr-Universität Bochum
Bochum, Germany
martin.moennigmann@rub.de

***Abstract*—Symmetries of linear MPC problems are reflected in symmetries of their explicit solutions. We recently showed these symmetries also appear in the set of the active sets that define the explicit solution. Consequently, symmetries can be used to speed up algorithms for the calculation of the set of active sets. In this paper, we exploit symmetries to accelerate the approach from [1] by improving the exploration of the combinatorial tree of active sets. The reductions of the computational effort that can be achieved are illustrated and analyzed with two examples.**

***Index Terms*—Explicit linear model predictive control, combinatorial parametric quadratic programming, symmetric constrained optimization**

I. Introduction

Explicit linear model predictive control (MPC) requires the parametric solution to a constrained linear-quadratic optimal control problem (OCP), which is known to be continuous piecewise-affine feedback law [2], [3]. Particularly for high-order problems, for problems with many constraints, and for problems with long prediction horizons, calculating the solution is computationally demanding. Reducing the computational effort of calculating the solution is the focus of an entire field of research. The first approaches proposed for this task exploit geometric relations of the affine pieces in the solution [2], [4], [5]. These approaches are competitive and implemented in mature software toolboxes [6], [7]. However, they require proper tuning to prevent missing out on small pieces of the piecewise-affine feedback law. An alternative approach was proposed by [1] and calculates the set of active sets that define an affine piece in the solution. The approach proceeds by enumerating possible active set candidates and examining them for being an element of the solution. Several refinements to the approach from [1] have been proposed. They exploit relations among the elements in the set of active sets for a more targeted selection and examination of the active set candidates, thus reducing the computational complexity [8]–[12]. Approaches that calculate the solution by enumerating and examining active set candidates are known as combinatorial approaches or implicit enumeration techniques. Combinatorial approaches often order the active sets that are candidates for the solution in a combinatorial tree.

Symmetries of a physical system and its constraints often result in symmetries of the associated constrained linear-quadratic OCP. Symmetries of the OCP are then reflected in input- and state-space transformations in the associated piecewise-affine feedback law [13]. For problems with constraints that are point-symmetric to the origin, symmetries were identified in the active sets [14] and used to improve the combinatorial approaches from [1] and [15] in [14] and [16], respectively. For problems with general symmetries, symmetries in the active sets were identified only recently and a strategy to improve combinatorial approaches for symmetric OCPs was proposed [17].

In this work, we use the strategy proposed in [17] to improve the combinatorial approach from [1]. This approach uses a combinatorial tree to enumerate the active sets that are candidates for the solution. Also, we propose a change to the strategy that makes the examination more efficient. This change further reduces the computational effort, for problems with many symmetries in particular. Reductions of the computational effort are analyzed by applying the approach to different examples.

Section II introduces constrained linear-quadratic OCPs and the combinatorial approach from [1]. Section III defines symmetries for OCPs and introduces the exploration strategy to improve combinatorial approaches proposed by [17]. Section IV presents two approaches similar to [1] but improved for symmetric OCPs. The new approaches are applied to two examples in Sect. V and the computational effort is analyzed. Finally, Sect. VI concludes this paper.

Notation: Let $I^a \in \mathbb{R}^{a\times a}$ denote the identity matrix and $1^a \in \mathbb{R}^a$ a column vector of ones. Furthermore, let $\mathcal{P}^c(\mathcal{M}) = \{m \in \mathcal{P}(\mathcal{M}) \,|\, |m| \leq c\}$, where $\mathcal{P}(\mathcal{M})$ refers to the power set of set $\mathcal{M}$. For a matrix $A \in \mathbb{R}^{m\times n}$ and ordered set $\mathcal{M} \subseteq \{1, ..., a\}$, we denote by $A_{\mathcal{M}} \in \mathbb{R}^{|\mathcal{M}|\times n}$ the submatrix of A containing all rows indicated by $\mathcal{M}$. Let operators $\otimes$ and $\times$ denote the Kronecker and Cartesian product, respectively, and let $A \circ \mathcal{M} = \{Am \,|\, m \in \mathcal{M}\}$. Furthermore, let $\max(\mathcal{M})$ and $\min(\mathcal{M})$ denote the greatest and smallest element of the set $\mathcal{M}$, respectively, where applicable. We say sets $\mathcal{M}$ and $\mathcal{N}$ partition $\mathcal{R}$ if $\mathcal{M} \cup \mathcal{N} = \mathcal{R}$ and $\mathcal{M} \cap \mathcal{N} = \emptyset$.

Support by the European Commission under grant no. 101079342 (Fostering Opportunities Towards Slovak Excellence in Advanced Control for Smart Industries) is gratefully acknowledged.

II. Problem statement and preliminaries

This paper treats finite-horizon constrained linear-quadratic OCPs of the form

$$\begin{aligned} \min_{U,X} \quad & \|x(N)\|_P^2 + \sum_{k=0}^{N-1} \left(\|x(k)\|_Q^2 + \|u(k)\|_R^2 \right) && \text{(1a)} \\ \text{s.t.} \quad & x(k+1) = Ax(k) + Bu(k),\ k = 0, ..., N-1 && \text{(1b)} \\ & u(k) \in \mathcal{U},\ k = 0, ..., N-1 && \text{(1c)} \\ & x(k) \in \mathcal{X},\ k = 0, ..., N-1 && \text{(1d)} \\ & x(N) \in \mathcal{T}, && \text{(1e)} \end{aligned}$$

with stage $k \in \mathbb{N}_0$, prediction horizon $N \in \mathbb{N}$, input $u(k) \in \mathbb{R}^m$, state $x(k) \in \mathbb{R}^n$, column vectors $U = \left(u^T(0), ..., u^T(N-1)\right)^T \in \mathbb{R}^{Nm}$ and $X = \left(x^T(1), ..., x^T(N)\right)^T \in \mathbb{R}^{Nn}$, weighting matrices for inputs $R \in \mathbb{R}^{m \times m}$, states $Q \in \mathbb{R}^{n \times n}$, and terminal state $P \in \mathbb{R}^{n \times n}$, system matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ of the discrete-time time-invariant linear system, and constraint set for inputs $\mathcal{U} \subset \mathbb{R}^m$, states $\mathcal{X} \subset \mathbb{R}^n$, and the terminal state $\mathcal{T} \subset \mathbb{R}^n$. We assume the initial state $x(0)$ is given, matrices R and Q are such that $R \succ 0$ and $Q \succeq 0$, (A, B) is stabilizable, sets $\mathcal{U}$ and $\mathcal{X}$ are compact full-dimensional polytopes that contain the origin in their interiors, and matrix P and set $\mathcal{T}$ are the optimal cost function matrix of the unconstrained infinite-horizon problem which implies $P \succ 0$, and the largest possible set such that the optimal feedback for the unconstrained infinite-horizon problem stabilizes the system without violating the constraints, respectively.

Substituting system (1b) in OCP (1) enables to rewrite the OCP as the quadratic program (QP)

$$\begin{aligned} \min_{U} \quad & \frac{1}{2} U^T H U + x(0)^T F U + \frac{1}{2} x(0)^T Y x(0) \\ \text{s.t.} \quad & GU \leq Ex(0) + w, \end{aligned} \qquad (2)$$

with $H \in \mathbb{R}^{Nm \times Nm}$, $F \in \mathbb{R}^{n \times Nm}$, $Y \in \mathbb{R}^{n \times n}$, $G \in \mathbb{R}^{q \times Nm}$, $E \in \mathbb{R}^{q \times n}$, and $w \in \mathbb{R}^q$. The assumptions on OCP (1) imply $H \succ 0$ in QP (2) [2]. Without restriction, we assume the inequality constraints in QP (2) are formulated such that $w = 1^q$.

Let q refer to the number of inequalities in QP (2) and let $\mathcal{Q} := \{1, ..., q\}$ denote the index set. Furthermore, let sets $\mathcal{A}(x(0))$ and $\mathcal{I}(x(0))$ denote the optimal active and inactive set for any $x(0)$ such that QP (2) has a solution, respectively, with

$$\mathcal{A}(x(0)) := \left\{ i \in \mathcal{Q} \,|\, G_{\{i\}} U^\star(x(0)) = E_{\{i\}} x(0) + w_{\{i\}} \right\}, \qquad \text{(3a)}$$

$$\mathcal{I}(x(0)) := \mathcal{Q} \backslash \mathcal{A}(x(0)), \qquad \text{(3b)}$$

and where $U^\star(x(0))$ denotes the optimal solution to the QP. We often drop $x(0)$ in the notation of an active and inactive set, and call an active set $\mathcal{A}$ *optimal* if it refers to the optimal solution $U^\star(x(0))$ for some $x(0)$ with (3a).

Solving QP (2) as a parametric program, where the parameter is the initial state $x(0)$, results in the optimal control law $x(0) \mapsto U^\star(x(0))$ which is a continuous piecewise affine function of $x(0)$ on a polytopic partition of the feasible parameter space [2, Sect. 4.1]. The affine pieces in the solution are represented by optimal active sets $\mathcal{A}$ that define full-dimensional state-space polytopes and such that matrix $G_\mathcal{A}$ has full row rank. We collect all active sets that satisfy these conditions in the solution set $\mathcal{M}$.

A. *Combinatorial approach from [1]*

The power set $\mathcal{P}(\mathcal{Q})$ contains all possible constraint combinations or, equivalently, all possible active sets. It follows that $\mathcal{M} \subseteq \mathcal{P}(\mathcal{Q})$. By definition of $\mathcal{M}$, all elements $\mathcal{A} \in \mathcal{M}$ are such that the matrix $G_\mathcal{A}$ has full row rank. Since $G_\mathcal{A} \in \mathbb{R}^{|\mathcal{A}| \times Nm}$, the rank of matrix $G_\mathcal{A}$ is bounded from above by $\min(|\mathcal{A}|, Nm)$. It follows that active sets $\mathcal{A}$ with $|\mathcal{A}| > Nm$ result in row rank deficient $G_\mathcal{A}$. Consequently, $\mathcal{M} \subseteq \mathcal{P}^{Nm}(\mathcal{Q})$. The combinatorial approach from [1] considers the active sets $\mathcal{A} \in \mathcal{P}^{Nm}(\mathcal{Q})$ as candidates for the solution.

Figure 1 shows a rooted tree that represents the candidates in $\mathcal{P}^{Nm}(\mathcal{Q})$. The tree was first proposed by [18, Def. 2.2] and is used by many combinatorial approaches [1], [10], [14]. We refer to the tree as a combinatorial tree. The combinatorial tree is organized such that candidates of identical cardinalities appear in the same level and candidates containing the lowest constraint indices appear leftmost within each level without restriction. Let the set of candidates in the subtree of a candidate $\mathcal{A}$, i.e., all descendants of $\mathcal{A}$ and $\mathcal{A}$ itself be denoted $\mathcal{D}(\mathcal{A})$,

$$\mathcal{D}(\mathcal{A}) := \left\{ \mathcal{A} \cup \tilde{\mathcal{A}} \,|\, \tilde{\mathcal{A}} \in \mathcal{P}(\{\max(\mathcal{A}) + 1, ..., q\}) \right\}.$$

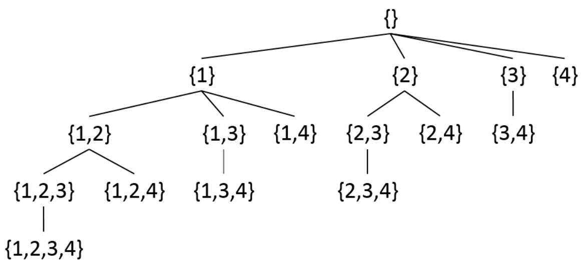

Fig. 1: Combinatorial tree for arbitrary OCP with $q = 4$ constraints.

The solution $\mathcal{M}$ is calculated by collecting all candidates $\mathcal{A} \in \mathcal{P}^{Nm}(\mathcal{Q})$ that satisfy the following conditions: (i) $G_\mathcal{A}$ has full row rank, (ii) $\mathcal{A}$ is optimal, and (iii) $\mathcal{A}$ defines a full-dimensional polytope. The approach from [1] is stated in Alg. 1. It explores all candidates in the order of increasing cardinality (line 2 in Alg. 1). When transferred to the combinatorial tree introduced before, the elements in the tree are explored from top to bottom. For each candidate, a rank test is executed to test for condition (i) (line 3). Candidates $\mathcal{A}$ such that matrix $G_{\mathcal{A}}$ is row rank deficient are dismissed. To test for condition (ii), the linear program (LP) (4) is solved (line 5),

$$\min_{U,x(0),\lambda_{\mathcal{A}},s_{\mathcal{I}},t} \quad -t \tag{4a}$$

$$\text{s.t.} \quad F^T x(0) + HU + (G_{\mathcal{A}})^T \lambda_{\mathcal{A}} = 0, \tag{4b}$$

$$1^{|\mathcal{A}|} t \leq \lambda_{\mathcal{A}}, \tag{4c}$$

$$G_{\mathcal{A}} U - E_{\mathcal{A}} x(0) - w_{\mathcal{A}} = 0, \tag{4d}$$

$$G_{\mathcal{I}} U - E_{\mathcal{I}} x(0) - w_{\mathcal{I}} + s_{\mathcal{I}} = 0, \tag{4e}$$

$$1^{|\mathcal{I}|} t \leq s_{\mathcal{I}},\ t \geq 0, \tag{4f}$$

with Lagrange multipliers $\lambda_{\mathcal{A}}$ and slack variables $s_{\mathcal{I}}$. Candidates such that LP (4) has a solution are optimal, and not optimal otherwise. If a solution to (4) exists, we denote corresponding optimization variables $U^\star$, $x^\star(0)$, $\lambda^\star_{\mathcal{A}}$, $s^\star_{\mathcal{I}}$, and $t^\star$. The following procedure enables the approach to identify many candidates at a time that are not optimal: For all candidates such that LP (4) has no solution, we solve LP (4) without (4b) and (4c) (lines 12–13). We call candidates such that this LP has a solution *feasible*, and *infeasible* otherwise. Infeasible active sets are not optimal because the LP to test for feasibility involves only a subset of the constraints of the LP to test for optimality. The set $\mathcal{S}$ collects candidates that are detected as infeasible (lines 14–15). Active sets $\mathcal{A}$ that are supersets of infeasible active sets are infeasible with [1, Thm. 1], and thus not optimal. Dismissing candidates because they are supersets of detected infeasible active sets (line 4) is further referred to as pruning. To test for the last condition (iii), it suffices to test candidates that result in $t^\star = 0$ for defining a full-dimensional polytope (lines 9–11). Candidates that result in $t^\star > 0$ define full-dimensional polytopes with [4, Thm. 2] since this result implies strict inequality for $0 < \lambda^\star_{\mathcal{A}}$ and $G_{\mathcal{I}} U^\star - E_{\mathcal{I}} x^\star(0) - w_{\mathcal{I}} < 0$ (lines 7–8).

Algorithm 1: Combinatorial approach from [1]

```
1  Initialization: set M = ∅, S = ∅
2  for every A ∈ P^{Nm}(Q) by incr. cardinality do
3      if rank(G_A) = |A| then
4          if A ⊉ Ã for all Ã ∈ S then
5              solve (4)
6              if solution exists then
7                  if solution t* > 0 then
8                      add A to M
9                  else
10                     if polytope def. by A is full-dim. then
11                         add A to M
12             else
13                 solve (4) without (4b) and (4c)
14                 if no solution exists then
15                     add A to S
16 Output: M
```

III. Symmetric OCPs and strategy for combinatorial tree

We follow [13, Def. 4] and call an OCP (1) symmetric to the pair (Θ, Ω), where matrices $\Theta \in \mathbb{R}^{n \times n}$ and $\Omega \in \mathbb{R}^{m \times m}$ are invertible, if the conditions

$$\begin{gathered} \Theta A = A\Theta, \quad \Theta B = B\Omega, \\ \Theta \circ \mathcal{X} = \mathcal{X}, \quad \Omega \circ \mathcal{U} = \mathcal{U}, \quad \Theta \circ \mathcal{T} = \mathcal{T}, \\ \Theta^T Q \Theta = Q, \quad \Omega^T R \Omega = R, \quad \Theta^T P \Theta = P \end{gathered} \tag{5}$$

hold. Note that symmetries of the system matrices and constraint sets in (5) result from symmetries of the underlying physical system and its constraints, respectively, and the weighting matrices can often be chosen to comply with some desired symmetry properties. A method for the identification of all pairs (Θ, Ω) that satisfy conditions (5) for an OCP (1) was proposed by [19]. We collect all pairs (Θ, Ω) that satisfy conditions (5) for an arbitrary but fixed OCP (1) in the set $\mathcal{G}$. Each pair $(\Theta, \Omega) \in \mathcal{G}$ causes the optimal control law of OCP (1) to be invariant under transformations of the input- and state-space with Ω and Θ, respectively [13].

Consider an OCP (1) that is reformulated as QP (2) with $w = 1^q$ and a pair $(\Theta, \Omega) \in \mathcal{G}$. With regard to the constraints in the QP (2), every constraint $i \in \mathcal{Q}$ has a symmetric constraint $j \in \mathcal{Q}$ such that

$$G_{\{i\}} = G_{\{j\}} \cdot \left(I^N \otimes \Omega\right) \text{ and } E_{\{i\}} = E_{\{j\}} \cdot \Theta \tag{6}$$

[17, Thm. 2]. We introduce the function $\pi^{(\Theta,\Omega)} : \mathcal{Q} \to \mathcal{Q}$ to represent the relations $j = \pi^{(\Theta,\Omega)}(i)$ from (6). Also, we introduce the function $\Pi^{(\Theta,\Omega)} : \mathcal{P}(\mathcal{Q}) \to \mathcal{P}(\mathcal{Q})$ to map all constraints in an active set $\mathcal{A} \in \mathcal{P}(\mathcal{Q})$ to their corresponding symmetric constraints with $\pi^{(\Theta,\Omega)}$. We call the active set $\mathcal{A}_j = \Pi^{(\Theta,\Omega)}(\mathcal{A}_i)$ the symmetric active set to $\mathcal{A}_i$ under pair (Θ, Ω).

Let the orbit $\mathcal{O}(\mathcal{A})$ collect the symmetric active sets to an active set $\mathcal{A}$ under all pairs $(\Theta, \Omega) \in \mathcal{G}$,

$$\mathcal{O}(\mathcal{A}) := \left\{ \Pi^{(\Theta,\Omega)}(\mathcal{A}) \,|\, (\Theta, \Omega) \in \mathcal{G} \right\}. \tag{7}$$

We often drop the argument $\mathcal{A}$ in the notation of an orbit. It was shown in [17] that the orbits of all active sets in $\mathcal{P}(\mathcal{Q})$ partition $\mathcal{P}(\mathcal{Q})$. Likewise, the orbits of all active sets in $\mathcal{P}^{Nm}(\mathcal{Q})$ partition $\mathcal{P}^{Nm}(\mathcal{Q})$ since the elements of an orbit have identical cardinalities. It follows that the set of candidates for the combinatorial approach from [1] introduced in Sect. II-A partitions into orbits.

All elements of an orbit have properties in common [17, Thm. 4] that are relevant for combinatorial approaches:

1) The elements of an orbit are either all optimal or all not optimal.
2) The polytopes defined by the elements of an orbit are either all full-dimensional or all lower-dimensional in state-space.

It is suggested by [17] to improve combinatorial approaches by only testing one active set and its polytope of each orbit for optimality and full-dimensionality, respectively, and applying the result to all active sets in the orbit. However, it is not trivial to efficiently identify active sets that can be dismissed from testing. The following strategy is suggested by [17]: For each orbit $\mathcal{O}$, we call the active set containing the lowest constraint indices, i.e.,

$$\mathcal{A} \in \mathcal{O} : \min(\mathcal{A} \backslash \tilde{\mathcal{A}}) < \min(\tilde{\mathcal{A}} \backslash \mathcal{A}) \; \forall \, \tilde{\mathcal{A}} \in \mathcal{O} \backslash \mathcal{A}$$

the *primary* active set. Accordingly, all orbit elements except for the primary active set are *non-primary*. It is sufficient to process only the primary active set and dismiss all other active sets of the orbit from testing. Assuming all active sets that are candidates are explored in the order of increasing cardinality and increasing constraint indices (when transferred to the combinatorial tree, this strategy is from top to bottom and in each level from left to right), the primary active set of each orbit is reached first. The residual elements of the orbits are non-primary and can therefore be dismissed. With [17, Thm. 5], the active sets of the subtree $\mathcal{D}(\mathcal{A})$ are non-primary if the root of the subtree $\mathcal{A}$ is non-primary. We refer to [17, Example 6] for an illustrative example.

IV. Improved combinatorial approaches

Based on the strategy described in Sect. III, we improve the combinatorial approach from [1] introduced as Alg. 1 in Sect. II-A. The improvement of the approach is achieved by only processing primary and dismissing non-primary candidates. The complete solution still results because the whole orbit is added to the solution $\mathcal{M}$ whenever a primary candidate is detected to be part of $\mathcal{M}$.

The improved approach is stated in Alg. 2. It processes all candidates $\mathcal{A} \in \mathcal{P}^{Nm}(\mathcal{Q})$ in the order of increasing cardinality and increasing constraint indices (line 2 in Alg. 2). This way, the primary candidate of each orbit is processed first. The set $\mathcal{N}$ collects candidates that are detected as non-primary. Any candidate $\mathcal{A}$ such that $\mathcal{A} \in \mathcal{D}(\bar{\mathcal{A}})$ for an $\bar{\mathcal{A}} \in \mathcal{N}$ is non-primary and thus dismissed (line 4). All elements of the orbit of a primary candidate, except the primary candidate itself, are non-primary. To keep the number of elements in $\mathcal{N}$ small, a non-primary candidate is only added to $\mathcal{N}$ if the candidate is not already an element of a set $\mathcal{D}(\bar{\mathcal{A}})$, $\bar{\mathcal{A}} \in \mathcal{N}$ (lines 17–19). All other lines in Alg. 2 (lines 3, 5–16, 20) remain unchanged from Alg. 1 but add the whole orbit to the solution $\mathcal{M}$ whenever a candidate is detected to be part of $\mathcal{M}$ (lines 9, 12).

Solving the LPs accounts for a large share of the computing time. The number of solved LPs for Alg. 2 is almost reduced by factor g, the number of symmetries, which will be explained in more detail in Sect. V. However, Alg. 2 executes additional computational operations that counteract the reduction of computational time, most notably the tests in lines 4 and 18 to test if an active set is a descendant of a detected non-primary active set. The additional computational effort is most evident for examples with a large number of symmetries for two reasons: Firstly, the set $\mathcal{N}$ contains more elements because a large number of symmetries entails a large number of non-primary active sets. Thus, the test in lines 4 and 18 is more expensive. Secondly, the larger the number of symmetries the larger the number of elements of an orbit. Thus, the for loop starting in line 17 runs more often.

We suggest a different procedure in Alg. 3 that obviates these issues and need Cor. 1 as a preparation. It extends the properties that elements of an orbit have in common by the property feasibility.

Algorithm 2: Improved combinatorial approach 1

1 **Initialization:** set $\mathcal{M} = \emptyset$, $\mathcal{S} = \emptyset$, $\mathcal{N} = \emptyset$
2 **for** *every* $\mathcal{A} \in \mathcal{P}^{Nm}(\mathcal{Q})$ *by incr. cardinality and by incr. constraint indices* **do**
3   **if** *rank*$(G_{\mathcal{A}}) = |\mathcal{A}|$ **then**
4   **if** $\mathcal{A} \notin \mathcal{D}(\bar{\mathcal{A}})$ *for all* $\bar{\mathcal{A}} \in \mathcal{N}$ **then**
5   **if** $\mathcal{A} \not\supseteq \tilde{\mathcal{A}}$ *for all* $\tilde{\mathcal{A}} \in \mathcal{S}$ **then**
6   solve (4)
7   **if** *solution exists* **then**
8   **if** *solution* $t^\star > 0$ **then**
9   add elements of $\mathcal{O}(\mathcal{A})$ to $\mathcal{M}$
10   **else**
11   **if** *polytope def. by* $\mathcal{A}$ *is full-dim* **then**
12   add elements of $\mathcal{O}(\mathcal{A})$ to $\mathcal{M}$
13   **else**
14   solve (4) without (4b) and (4c)
15   **if** *no solution exists* **then**
16   add $\mathcal{A}$ to $\mathcal{S}$
17   **for** *every* $\breve{\mathcal{A}} \in \mathcal{O}(\mathcal{A}) \backslash \mathcal{A}$ **do**
18   **if** $\breve{\mathcal{A}} \notin \mathcal{D}(\bar{\mathcal{A}})$ *for all* $\bar{\mathcal{A}} \in \mathcal{N}$ **then**
19   add $\breve{\mathcal{A}}$ to $\mathcal{N}$
20 **Output:** $\mathcal{M}$

Corollary 1. *Consider an OCP* (1)*. The active sets of an orbit are either all feasible or all infeasible.*

Proof. For an arbitrary active set $\mathcal{A}$, let the sets $\mathcal{F}_{x(0)}(\mathcal{A})$ and $\mathcal{F}_U(\mathcal{A})$ contain all initial states $x(0)$ and input vectors U, respectively, such that a solution to LP (4) without (4b) and (4c) exists, i.e.,

$$\mathcal{F}_{x(0)}(\mathcal{A}) \times \mathcal{F}_U(\mathcal{A}) := \big\{(x(0), U) \in \mathbb{R}^n \times \mathbb{R}^{Nm} \mid G_{\mathcal{A}}U + E_{\mathcal{A}}x(0) + w_{\mathcal{A}} = 0,\, G_{\mathcal{I}}U + E_{\mathcal{I}}x(0) + w_{\mathcal{I}} \leq 0\big\}\,.$$

An active set $\mathcal{A}$ that is feasible results in $\mathcal{F}_{x(0)}(\mathcal{A}) \times \mathcal{F}_U(\mathcal{A}) \neq \emptyset$, an active set $\mathcal{A}$ that is infeasible results in $\mathcal{F}_{x(0)}(\mathcal{A}) \times \mathcal{F}_U(\mathcal{A}) = \emptyset$. Consider an orbit $\mathcal{O}$ and two of its elements $\mathcal{A}_i, \mathcal{A}_j \in \mathcal{O}$. The proof is done by showing that if $\mathcal{F}_{x(0)}(\mathcal{A}_i) \times \mathcal{F}_U(\mathcal{A}_i) = \emptyset$ then $\mathcal{F}_{x(0)}(\mathcal{A}_j) \times \mathcal{F}_U(\mathcal{A}_j) = \emptyset$ and if $\mathcal{F}_{x(0)}(\mathcal{A}_i) \times \mathcal{F}_U(\mathcal{A}_i) \neq \emptyset$ then $\mathcal{F}_{x(0)}(\mathcal{A}_j) \times \mathcal{F}_U(\mathcal{A}_j) \neq \emptyset$. Active sets $\mathcal{A}_i$ and $\mathcal{A}_j$ are considered to be elements of the same orbit which implies there exists a pair $(\Theta, \Omega) \in \mathcal{G}$ such that $\mathcal{A}_j = \Pi^{(\Theta,\Omega)}(\mathcal{A}_i)$. It holds

$$\mathcal{F}_{x(0)}(\mathcal{A}_j) = \Theta \circ \mathcal{F}_{x(0)}(\mathcal{A}_i),\; \mathcal{F}_U(\mathcal{A}_j) = \Omega \circ \mathcal{F}_U(\mathcal{A}_i) \tag{8}$$

since pair (Θ, Ω) causes transformations of the input- and state-space with Ω and Θ, respectively. Equation (8) implies either both sets $\mathcal{F}_{x(0)}(\mathcal{A}_j)$ and $\mathcal{F}_{x(0)}(\mathcal{A}_i)$ are empty or both are not empty and, in the same way, either both sets $\mathcal{F}_U(\mathcal{A}_j)$ and $\mathcal{F}_U(\mathcal{A}_i)$ are empty or both are not empty. ☐

Corollary 1 enables adding the whole orbit to set $\mathcal{S}$ whenever an active set is detected as infeasible. To keep the number of elements in set $\mathcal{S}$ small, orbit elements are only added to $\mathcal{S}$ if they are not already a superset of a detected infeasible active set in $\mathcal{S}$ (line 23–26 in Alg. 3). Due to the larger number of detected infeasible active sets in $\mathcal{S}$, the effectiveness of pruning increases. It is therefore advantageous to dismiss candidates that are supersets of detected infeasible active sets (pruning) first (line 4) and dismiss candidates which are descendants of detected non-primary active sets second (line 5). It follows that we need not store non-primary sets that are infeasible in $\mathcal{N}$ because these candidates are dismissed by pruning in line 4 and never appear in line 5. Therefore, lines 17–19 from Alg. 2 are implemented in lines 13–15 and 19–21 in Alg. 3.

Algorithm 3: Improved combinatorial approach 2

1 **Initialization:** set $\mathcal{M} = \emptyset$, $\mathcal{S} = \emptyset$, $\mathcal{N} = \emptyset$
2 **for** *every* $\mathcal{A} \in \mathcal{P}^{Nm}(\mathcal{Q})$ *by incr. cardinality and by incr. constraint indices* **do**
3   **if** *rank*$(G_{\mathcal{A}}) = |\mathcal{A}|$ **then**
4   **if** $\mathcal{A} \not\supseteq \tilde{\mathcal{A}}$ *for all* $\tilde{\mathcal{A}} \in \mathcal{S}$ **then**
5   **if** $\mathcal{A} \notin \mathcal{D}(\bar{\mathcal{A}})$ *for all* $\bar{\mathcal{A}} \in \mathcal{N}$ **then**
6   solve (4)
7   **if** *solution exists* **then**
8   **if** *solution* $t^\star > 0$ **then**
9   add elements of $\mathcal{O}(\mathcal{A})$ to $\mathcal{M}$
10   **else**
11   **if** *polytope def. by* $\mathcal{A}$ *is full-dim* **then**
12   add elements of $\mathcal{O}(\mathcal{A})$ to $\mathcal{M}$
13   **for** *every* $\breve{\mathcal{A}} \in \mathcal{O}(\mathcal{A}) \backslash \mathcal{A}$ **do**
14   **if** $\breve{\mathcal{A}} \notin \mathcal{D}(\bar{\mathcal{A}})$ *for all* $\bar{\mathcal{A}} \in \mathcal{N}$ **then**
15   add $\breve{\mathcal{A}}$ to $\mathcal{N}$
16   **else**
17   solve (4) without (4b) and (4c)
18   **if** *solution exists* **then**
19   **for** *every* $\breve{\mathcal{A}} \in \mathcal{O}(\mathcal{A}) \backslash \mathcal{A}$ **do**
20   **if** $\breve{\mathcal{A}} \notin \mathcal{D}(\bar{\mathcal{A}})$ *for all* $\bar{\mathcal{A}} \in \mathcal{N}$ **then**
21   add $\breve{\mathcal{A}}$ to $\mathcal{N}$
22   **else**
23   **for** *every* $\breve{\mathcal{A}} \in \mathcal{O}(\mathcal{A})$ **do**
24   **if** $\breve{\mathcal{A}} \not\supseteq \tilde{\mathcal{A}}$ *for all* $\tilde{\mathcal{A}} \in \mathcal{S}$ **then**
25   add $\breve{\mathcal{A}}$ to $\mathcal{S}$
26 **Output:** $\mathcal{M}$

The changes implemented in Alg. 3 cause two things compared to Alg. 2: Firstly, the set $\mathcal{N}$ contains fewer elements because infeasible active sets are not elements anymore. This makes the test in lines 5, 14, and 20 less expensive. Secondly, testing if a candidate is a descendant of a detected non-primary active set is executed less often because the testing if a candidate is a descendant of a detected infeasible active set is executed before and dismisses many candidates already.

V. Computational analysis

We analyze the computational effort of Algs. 2 and 3 with two examples and compare them to the original approach from [1] (Alg. 1).

Example 1. *Consider the system [20, Example 1]*

$$x(k+1) = \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix} x(k) + I^2 u(k),$$

with input and state constraints $|u_i(k)| \leq 1$, $i = 1, 2$, *and* $|x_i(k)| \leq 1$, $i = 1, 2$, *respectively, weighting matrices* $Q = I^2$, $R = 5{,}000 \cdot I^2$, *and prediction horizon* $N = 3$. *The terminal weighting matrix* P *and set* $\mathcal{T}$ *are as described in Sect. II.*

TABLE I: Computational data for Example 1

	Algorithm 1	Algorithm 2	Algorithm 3
# solved LPs	11,969	3,059	3,043
# test pruning	47,545	11,887	47,647
# test descendant	0	83,202	13,755
computing time	77.8 s	26.0 s	23.6 s

TABLE II: Computational data for Example 2

	Algorithm 1	Algorithm 2	Algorithm 3
# solved LPs	111,681	14,197	14,149
# test pruning	3,264,401	408,051	3,265,129
# test descendant	0	6,120,750	266,252
computing time	1,001 s	706 s	478 s

Example 1 has four pairs $(\Theta_i, \Omega_i) \in \mathcal{G}$, $i = 1, \ldots, 4$, that satisfy (5). The pairs correspond to rotations of the input- and state-space by $\varphi_1 = 0°$, $\varphi_2 = 90°$, $\varphi_3 = 180°$, and $\varphi_4 = 270°$ with

$$(\Theta_i, \Omega_i) = \left(\begin{pmatrix} \cos(\varphi_i) & -\sin(\varphi_i) \\ \sin(\varphi_i) & \cos(\varphi_i) \end{pmatrix}, \begin{pmatrix} \cos(\varphi_i) & -\sin(\varphi_i) \\ \sin(\varphi_i) & \cos(\varphi_i) \end{pmatrix} \right). \tag{9}$$

All algorithms process $|\mathcal{P}^{Nm}(\mathcal{Q})| = 499{,}178$ candidates and determine the solution with $|\mathcal{M}| = 73$ elements. Rank tests are executed for each of the candidates by all algorithms. The remaining computing operations of the algorithms can be summarized to be the number of solved LPs to test for optimality with (4) and feasibility with (4) without (4b) and (4c), the number of tests to test if a candidate is a superset of a detected infeasible active set (test pruning) and the number of tests to test if a candidate is a descendant of a detected non-primary active set (test descendant). These numbers differ with the different procedures of the algorithms and are listed in Tab. I together with the computing times.

We introduce a second example with more symmetries.

Example 2. *Consider the system*

$$x(k+1) = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} x(k) + I^2 u(k),$$

with weighting matrices $Q, R = I^2$ *and prediction horizon* $N = 3$, *and where the constraint sets for inputs and states are octagons with spans* 20 *and* 2, *respectively, that are centered at the origin. The terminal weighting matrix* P *and set* $\mathcal{T}$ *are as described in Sect. II.*

Example 2 has eight pairs $(\Theta_i, \Omega_i) \in \mathcal{G}$, $i = 1, \ldots, 8$, that satisfy (5). The pairs correspond to rotations of the input- and state-space by $\varphi_1 = 0°$, $\varphi_2 = 45°$, $\varphi_3 = 90°$, $\varphi_4 = 135°$, $\varphi_5 = 180°$, $\varphi_6 = 225°$, $\varphi_7 = 270°$, and $\varphi_8 = 315°$ with (9). All algorithms process $|\mathcal{P}^{Nm}(\mathcal{Q})| = 36{,}684{,}859$ candidates for Example 2 and determine the solution with $|\mathcal{M}| = 97$ elements. Other computational data is listed in Tab. II.

It is evident from the data in Tabs. I and II that Algs. 2 and 3 require less computing time than Alg. 1. The reduction mainly results because fewer LPs are solved for Algs. 2 and 3 than for Alg. 1 and solving the LPs occupies a large share of the computing time. All algorithms generally dismiss candidates $\mathcal{A}$ such that matrix $G_{\mathcal{A}}$ is row rank deficient and that are supersets of detected infeasible active sets from solving an LP. Algorithms 2 and 3 additionally do not solve LPs for non-primary candidates such that all but one active set out of each orbit are dismissed from solving an LP. The number of elements of an orbit depends on the number of symmetries of the problem. Considering an OCP with g symmetries, each orbit contains up to g elements, see (7), so the number of solved LPs for Algs. 2 and 3 decreases to about $1/g$ compared to Alg. 1. However, Algs. 2 and 3 require additional computational effort for executing tests descendant. This effort diminishes the reduction of computational time, so the reduction in the computing times of Algs. 2 and 3 when compared to Alg. 1 is smaller than the reduction in the number of solved LPs. We will focus on that in the next paragraph.

Furthermore, the data in Tabs. I and II reveal that Alg. 3 requires less computing time than Alg. 2, for the highly symmetric Example 2 in particular. The differences result from the different numbers of executed tests pruning and tests descendant. In Alg. 2, tests descendant are executed for all candidates $\mathcal{A}$ such that matrix $G_{\mathcal{A}}$ has full row rank (line 4) and for all elements of the orbits of active sets that satisfy the rank criterion and that are primary (line 17). In Alg. 3, in contrast, tests descendant are executed for all candidates that satisfy the rank criterion and that are not supersets of detected infeasible active sets (line 5) and for all elements of the orbits of active sets that satisfy the rank criterion and that are primary and feasible (lines 14, 20). It follows that Alg. 3 executes fewer tests descendant than Alg. 2 since the criteria for the candidates being tested are more restrictive. Furthermore, executing a test descendant in Alg. 3 requires less computational effort than in Alg. 2 because $\mathcal{N}$ contains only feasible active sets and therefore fewer elements. In Alg. 2 tests pruning are executed for all candidates $\mathcal{A}$ such that matrix $G_{\mathcal{A}}$ has full row rank and that are non-primary (line 5) while Alg. 3 executes test pruning for all candidates that satisfy the rank criterion (line 4) and for all elements of the orbits of active sets that satisfy the rank criterion, that are not dismissed by test pruning, and that are primary and infeasible (line 24). As a result, the numbers of tests pruning for Alg. 2 are almost decreased by factor g compared to Alg. 3. In summary, Alg. 3 results in fewer tests descendant than Alg. 2 but more tests pruning. The more symmetries a problem has, the more the ratio between computational savings and expenditures shifts towards savings for Alg. 3 making Alg. 3 the more effective approach.

VI. Conclusion

We applied the methods from [17] for improving combinatorial approaches for symmetric OCPs to the approach from [1]. The improved approach requires fewer LPs to be solved resulting in a lower computational effort. The number of solved LPs drops by a factor of g, the number of symmetries of the OCP. However, the approach requires additional computing operations which counteract the reduction of the computational effort. These operations occur frequently for OCPs with many symmetries. For this reason, we presented a second approach that makes pruning more effective by taking symmetric feasibility properties into account. This results in fewer required additional computing operations. For OCPs with many symmetries, the second approach showed to be the approach that requires less computational effort.

References

[1] A. Gupta, S. Bhartiya, and P. Nataraj, "A novel approach to multiparametric quadratic programming," *Automatica*, vol. 47, pp. 2112–2117, 2011.

[2] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.

[3] M. M. Seron, G. C. Goodwin, and J. A. De Doná, "Characterisation of receding horizon control for constrained linear systems," *Asian Journal of Control*, vol. 5, pp. 271–286, 2003.

[4] P. Tøndel, T. A. Johanson, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica*, vol. 39, pp. 489–497, 2003.

[5] M. Baotić, "An efficient algorithm for multi-parametric quadratic programming," ETH Zürich, Tech. Rep., 2002, AUT02-05.

[6] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/∼mpt.

[7] A. Bemporad, "Hybrid Toolbox - User's Guide," 2004, http://cse.lab.imtlucca.it/∼bemporad/hybrid/toolbox.

[8] M. Herceg, C. N. Jones, M. Kvasnica, and M. Morari, "Enumeration-based approach to solving parametric linear complementarity problems," *Automatica*, vol. 62, pp. 243–248, 2015.

[9] R. Oberdieck, N. A. Diangelakis, and E. N. Pistikopoulos, "Explicit model predictive control: A connected-graph approach," *Automatica*, vol. 76, pp. 103–112, 2017.

[10] P. Ahmadi-Moshkenani, T. A. Johansen, and S. Olaru, "Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3221–3231, Oct 2018.

[11] M. Mönnigmann, "On the structure of the set of active sets in constrained linear quadratic regulation," *Automatica*, vol. 106, pp. 61–69, 2019.

[12] R. Mitze and M. Mönnigmann, "Improved active set dynamic programming for solving linear-quadratic optimal control problems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 1764–1769.

[13] C. R. Danielson and F. Borrelli, "Symmetric Linear Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1244–1259, 2015.

[14] C. Feller, T. A. Johanson, and S. Olaru, "An improved algorithm for combinatorial multi-parametric quadratic programming," *Automatica*, vol. 49, pp. 1370–1376, 2013.

[15] R. Mitze and M. Mönnigmann, "A dynamic programming approach to solving constrained linear-quadratic optimal control problems," *Automatica*, vol. 120, p. 109132, 2020b.

[16] R. Mitze and M. Mönnigmann, "Dynamic programming for explicit linear MPC with point-symmetric constraints," in *21st IFAC World Congress, Berlin*, 2020a, pp. 6999–7004.

[17] R. Mitze, M. Kvasnica, and M. Mönnigmann, "Exploiting symmetries in active set enumeration for constrained linear-quadratic optimal control," *Automatica*, vol. 151, p. 110900, 2023.

[18] R. Rymon, "Search through systematic set enumeration," in *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92)*, 1992, pp. 539–550.

[19] C. R. Danielson and F. Borrelli, "Identification of the symmetries of linear systems with polytopic constraints," in *Proc. of the American Control Conference*, 2014, pp. 4218–4223.

[20] C. R. Danielson and F. Borrelli, "Symmetric explicit model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 132–137, 2012.