An improved algorithm for combinatorial multi-parametric quadratic programming

Christian Feller [a,b], Tor Arne Johansen [b], Sorin Olaru [c],

[a] *Institute for Systems Theory and Automatic Control, University of Stuttgart, Pfaffenwaldring 9, 70550 Stuttgart, Germany*

[b] *Department of Engineering Cybernetics, NTNU, O.S. Bragstads plass 2D, 7491 Trondheim, Norway*

[c] *SUPÉLEC Systems Sciences (E3S) – Automatic Control Department, 3 rue Joliot-Curie, 91192 Gif-Sur-Yvette, France,*

Abstract

The goal of multi-parametric quadratic programming (mpQP) is to compute analytic solutions to parameter-dependent constrained optimization problems, e.g., in the context of explicit linear MPC. We propose an improved combinatorial mpQP algorithm that is based on implicit enumeration of all possible optimal active sets and a simple saturation matrix pruning criterion which uses geometric properties of the constraint polyhedron for excluding infeasible candidate active sets. In addition, techniques are presented that allow to reduce the complexity of the discussed algorithm in the presence of symmetric problem constraints. Performance improvements are discussed for two example problems from the area of explicit linear MPC.

Key words: Multi-parametric programming; Explicit constrained linear quadratic regulators; Predictive control

1 Introduction

Multi-parametric programming (mpP) techniques can be used to compute explicit solutions to parameter-dependent constrained optimization problems, e.g., as they occur in the area of linear model predictive control (MPC). In this work, we will focus on strictly convex multi-parametric quadratic programming (mpQP) problems, which are related to linear MPC problems with a quadratic cost function, i.e., the constrained finite-horizon LQR problem. In general, the solution has the form of a piecewise affine function over a polyhedral partition of the parameter space into so-called *critical regions*, where each region corresponds to a set of optimal active constraints.

Most of the mpQP algorithms reported in the literature are based on geometric methods and apply recursive exploration strategies in order to identify all critical regions of the explicit solution. In their famous paper, [2] proposed a simple algorithm that subdivides the parameter space into polyhedral regions by reversing recursively the facet-defining hyperplanes of all previously identified regions. Unfortunately, this approach introduces artificial cuts in the parameter space, which can result in unnecessary and redundant partitioning. More efficient exploration strategies were presented by [1], [13], and [14], based on the assumption that for each facet of a critical region there exists only one neighboring critical region that is adjacent to this facet. The parameter space is then explored iteratively by stepping over all the facets of already identified regions and solving the mpQP problem for new parameter vectors, or by examining the type of the facet-defining hyperplanes, respectively. In [10], the authors proposed an algorithm that combines the approaches from [2] and [13] in order to handle situations in which this facet-to-facet property does not hold. Moreover, additional geometric approaches were proposed by [9] and [8]: in [9], the authors consider all possible configurations of the constraint polyhedron in order to induce a partition of the input space, while a parametrized polyhedra approach in the combined (input+parameter) space is used in [8].

Recently, [6] have presented a new combinatorial mpQP approach that is based on an implicit enumeration of all possible constraint combinations in form of candidate active sets. While this combinatorial approach does

⋆ This paper was not presented at any IFAC meeting. Corresponding author C. Feller Tel. +49-711-685 67745. Fax +49-711-685 67735.

Email addresses: `christian.feller@ist.uni-stuttgart.de` (Christian Feller), `tor.arne.johansen@itk.ntnu.no` (Tor Arne Johansen), `sorin.olaru@supelec.fr` (Sorin Olaru).

not rely on an explicit geometric exploration strategy and may have some structural advantages over existing geometric algorithms, one disadvantage is the combinatorial complexity with respect to the number of possible candidate active sets.

In this paper, we propose an improved combinatorial mpQP algorithm that uses some of the underlying geometric properties of the involved problem constraints in order to increase the efficiency of the combinatorial active set enumeration. The algorithm is based on a new pruning mechanism that allows to detect infeasible combinations of active constraints by a simple row sum check on the so-called saturation matrix of the non-parametrized constraint polyhedron. Furthermore, we present additional complexity reduction techniques which are based on exploiting symmetries in the mpQP problem formulation. Two example problems from the area of explicit linear MPC are used to demonstrate the achieved performance improvements.

2 Multi-parametric quadratic programming

In the following, we will focus on standard mpQP problems of the form

$$V_z^*(x) = \min_z \frac{1}{2} z^T H z \tag{1a}$$
$$\text{s. t. } \; Gz \leq W + Sx \;, \tag{1b}$$

where $z \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ denote the vectors of optimization variables and parameters, and $H \in \mathbb{R}^{m\times m}$, $G \in \mathbb{R}^{q\times m}$, $W \in \mathbb{R}^q$, $S \in \mathbb{R}^{q\times n}$ are real matrices. We assume that all constraints on x are included in (1b) and that the problem is strictly convex, i.e., that $H \succ 0$. Without loss of generality, we further assume that (1b) does not contain any redundant constraints [13].

2.1 Analytic solutions to mpQP problems

As shown by [2] and [13], we can solve (1) by applying the Karush-Kuhn-Tucker (KKT) conditions, which for this problem are given by

$$Hz + G^T\lambda = 0, \quad \lambda \in \mathbb{R}^q, \tag{2a}$$
$$\lambda^i \left(G^i z - W^i - S^i x\right) = 0, \quad i = 1, \ldots, q, \tag{2b}$$
$$\lambda \geq 0, \quad Gz \leq W + Sx. \tag{2c}$$

Here, the superscript index i denotes the i^{th} row of a matrix or vector and λ refers to the vector of Lagrangian multipliers. Note that here the KKT conditions are not only necessary but also sufficient conditions for optimality since we assume $H \succ 0$.

Definition 1 (Optimal active set) *Let $z^*(x)$ be the optimal solution to (1) for a given parameter vector x and let $\mathcal{Q} := \{1, \ldots, q\}$ refer to the index set of all constraints in (1b). Then, the optimal active set $\mathcal{A}^*(x)$ is defined as the index set of active constraints at the optimum: $\mathcal{A}^*(x) := \{i \in \mathcal{Q} \mid G^i z^*(x) - W^i - S^i x = 0\}$.*

Assuming that we know the optimal active set $\mathcal{A}^*(x)$ for a given x, we can form submatrices $G^{\mathcal{A}^*(x)}$, $W^{\mathcal{A}^*(x)}$, $S^{\mathcal{A}^*(x)}$ that contain the constraints associated to the indices in $\mathcal{A}^*(x)$. In the following, we drop the explicit parametrization of $\mathcal{A}^*(x)$ for the ease of notation.

Definition 2 (LICQ) *For an index set $\mathcal{A} \subseteq \mathcal{Q}$, the linear independence constraint qualification (LICQ) holds if the gradients of the corresponding constraints are linearly independent, i.e., if $G^{\mathcal{A}}$ has full row rank.*

When assuming that LICQ holds for a given optimal active set $\mathcal{A}$, we can use the first two equations of the KKT conditions to derive the parameter-dependent optimizer

$$z_{\mathcal{A}}(x) = H^{-1}(G^{\mathcal{A}})^T H_{G^{\mathcal{A}}}^{-1} \left(W^{\mathcal{A}} + S^{\mathcal{A}} x\right) \tag{3}$$

for a fixed parameter vector x, where $H_{G^{\mathcal{A}}}^{-1} := (G^{\mathcal{A}} H^{-1} (G^{\mathcal{A}})^T)^{-1}$ always exists since $H \succ 0$ and LICQ holds [2,13]. Moreover, the two remaining KKT inequality conditions characterize the so-called *critical region* in which the solution (3) remains optimal when varying the parameter x:

$$-H_{G^{\mathcal{A}}}^{-1} \left(W^{\mathcal{A}} + S^{\mathcal{A}} x\right) \geq 0. \tag{4a}$$
$$GH^{-1}(G^{\mathcal{A}})^T H_{G^{\mathcal{A}}}^{-1} \left(W^{\mathcal{A}} + S^{\mathcal{A}} x\right) \leq W + Sx. \tag{4b}$$

This polyhedral region in the state space is the largest set of parameters for which the combination of active constraints at the optimizer remains unchanged, i.e., for which $\mathcal{A}$ remains the optimal active set. Thus, by identifying all optimal active sets $\mathcal{A}_i$, the parameter space is implicitly partitioned into several critical regions $CR_{\mathcal{A}_i}$, and the optimizer can be represented as a continuous piecewise affine function of the parameter vector x [2,13]. Note that if the LICQ assumption fails to hold, i.e., if the rows of $G^{\mathcal{A}}$ are linearly dependent, the inverse $H_{G^{\mathcal{A}}}^{-1}$ does not exist and further methods have to be applied in order to obtain a representation of $z_{\mathcal{A}}$ and $CR_{\mathcal{A}}$, e.g., proceeding with full-rank subsets of the active constraints or using projections in the (λ, x)-space to obtain the full-dimensional critical region [2,14].

2.2 Combinatorial mpQP

While most of the existing geometric mpQP algorithms construct the solution by identifying all critical regions in a recursive parameter space exploration, the combinatorial mpQP algorithm presented by [6] operates directly on the level of possible optimal active sets. The main idea of the approach is the implicit enumeration of all possible combinations of active constraints, which we will shortly summarize in the following.

Consider again the set $\mathcal{Q} = \{1, \ldots, q\}$ referring to the constraint indices in (1b). Then, the active set $\mathcal{A}(z,x)$ can be described as $\mathcal{A}(z,x) := \{i \in \mathcal{Q} | G^i z - W^i - S^i x = 0\}$ while the corresponding set of inactive constraints $\mathcal{J}(z,x)$ is given by the set difference of $\mathcal{Q}$ and $\mathcal{A}$, i.e., $\mathcal{J}(z,x) := \mathcal{Q} \setminus \mathcal{A}(z,x)$. As pointed out by [6], all possible optimal active sets are included in the set $\mathcal{P}'(\mathcal{Q}) := \big\{\mathcal{A}_1 = \{\ \}, \mathcal{A}_2 = \{1\}, \ldots, \mathcal{A}_{q+1} = \{q\}, \ldots \mathcal{A}_{q+2} = \{1,2\}, \ldots, \mathcal{A}_{n'_{\mathcal{A}}} = \{\tilde{q} - m + 1, \ldots, q\}\big\}$ which is a subset of the power set $\mathcal{P}(\mathcal{Q}) = \big\{\mathcal{A}_1 = \{\ \}, \ldots, \mathcal{A}_{2^q} = \{1, \ldots, q\}\big\}$ and consists of $n'_{\mathcal{A}} = \sum_{i=0}^{\tilde{m}} \binom{q}{i} \leq 2^q$ index sets. Here, $\tilde{q}$ and $\tilde{m}$ are defined as $\tilde{q} = \max\{m, q\}$ and $\tilde{m} = \min\{m, q\}$, respectively. In order to identify all optimal active sets, [6] suggest to choose candidate active sets $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$ in the order of increasing cardinality and use the LP

$$\max_{z,x,\lambda^{\mathcal{A}_i},s^{\mathcal{J}_i}} \quad t \tag{5a}$$

$$\text{s.t.} \quad te_1 \leq \lambda^{\mathcal{A}_i},\ te_2 \leq s^{\mathcal{J}_i} \tag{5b}$$

$$t \geq 0,\ \lambda^{\mathcal{A}_i} \geq 0,\ s^{\mathcal{J}_i} \geq 0 \tag{5c}$$

$$Hz + (G^{\mathcal{A}_i})^T \lambda^{\mathcal{A}_i} = 0 \tag{5d}$$

$$G^{\mathcal{A}_i} z - S^{\mathcal{A}_i} x - W^{\mathcal{A}_i} = 0 \tag{5e}$$

$$G^{\mathcal{J}_i} z - S^{\mathcal{J}_i} x - W^{\mathcal{J}_i} + s^{\mathcal{J}_i} = 0 \tag{5f}$$

to check whether there exists a feasible point in the parameter space for which $\mathcal{A}_i$ is the optimal active set. Here, in addition to the already introduced variables and matrices, t is a scalar optimization variable and $e_1 = [1, \ldots, 1]^T$, $e_2 = [1,\ \ldots,\ 1]^T$ are vectors of appropriate sizes corresponding to the vector of Lagrangian multipliers $\lambda^{\mathcal{A}_i}$ and the vector of slack variables $s^{\mathcal{J}_i}$, respectively. Clearly, if the LP (5) has a feasible solution, then there exist feasible $z_{\mathcal{A}_i}$, $x_{\mathcal{A}_i}$, $\lambda^{\mathcal{A}_i}$, $s^{\mathcal{J}_i}$ satisfying the KKT conditions, and $\mathcal{A}_i$ is an optimal active set. On the other hand, infeasibility of the LP (5) implies that $\mathcal{A}_i$ is not an optimal active set, and $z_{\mathcal{A}_i}$, $CR_{\mathcal{A}_i}$ need not be computed. Moreover, if the LP (5) is also infeasible when only feasibility constraints are considered, i.e., when all constraints related to $\lambda^{\mathcal{A}_i}$ are discarded, then $\mathcal{A}_i$ represents an infeasible combination of active constraints.

Since checking the LP (5) for every candidate set by enumerating $\mathcal{P}'(\mathcal{Q})$ explicitly may be impractical even for relatively small values of m and q, [6] propose in addition the following pruning criterion, which reduces the number of candidate sets and makes the enumeration of $\mathcal{P}'(\mathcal{Q})$ implicit.

Criterion 3 (Pruning of candidate active sets) *If a candidate active set $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$*

(i) *leads to violation of the LICQ condition, or*
(ii) *represents an infeasible constraint combination,*

then $\mathcal{A}_i$ and all its supersets can be excluded from further consideration in the enumeration of $\mathcal{P}'(\mathcal{Q})$.

The first of these pruning conditions becomes apparent when considering the structure of the combinatorial enumeration process. Since the algorithm proceeds through the elements of $\mathcal{P}'(\mathcal{Q})$ in order of increasing cardinality, all full-rank subsets of a candidate set with LICQ failure will be covered automatically. Thus, the method of computing the control law and the critical regions in cases of LICQ failure by considering all full-rank subsets is, in a sense, inherently included in the combinatorial approach. Therefore, every candidate active set $\mathcal{A}_i$ for which $G^{\mathcal{A}_i}$ does not have full row rank can be discarded, and, since LICQ will of course also be violated for all $\mathcal{A}_j \supset \mathcal{A}_i$, the same holds for all its supersets. In such a case, however, there will be weakly active constraints ($\lambda_i = 0$) and overlapping critical regions may occur. See [3] for more details. The second pruning condition follows directly from the fact that an infeasible system of equations and inequalities, i.e., here the LP (5), will still be infeasible when some of the inequalities are treated as equations [6].

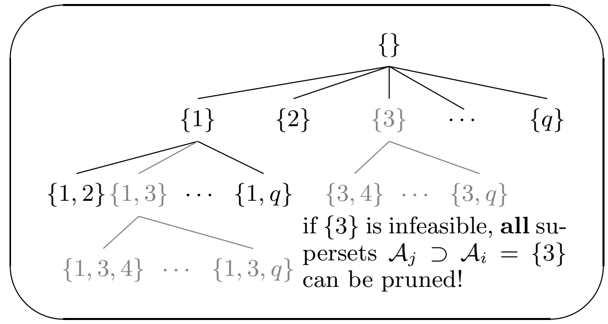

Fig. 1. Combinatorial enumeration strategy with pruning of infeasible candidate active sets in form of an active set tree. Note that pruning infeasible constraint combinations globally in the whole tree is crucial for achieving optimal efficiency in the enumeration.

Using these results, the combinatorial mpQP algorithm of [6] can be summarized as follows. See also Figure 1 for a graphical illustration of the combinatorial enumeration strategy and the involved pruning process.

Algorithm 1 (Combinatorial mpQP,[6])

1. *choose $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$ in order of increasing cardinality;*
2. *if $\mathcal{A}_i$ not pruned and $G^{\mathcal{A}_i}$ has full row rank, solve (5)*
 ∟ *if feasible, use (3) and (4) to construct $z_{\mathcal{A}_i}$ and $CR_{\mathcal{A}_i}$*
 ∟ *if infeasible, solve (5) without optimality constraints*
  ∟ *if infeasible, add all $\mathcal{A}_j \supset \mathcal{A}_i$ to the pruned sets;*
3. *return to **1.** until the whole set $\mathcal{P}'(\mathcal{Q})$ is explored.*

---

3 An improved combinatorial mpQP algorithm

In the previous section we have reviewed the combinatorial mpQP approach proposed by [6]. While it has some structural advantages over existing geometric approaches, the main disadvantage of the approach is given by its combinatorial complexity. In the following, we will present an improved combinatorial mpQP algorithm that is also based on the implicit enumeration approach but uses a simple matrix check to exclude infeasible constraint combinations; see also [4]. Furthermore, we present additional complexity reduction techniques which are based on exploiting symmetries in the mpQP problem formulation. The basic idea underlying both approaches is to use some of the geometric properties of the mpQP problem constraints for increasing the efficiency of the combinatorial enumeration.

3.1 *An infeasibility check based on the constraint polyhedron geometry: saturation matrix pruning*

Consider the mpQP problem (1). In the following, we want to exploit the fact that the constraints (1b) can also be represented as a non-parametrized polyhedron in the augmented (variable+parameter) space [8]:

$$\tilde{P} = \left\{ [z, x] \in \mathbb{R}^{m+n} \;\middle|\; \begin{bmatrix} G & -S \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \leq W \right\}. \tag{6}$$

We will show in this section how this augmented constraint polyhedron can be used to derive a necessary and sufficient condition for the identification of infeasible constraint combinations in the context of the combinatorial active set enumeration. Our approach is based on the the following theorem.

Theorem 4 (Infeasibility of candidate active sets) *Let $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$ be a candidate active set related to the constraint matrices G, W, and S of problem (1). Then $\mathcal{A}_i$ represents an infeasible combination of active constraints if and only if the constraints associated to $\mathcal{A}_i$ are not active together at any vertex $v_k \in \mathbb{R}^{m+n}$ of $\tilde{P}$, i.e., $\nexists\, v_k$ s.t. $\begin{bmatrix} G^{\mathcal{A}_i} & -S^{\mathcal{A}_i} \end{bmatrix} v_k - W^{\mathcal{A}_i} = 0$ for $k = 1, \ldots, n_v$.*

PROOF. If the constraint combination corresponding to $\mathcal{A}_i$ is not active at any vertex of $\tilde{P}$, it is also not active at any facet of $\tilde{P}$. Hence, the constraint hyperplanes related to $\mathcal{A}_i$ do not intersect in the feasible part of the augmented space $\mathbb{R}^{m+n}$, which means that $\mathcal{A}_i$ does not represent a feasible combination of active constraints. The reverse direction follows trivially. □

In order to exploit Theorem 4 in the combinatorial enumeration process, we make use of the saturation matrix $\mathcal{S}$ of the constraint polyhedron:

Definition 5 (Saturation matrix $\mathcal{S}$ of $\tilde{P}$ [15]) *As saturation matrix of the constraint polyhedron $\tilde{P}$ we denote the binary matrix $\mathcal{S} \in \{0,1\}^{n_v \times q}$ defined as*

$$\mathcal{S}_{kj} = \begin{cases} 1 & \text{if } \begin{bmatrix} G^j & -S^j \end{bmatrix} v_k - W^j = 0 \\ 0 & \text{if } \begin{bmatrix} G^j & -S^j \end{bmatrix} v_k - W^j \neq 0 \end{cases}, \tag{7}$$

where $k = 1, \ldots, n_v$, $j = 1, \ldots, q$. Hence, the entry $\mathcal{S}_{kj}$ indicates whether constraint j is active at vertex v_k of $\tilde{P}$.

Combining Theorem 4 and Definition 5, we can conclude that a candidate set $\mathcal{A}_i$ can only represent a feasible combination of active constraints if and only if there exists at least one row in the saturation matrix $\mathcal{S}$ that contains only nonzero elements in the columns related to the indices in $\mathcal{A}_i$. Hence, we can formulate the following corollary which allows to identify infeasible candidate active sets by performing a simple row sum check on $\mathcal{S}$.

Corollary 6 (Infeasibility condition for $\mathcal{A}_i$) *Let $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$ be a candidate active set and let $\mathcal{S}$ denote the saturation matrix of the constraint polyhedron $\tilde{P}$. Then, a necessary and sufficient condition for the infeasibility of $\mathcal{A}_i$ is given by*

$$\sum_{j \in \mathcal{A}_i} \mathcal{S}_{kj} < |\mathcal{A}_i| \quad \forall k \in \{1, \ldots, n_v\}, \tag{8}$$

where $|\mathcal{A}_i|$ denotes the cardinality of $\mathcal{A}_i$, i.e., the number of constraint indices in the candidate active set.

Note, however, that Theorem 4 and Corollary 6 only provide a condition for infeasibility of (5) and that optimality is not taken into account.
Based on this saturation matrix pruning condition, we propose the following combinatorial mpQP algorithm:

Algorithm 2 (Improved combinatorial mpQP)

---

1. *compute the saturation matrix $\mathcal{S}$ of the problem;*
2. *choose $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$ in order of increasing cardinality;*
3. *if $G^{\mathcal{A}_i}$ has full row rank, check condition (8)*
  ∟ *if $\mathcal{A}_i$ is identified as infeasible, go to **4.**;*
  ∟ *else, try to solve the LP (5)*
    ∟ *if feasible, use (3) and (4) to construct $z_{\mathcal{A}_i}$, $CR_{\mathcal{A}_i}$;*
    ∟ *if infeasible, go to **4.**;*
4. *return to **2.** until the whole set $\mathcal{P}'(\mathcal{Q})$ is explored.*

As in Algorithm 1, the enumeration proceeds through $\mathcal{P}'(\mathcal{Q})$ in the order of increasing cardinality and exploits Criterion 3 in order to reduce the number of candidate active sets. However, by making use of Corollary 6, all candidate sets that are related to infeasible combinations of active constraints can be excluded by simply checking the row sums of the saturation matrix. Furthermore, since infeasibility of the LP (5) will now in all cases arise from suboptimality, it is generally not worthwhile to solve the modified second LP and perform an additional pruning of infeasible candidate sets. This results in a further reduction in the number of LPs and eliminates the need for an explicit pruning mechanism. Moreover, in contrast to the algorithm proposed in [6], each candidate set $\mathcal{A}_i$ can now be checked independently from all other sets $\mathcal{A}_j \in \mathcal{P}'(\mathcal{Q})$, which would allow easy parallelization of the enumeration procedure in Algorithm 2. However, one disadvantage of the approach is that constructing and handling the saturation matrix $\mathcal{S}$ may become computationally demanding with increasing complexity of the constraint polyhedron $\tilde{P}$. While the dimension of $\tilde{P}$ grows linearly with m and n, there is to the authors knowledge no analytic expression to estimate the resulting number of vertices n_v. However, the upper bound on n_v will usually grow exponentially with the dimension and the number of constraints q, which definitely restricts the class of tractable problems. One way to construct the saturation matrix $\mathcal{S}$ is to compute all vertices of $\tilde{P}$ by using external vertex enumeration algorithms, e.g., as they are included in the CDD package [5]. On the other hand, some double description based packages like Polylib [15] compute the vertices, and in some cases even $\mathcal{S}$ itself, automatically when constructing the constraint polyhedron $\tilde{P}$. We do not discuss the construction of $\mathcal{S}$ and the underlying vertex enumeration problem here in detail and refer the reader to the relevant literature, e.g., [15].

3.2 Symmetry-based complexity reduction techniques

Since problem (1) is symmetric in the cost function, symmetries in the problem constraints will, in general, lead to symmetric mpQP solutions [12]. In this section, we want to present some results concerning symmetry exploitation in the context of combinatorial mpQP. On the one hand, we extend the existing results for completely symmetric constraints to the combinatorial approach, and, on the other hand, we introduce a concept of partial symmetry in mpQP and derive sufficient conditions for exploiting this kind of incomplete symmetry in the combinatorial enumeration.

Definition 7 (Symmetric mpQP problem) *An mpQP problem of the form (1) is called symmetric if all the involved constraints are symmetric, i.e., if they can be represented in the form*

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} z \leq \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} + \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} x \;, \tag{9}$$

with $G_2 = -G_1,\ W_2 = W_1,\ S_2 = -S_1$.

Note that since linear systems are inherently symmetry preserving, all mpQP problems related to linear MPC applications with symmetric input, state, or output constraints will in general be of this form. Moreover, also standard terminal set constraints based on LQR invariant sets will typically result in symmetric problem constraint formulations.

Definition 8 (Symmetric active set) *Let $\mathcal{A}_i$ be an active set for a symmetric mpQP problem. Then, we define as the symmetric active set $\mathcal{A}_i^-$ the set containing the row indices of those constraints which are symmetric to the constraints in $\mathcal{A}_i$, i.e.,*

$$G^{\mathcal{A}_i} = -G^{\mathcal{A}_i^-},\ W^{\mathcal{A}_i} = W^{\mathcal{A}_i^-},\ S^{\mathcal{A}_i} = -S^{\mathcal{A}_i^-}. \tag{10}$$

Lemma 9 (Symmetry in combinatorial mpQP) *For an mpQP problem that is symmetric in the sense of Definition 7, the LP (5) for the candidate set $\mathcal{A}_i^-$ has a feasible solution if and only if the LP for the symmetric candidate set $\mathcal{A}_i$ has a feasible solution; i.e., the LPs for $\mathcal{A}_i$ and $\mathcal{A}_i^-$ are equivalent.*

PROOF. Can be shown easily by inserting $x^- = -x$, $z^- = -z$ and the symmetry relations (10) into LP (5).□

Theorem 10 (Symmetric candidate set branches) *Consider a symmetric mpQP problem according to Definition 7. Let $i \in \mathcal{Q}$ denote an arbitrary constraint index and $i^- \in \mathcal{Q} \setminus i$ with $i^- > i$ the index of the corresponding symmetric constraint. Furthermore, let $\mathcal{B}_i$ and $\mathcal{B}_{i^-}$ be the branches of the active set tree related to the root nodes $\mathcal{A}_i = \{i\}$ and $\mathcal{A}_i^- = \{i^-\}$, respectively. If the constraints of the mpQP problem are arranged in such a way that it holds*

$$i_2 > i_1 \Rightarrow i_2^- > i_1^- \quad \forall\, i_1, i_2 \in \mathcal{Q},\ i_1 \neq i_2, \tag{11}$$

then, all candidate sets in $\mathcal{B}_{i^-}$ are symmetric to the candidate sets in $\mathcal{B}_i$ Hence, the complete branch $\mathcal{B}_{i^-}$ can be checked implicitly by checking all sets in branch $\mathcal{B}_i$.

PROOF. Branch $\mathcal{B}_{i^-}$ of the active set tree contains all ordered selections of the indices $\{i^-, i^-+1, \ldots, q\}$, all of which have i^- as the first set element. If we now construct the symmetric set for each candidate set in branch $\mathcal{B}_{i^-}$ by replacing all indices with their symmetric partner, condition (11) ensures that the index order will stay the same and that all symmetric sets will have the index i as the first element. Hence, they will, by definition, be contained in $\mathcal{B}_i$. □

Theorem 10 shows that only half of the active set branches need to be considered in the case of symmetric mpQP problem constraints. However, since the tree structure underlying the active set enumeration is not balanced, the way in which the symmetric constraints are arranged in the matrices G, W, and S will heavily influence the achievable effect of symmetry exploitation. Two possible constraint matrix configurations satisfying condition (11) are, for example, the ordering in two symmetric blocks, as in Equation (9), or in alternating symmetric pairs, i.e., $G^i = -G^{i+1}$, $S^i = -S^{i+1}$, $W^i = W^{i+1}$, $i = 1, 3, \ldots, q-1$. Since it makes the best use of the asymmetric tree structure, it is obvious that arranging the constraints in symmetric pairs allows to prune the maximal number of symmetric candidate sets and will result in the maximal complexity reduction of about 50 %, i.e., $n_{LP}^{\text{sym}} = \frac{1}{2}(n_{LP}+1)$ since $\mathcal{A}_0 = \{\ \}$ has no symmetric partner set (see [3] for more details). Consequently, this constraint matrix configuration is used for exploiting symmetries in the numerical examples in Section 4. Note that rearranging the constraints in the matrices G, W, and S does not change the analytic mpQP solution.

Of course, complete symmetry in the sense of Definition 7 cannot be assumed in general. Especially in the context of explicit linear MPC, situations can occur in which asymmetric box constraints on z and x have to be considered. In this case, the constraint matrices G and S may still have a symmetric structure, while only the matrix W is afflicted by the constraints asymmetry. In the following, we will present some results for this class of *partially symmetric* mpQP problems.

Definition 11 (Partial symmetry in mpQP) *We refer to an mpQP problem of the form (1) as partially symmetric if the involved constraints can be rewritten as*

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} z \le \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} + \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} x \, , \tag{12}$$

where $G_2 = -G_1$, $W_2 = W_1 + \delta W$, $S_2 = -S_1$. Clearly, the only difference to a completely symmetric problem in the sense of Definition 7 is the additional term δW.

Furthermore, we can define the symmetric set $\mathcal{A}_i^-$ to a candidate active set $\mathcal{A}_i$ similar to the case of complete symmetry, with the modification that for a partially symmetric mpQP problem it holds

$$G^{\mathcal{A}_i} = -G^{\mathcal{A}_i^-},\ W^{\mathcal{A}_i} = W^{\mathcal{A}_i^-} + \Delta W^{\mathcal{A}_i},\ S^{\mathcal{A}_i} = -S^{\mathcal{A}_i^-}, \tag{13}$$

where ΔW is defined as $\Delta W = [-\delta W, \delta W]^T$. Now, as the constraints are not completely symmetric, Lemma 9 does not hold and the LPs for $\mathcal{A}_i$ and $\mathcal{A}_i^-$ will, in general, not be equivalent. Note that it may be possible to recover the complete symmetry of the constraints (5) by a change of coordinates. However, in general this will lead to a loss of symmetry in the cost function $V_z(x)$. In the following we will derive a sufficient condition that allows to prove feasibility of the LP (5) for a candidate active set $\mathcal{A}_i^-$ based on computations related to the (partially) symmetric set $\mathcal{A}_i$.

Let us assume $\mathcal{A}_i$ is an optimal active set and we know the Lagrangian multipliers $\lambda^{\mathcal{A}_i}$ and the vector of slack variables $s^{\mathcal{J}_i}$ which satisfy the KKT conditions for feasible x and $z(x)$, i.e., $\lambda^{\mathcal{A}_i}$, $s^{\mathcal{J}_i}$, x, z solve the LP (5). Using the assumption $x^- = -x$ and the partial symmetry relations (13) between the constraint matrices, we can formulate similar KKT conditions for the symmetric candidate set $\mathcal{A}_i^-$:

$$\begin{aligned} Hz^- - (G^{\mathcal{A}_i})^T \lambda^{\mathcal{A}_i^-} &= 0 && \text{(14a)} \\ -G^{\mathcal{A}_i} z^- - S^{\mathcal{A}_i} x - (W^{\mathcal{A}_i} - \Delta W^{\mathcal{A}_i}) &= 0 && \text{(14b)} \\ -G^{\mathcal{J}_i} z^- - S^{\mathcal{J}_i} x - (W^{\mathcal{J}_i} - \Delta W^{\mathcal{J}_i}) + s^{\mathcal{J}_i^-} &= 0 && \text{(14c)} \\ \lambda^{\mathcal{A}_i^-} \ge 0,\ s^{\mathcal{J}_i^-} &\ge 0. && \text{(14d)} \end{aligned}$$

Since complete symmetry does not hold, the optimizer is not symmetric and we have

$$z^- = -z + \Delta z \overset{(5d)}{=} H^{-1}(G^{\mathcal{A}_i})^T \lambda^{\mathcal{A}_i} + \Delta z \, . \tag{15}$$

Inserting this into Equation (14a), we get for Δz

$$\Delta z = H^{-1}(G^{\mathcal{A}_i})^T (\lambda^{\mathcal{A}_i^-} - \lambda^{\mathcal{A}_i}). \tag{16}$$

Furthermore, inserting (15) and (16) into (14b) and exploiting $G^{\mathcal{A}_i} z - S^{\mathcal{A}_i} x - W^{\mathcal{A}_i} = 0$ yields

$$\lambda^{\mathcal{A}_i^-} = \lambda^{\mathcal{A}_i} + H^{-1}_{G^{\mathcal{A}_i}} \Delta W^{\mathcal{A}_i}, \tag{17}$$

where $H^{-1}_{G^{\mathcal{A}_i}} = (G^{\mathcal{A}_i} H^{-1} (G^{\mathcal{A}_i})^T)^{-1}$. In a similar way, we can insert the Equations (15) and (16) into Equation (14c) and exploit (5f) in order to write the vector of slack variables for $\mathcal{A}_i^-$ as

$$s^{\mathcal{J}_i^-} = s^{\mathcal{J}_i} + G^{\mathcal{J}_i} H^{-1} (G^{\mathcal{A}_i})^T H^{-1}_{G^{\mathcal{A}_i}} \Delta W^{\mathcal{A}_i} - \Delta W^{\mathcal{J}_i}. \tag{18}$$

In combination with the feasibility and optimality conditions $\lambda^{\mathcal{A}_i^-} \ge 0$, $s^{\mathcal{J}_i^-} \ge 0$ (see Equation (14d)), the obtained Equations (17) and (18) yield sufficient conditions for the optimality of the candidate set $\mathcal{A}_i^-$ in terms of its symmetric set $\mathcal{A}_i$.

Theorem 12 (Exploiting partial symmetry) *Consider the class of partially symmetric mpQP problems in the sense of Definition 11. Let $\mathcal{A}_i$ denote the considered candidate active set and $\mathcal{A}_i^-$ its symmetric set in the sense of partial symmetry. If $\mathcal{A}_i$ is an optimal active set and the vectors of Lagrangian multipliers $\lambda^{\mathcal{A}_i}$ and slack variables $s^{\mathcal{J}_i}$ in the feasible solution of the LP (5) satisfy*

$$\lambda^{\mathcal{A}_i} \ge -H^{-1}_{G^{\mathcal{A}_i}} \Delta W^{\mathcal{A}_i}, \tag{19}$$

$$s^{\mathcal{J}_i} \ge -G^{\mathcal{J}_i} H^{-1} (G^{\mathcal{A}_i})^T H^{-1}_{G^{\mathcal{A}_i}} \Delta W^{\mathcal{A}_i} + \Delta W^{\mathcal{J}_i}, \tag{20}$$

then $\mathcal{A}_i^-$ will also be an optimal active set and the corresponding LP needs not be solved.

PROOF. The conditions on $\lambda^{\mathcal{A}_i}$ and $s^{\mathcal{J}_i}$ guarantee by construction that there exist feasible $\lambda^{\mathcal{A}_i^-}$, $s^{\mathcal{J}_i^-}$ for LP (5) with $x_{\mathcal{A}_i^-} = -x_{\mathcal{A}_i}$, $z_{\mathcal{A}_i^-} = -z_{\mathcal{A}_i} + \Delta z_{\mathcal{A}_i}$. □

There are different ways to make use of Theorem 12 in Algorithm 1. One way is to simply check the conditions on $\lambda^{\mathcal{A}_i}$ and $s^{\mathcal{J}_i}$ for every candidate set $\mathcal{A}_i$ for which the LP (5) results in a feasible solution. However, the solution of (5) is not unique and the efficiency of this approach is limited by the assumption $x^- = -x$. Another possibility, which is used for the numerical example computations in the next section, is to include the conditions

(19) and (20) as additional constraints in the LP formulation. In this case, $\mathcal{A}_i$ and $\mathcal{A}_i^-$ can be checked simultaneously by an augmented LP, and the maximal number of symmetric optimal active sets will be found. Unfortunately, no conclusions can be drawn from infeasibility of the augmented LP, in which case the standard LPs for $\mathcal{A}_i$ and $\mathcal{A}_i^-$ have to be solved in addition. Hence, this approach is mainly suited for problems with a relatively small number of suboptimal candidate active sets. Note, however, that for $\Delta W = 0$, i.e., for the case of complete symmetry, the conditions (19) and (20) reduce to the redundant standard conditions $\lambda^{\mathcal{A}_i} \geq 0,\ s^{\mathcal{J}_i} \geq 0$.

4 Numerical Examples

In order to compare our algorithm with the one from [6], we implemented and tested Algorithm 1, Algorithm 2, and two additional versions of Algorithm 2 that combine the saturation matrix pruning with the symmetry exploitation techniques from Section 3.2. The saturation matrix $\mathcal{S}$ was constructed by computing the vertices of the constraint polyhedron with the $\mathbb{MPT}$ extreme point solver `extreme()` using CDD via cddmex [7]. Several mpQP problems have been solved, related to varying horizon linear MPC open-loop optimal control problems for the following two example systems. All computations were performed on a 3 GHz Dual Core PC with 8 GB RAM, running MATLAB 7.11 and $\mathbb{MPT}$ 2.6.3. More details and numerical results can be found in [3] and, for the saturation matrix case, in [4].

Example 1. Considered is the discrete-time double integrator system discussed by [13] with a discretization time of $T_s = 0.3\,\text{s}$. For this system, the linear MPC open-loop optimal control problem is formulated using a quadratic cost function with the weight matrices $Q = \text{diag}(1, 0),\ R = 1,\ P = P_{LQR}$ and the input and state constraints $|u| \leq 1,\ |x_2| \leq 0.8,\ x(t + N) \in \Omega_{LQR}$. Here, P_{LQR} has been computed from the algebraic Riccati equation and Ω_{LQR} is the LQR invariant set. The resulting mpQP problems are symmetric in the sense of Definition 7.

Example 2. As a second example, we considered the laboratory model helicopter described by [13], which is given in form of a linear state space model involving six system states and two inputs. The MPC optimization problems were formulated using $Q = \text{diag}(100, 100, 10, 10, 400, 200),\ R = I_{2\times2},\ P = P_{LQR}$ and the input constraints $-1 \leq u_i \leq 3,\ i = 1, 2$. The mpQP problems for this example are partially symmetric in the sense of Definition 11.

The numerical results for the two examples are presented in Table 1 and Table 2, respectively. Here, N denotes the horizon in the MPC problem formulation, n_r the number of critical regions in the state space partition, and n_{LP} the number of optimization problems of type (5) that were solved in the combinatorial enumeration process. The superscripts “cSym” and “pSym” refer to the algorithm versions exploiting complete and partial symmetry. As can be seen, the number of LPs is considerably reduced by using the proposed saturation matrix pruning criterion and exploiting the symmetries in the problem constraints. An exception is the helicopter example for $N = 5$. Exploiting partial symmetry with the augmented LP approach results here in an increased number of optimization problems since the number of additional LPs that are needed in cases where the augmented LP is infeasible exceeds the number of LPs that are saved by exploiting Theorem 12.

Table 1
Results for the double integrator example.

N	n_r	$n_{LP,\max}$	$n_{LP,\text{Alg.1}}$	$n_{LP,\text{Alg.2}}$	$n_{LP,\text{Alg.2}}^{\text{cSym}}$
1	11	15	13	13	7
2	33	172	131	77	39
3	57	1 794	631	383	192
4	83	17 902	6 695	1 733	867
5	111	174 437	30 717	7 569	3 785
6	135	1 676 116	263 503	32 017	16 009

Table 2
Results for the helicopter example.

N	n_r	$n_{LP,\max}$	$n_{LP,\text{Alg.1}}$	$n_{LP,\text{Alg.2}}$	$n_{LP,\text{Alg.2}}^{\text{pSym}}$
1	9	11	9	9	5
2	81	163	81	81	41
3	729	2 510	729	729	365
4	4 461	39 203	8 589	6 561	6 129
5	18 413	616 666	99 119	59 049	76 089

The achievable computation times depend, of course, on the speed of the computer system and the efficiency of the used LP solvers. However, we made the following qualitative observations. For small-scale problems, e.g., like the double integrator example, the effort of constructing and handling the saturation matrix is negligible, and the reduction in the computation time will, in general, almost be equivalent to the reduction in the number of LPs. Hence, concerning the results in Table 1, Algorithm 2 with symmetry exploitation is approximately up to 16 times faster than the algorithm of [6]. On the other hand, when considering more complex mpQP problems, e.g., like the helicopter example, computing all the vertices of the constraint polyhedron and constructing the saturation matrix becomes computationally more demanding, and the achievable benefit depends on the speed of the LP solver. If a fast solver, e.g., from the NAG Toolbox [11], is used, the time needed for the saturation matrix operations may even exceed the time that is saved by reducing the number of LPs. Of course, this point may be addressed by using a more efficient extreme point solver or a specialized library like Polylib for the construction of the saturation matrix $\mathcal{S}$. Moreover, for growing size of $\mathcal{S}$, it

might be helpful to exploit sparsity or to minimize the number of saturation matrix checks by pruning infeasible constraint combinations explicitly. However, up to now, these points have not been investigated in detail and may be considered as possible future work.
Another interesting point is to compare the performance of the combinatorial algorithms with the performance of the geometric mpQP solver that is included in the $\mathbb{MPT}$. Here, our benchmark tests showed that the original combinatorial approach is rather badly-suited for small-scale mpQP problems with a large number of constraints. In such situations, the proposed methods can achieve significant performance improvements, which may help to establish combinatorial mpQP as a real alternative to existing geometric approaches. On the other hand, the combinatorial mpQP algorithms seem to have significant speed and robustness advantages over the geometric $\mathbb{MPT}$ algorithm when considering moderately constrained higher-dimensional problems, e.g., like the helicopter example. More detailed results for different LP solvers can be found in [3].

5 Conclusion

In this paper, we have proposed an improved combinatorial mpQP algorithm that uses a saturation matrix pruning criterion for excluding infeasible candidate active sets, which allows to speed up the combinatorial enumeration process and eliminates the need for an explicit pruning mechanism. In addition, techniques have been presented that allow further complexity reduction by exploiting symmetries in the mpQP problem formulation. The results show the benefit of using geometric properties of the mpQP problem constraints for improving the efficiency of combinatorial mpQP. Interesting future topics could be to design and incorporate suboptimality-based pruning criteria or to exploit the decoupling character of the proposed pruning mechanism in a parallelized combinatorial mpQP algorithm.

Acknowledgements

We would like to thank the authors of [6] for giving us access to the MATLAB code of their algorithm. The third author acknowledges the support of the RCN Yggdrasil project 202632/V11.

References

[1] M. Baotić. An Efficient Algorithm for Multi-Parametric Quadratic Programming. Technical Report AUT02-05, ETH Zürich, 2002.

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

[3] C. Feller. A Non-Geometric Approach to Multi-Parametric Programming. Diploma Thesis IST-80, University of Stuttgart, 2011.

[4] C. Feller, T. A. Johansen, and S. Olaru. Combinatorial multi-parametric quadratic programming with saturation matrix based pruning. In *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, Hawaii, 2012. *accepted*.

[5] K. Fukuda. The cdd and cddplus Homepage. `http://www.inf.ethz.ch/personal/fukudak/cdd_home/`.

[6] A. Gupta, S. Bhartiya, and P. S. V. Nataraj. A Novel Approach to Multiparametric Quadratic Programming. *Automatica*, 47:2112–2117, September 2011.

[7] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), `http://control.ee.ethz.ch/~mpt/`, 2004.

[8] S. B. Olaru and D. Dumur. A Parametrized Polyhedra Approach for Explicit Constrained Predictive Control. In *Proc. of the 43rd IEEE Conference on Decision and Control*, volume 2, pages 1580–1585, 2004.

[9] M. M. Seron, G. C. Goodwin, and J. A. De Doná. Characterisation of receding horizon control for constrained linear systems. *Asian Journal of Control*, 5(2):271–286, 2003.

[10] J. Spjøtvold, E. C. Kerrigan, C. N. Jones, P. Tøndel, and T.A. Johansen. On the facet-to-facet property of solutions to convex parametric quadratic programs. *Automatica*, 42:2209–2214, 2006.

[11] The Numerical Algorithms Group Ltd; Oxford, UK. NAG Toolbox for MATLAB. Available: `http://www.nag.co.uk/`.

[12] P. Tøndel. *Constrained Optimal Control via Multiparametric Quadratic Programming*. PhD thesis, NTNU, Trondh., 2003.

[13] P. Tøndel, T. A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39:489–497, 2003.

[14] P. Tøndel, T. A. Johansen, and A. Bemporad. Further Results on Multiparametric Quadratic Programming. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3173–3178, Hawaii, 2003.

[15] D. K. Wilde. A Library for Doing Polyhedral Operations. Scientific Report, INRIA, 1993. Polylib is available at: `http://www.irisa.fr/polylib/`.