Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Dynamic programming for explicit linear MPC with point-symmetric constraints ⋆

R. Mitze and M. Mönnigmann *

* *Automatic Control and Systems Theory, Ruhr-Universität Bochum, Bochum, Germany. E-mail:* ruth.mitze@rub.de *and* martin.moennigmann@rub.de.

Abstract: The solution to a constrained linear-quadratic optimal control problem can be expressed as a set of active sets. We recently proposed an algorithm that constructs this set by iteratively increasing the horizon. In the present paper, we improve this algorithm for problems with point-symmetric constraints. This is done by showing that such problems can be reformulated as symmetric quadratic programs and by exploiting properties of symmetric quadratic programs in the algorithm. A considerable reduction of the computational effort can be achieved, which will be demonstrated with an example.

Keywords: constrained LQR, predictive control, combinatorial quadratic programming

1. INTRODUCTION

The solution of a constrained linear-quadratic optimal control problem (OCP) is a piecewise-affine feedback law (Bemporad et al., 2002; Seron et al., 2003). The number of affine pieces often drastically increases with the problem order and the horizon. Thus, solving such problems explicitly is challenging. Various approaches exist. Geometric approaches exploit the geometric structure of the solution (Bemporad et al., 2002; Tøndel et al., 2003; Baotić, 2002). They are competitive and therefore established in toolboxes (Herceg et al., 2013; Bemporad, 2004). Combinatorial approaches (also referred to as implicit enumeration techniques) exploit that every affine piece in the solution is defined by a unique active set. Combinatorial algorithms calculate the set of active sets that defines the solution (Gupta et al., 2011; Feller et al., 2013; Oberdieck et al., 2017; Ahmadi-Moshkenani et al., 2018; Herceg et al., 2015). Dynamic programming approaches have been used to determine the explicit solution by iteratively increasing the horizon (Muñoz de la Peña et al., 2004; Mare and De Dona, 2007). An approach that combines the combinatorial approach with dynamic programming was recently presented (Mitze and Mönnigmann, 2019). The approach is based on the relationship between the solutions for different horizons (Mönnigmann, 2019). Its procedure also allows to detect when the solution for a finite horizon equals the solution for the infinite horizon as the finite horizon is increased iteratively.

Symmetries in linear-quadratic OCPs can be used to reduce the memory requirements of the explicit solution (Danielson, 2014). We exploit symmetries to reduce the computational effort for determining the explicit solution. More precisely, we improve the dynamic programming algorithm proposed in Mitze and Mönnigmann (2019) for linear-quadratic OCPs with point-symmetric constraints. We show that this problem class can be reformulated as symmetric quadratic programs (QPs) and exploit properties of symmetric QPs (Feller et al., 2013).

Section 2 introduces some facts about linear-quadratic OCPs, symmetric QPs, and the dynamic programming algorithm. Sections 3 and 4 propose a new algorithm and illustrate it with an example, respectively. Conclusions are given in Sect. 5.

1.1 Notation

Consider a matrix $M \in \mathbb{R}^{a\times b}$ and an ordered set $\mathcal{M} \subseteq \{1, ..., a\}$. Let $M_{\mathcal{M}} \in \mathbb{R}^{|\mathcal{M}|\times b}$ denote the submatrix of M containing all rows indicated by $\mathcal{M}$. Furthermore, let $\oplus$ denote the Minkowski addition.

2. PROBLEM STATEMENT AND PRELIMINARIES

The constrained linear-quadratic OCP treated in the present paper reads

$$\min_{U,X} \quad x(N)^T P x(N) + \sum_{k=0}^{N-1} \big(x(k)^T Q x(k) + u(k)^T R u(k)\big) \tag{1a}$$

$$\text{s.t.} \quad x(k+1) = Ax(k) + Bu(k),\ k = 0, ..., N-1 \tag{1b}$$

$$x(k) \in \mathcal{X},\ k = 0, ..., N-1 \tag{1c}$$

$$u(k) \in \mathcal{U},\ k = 0, ..., N-1 \tag{1d}$$

$$x(N) \in \mathcal{T}, \tag{1e}$$

where $x(0) \in \mathbb{R}^n$ is given, $U = \big(u^T(0), ..., u^T(N-1)\big)^T \in \mathbb{R}^{Nm}$ and $X = \big(x^T(1), ..., x^T(N)\big)^T \in \mathbb{R}^{Nn}$ collect the inputs $u(k) \in \mathbb{R}^m$ and states $x(k) \in \mathbb{R}^n$, respectively, $Q \in \mathbb{R}^{n\times n}$, $Q \succeq 0$ and $R \in \mathbb{R}^{m\times m}$, $R \succ 0$ are the weighting matrices for states and inputs, respectively, $N \in \mathbb{N}$ is the horizon, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ define the discrete-time time-invariant system, (A, B) is stabilizable, $\mathcal{U}$ and $\mathcal{X}$ are compact full-dimensional polytopes that are point-symmetric with respect to the origin ($-u \in \mathcal{U} \Leftrightarrow u \in \mathcal{U}$,

⋆ Support by the German Federal Ministry for Economic Affairs and Energy under grant 0324125C is gratefully acknowledged.

$-x \in \mathcal{X} \Leftrightarrow x \in \mathcal{X}$) and that contain the origin in their interiors, and P and K are the optimal cost function matrix and optimal feedback matrix, respectively, of the unconstrained infinite-horizon problem which implies $P \succ 0$. The terminal set $\mathcal{T}$ is chosen to be the largest possible set with the properties $\mathcal{T} \subseteq \mathcal{X}$, $Kx \in \mathcal{U}$ and $(A + BK)x \in \mathcal{T}$ for all $x \in \mathcal{T}$.

We assume the constraints to be ordered by increasing stage

$$\begin{array}{c} u(0) \in \mathcal{U}, \quad x(0) \in \mathcal{X}, \\ \vdots \\ u(N-1) \in \mathcal{U}, \quad x(N-1) \in \mathcal{X}, \\ x(N) \in \mathcal{T} \end{array} \tag{2}$$

with $k = 0, \ldots, N$. Let $q_{\mathcal{UX}}$ and $q_{\mathcal{T}}$ denote the number of input and state constraints and the number of terminal constraints, respectively. This implies the total number of constraints amounts to $q = Nq_{\mathcal{UX}} + q_{\mathcal{T}}$ for horizon N. We define the sets $\mathcal{Q} = \{1, ..., q\}$ and $\mathcal{Q}_0 = \{1, ..., q_{\mathcal{UX}}\}$.

By substituting (1b), (1) can be reformulated as the QP

$$\begin{array}{ll} \min\limits_{U} & \frac{1}{2}x(0)^T Y x(0) + x(0)^T F U + \frac{1}{2} U^T H U \\ \text{s.t.} & GU \leq Ex(0) + w, \end{array} \tag{3}$$

with $Y \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times Nm}$, $H \in \mathbb{R}^{Nm \times Nm}$, $G \in \mathbb{R}^{q \times Nm}$, $E \in \mathbb{R}^{q \times n}$, $w \in \mathbb{R}^q$, and where (3) inherits the constraint order (2). The assumptions on (1) imply $H \succ 0$ (Bemporad et al., 2002).

For any $x(0)$ such that (1) has a solution, let $\mathcal{A}(x(0)) = \{i \in \mathcal{Q} | G_{\{i\}} U = w_{\{i\}} + E_{\{i\}} x(0)\}$ and $\mathcal{I}(x(0)) = \mathcal{Q} \backslash \mathcal{A}(x(0))$ refer to the optimal active set and its corresponding inactive set, respectively. We call $\mathcal{A}(x(0))$ an *optimal* active set to distinguish it more clearly from *candidate* active sets introduced in Sect. 2.1.

When solving (3) as a parametric program with parameter $x(0)$, the resulting optimal control law is a continuous piecewise affine function of $x(0)$ on a polytopic partition (Bemporad et al., 2002, Sect. 4.1). Let $\mathcal{M}_N$ refer to the set of all optimal active sets $\mathcal{A}$ such that $G_{\mathcal{A}}$ has full row rank and such that $\mathcal{A}$ defines a full-dimensional polytope. Let $\mathcal{S}_N$ refer to all optimal active sets including those that define lower-dimensional polytopes such as facets and vertices ($\mathcal{S}_N \supseteq \mathcal{M}_N$).

2.1 Symmetric QP

We call a QP (3) symmetric, if $G_{\mathcal{M}} = -G_{\mathcal{N}}$, $E_{\mathcal{M}} = -E_{\mathcal{N}}$ and $w_{\mathcal{M}} = w_{\mathcal{N}}$ holds for some sets $\mathcal{M}$ and $\mathcal{N}$ such that $\mathcal{M} \cup \mathcal{N} = \mathcal{Q}$, $\mathcal{M} \cap \mathcal{N} = \emptyset$. In the same sense we call two constraints i and j symmetric, if $G_i = -G_j$, $E_i = -E_j$ and $w_i = w_j$, and we call two active sets $\mathcal{A}_i$ and $\mathcal{A}_j$ symmetric, if $G_{\mathcal{A}_i} = -G_{\mathcal{A}_j}$, $E_{\mathcal{A}_i} = -E_{\mathcal{A}_j}$ and $w_{\mathcal{A}_i} = w_{\mathcal{A}_j}$. All definitions stated here are compatible with those stated by Feller et al. (2013), who proved the following lemma.

Lemma 1. (Feller et al. (2013, Lem. 9)). Consider a symmetric QP. An active set is optimal if and only if its associated symmetric active set is optimal.

2.2 Iterative construction of $\mathcal{M}_N$ for non-symmetric problems

The dynamic programming algorithm presented in Mitze and Mönnigmann (2019) solves constrained linear-quadratic OCPs by iteratively increasing the horizon. The essential relationship between the solutions for successive horizons N and $N+1$ is stated in Lem. 2 (see also Mönnigmann (2019)). Let $\mathcal{P}(\mathcal{Q}_0)$ refer to the power set of $\mathcal{Q}_0$.

Lemma 2. (Mitze and Mönnigmann (2019, Cor. 1)). Consider an OCP (1) and assume its constraints are ordered as in (2). Assume we know $\mathcal{S}_N$. Then

$$\mathcal{S}_{N+1} = \mathcal{R}^{(1)} \cup \mathcal{R}^{(2)}, \tag{4}$$

with

$$\mathcal{R}^{(1)} = \{\mathcal{A} \in \mathcal{S}_N | \mathcal{A} \subseteq \{1, ..., Nq_{\mathcal{UX}}\}\}, \tag{5a}$$

$$\mathcal{R}^{(2)} \subseteq \left\{\mathcal{A}_j \cup (\mathcal{A}_l \oplus \{q_{\mathcal{UX}}\}) | \mathcal{A}_j \in \mathcal{P}(\mathcal{Q}_0),\, \mathcal{A}_l \in \tilde{\mathcal{S}}_N\right\}, \tag{5b}$$

where $\tilde{\mathcal{S}}_N$ contains all elements of $\mathcal{S}_N$ that have at least one active constraint in stage $k = N-1$ or $k = N$, i.e.,

$$\tilde{\mathcal{S}}_N = \{\mathcal{A} \in \mathcal{S}_N | \mathcal{A} \not\subseteq \{1, ..., (N-1)q_{\mathcal{UX}}\}\}. \tag{6}$$

An algorithm that implements Lem. 2 is given in Alg. 2 in Mitze and Mönnigmann (2019). It results from Alg. 2 in the present paper, if the superscript "red." is omitted. Specifically, (5a) is implemented in lines 4,5 and the superset in (5b) is implemented in lines 8,9 of Alg. 2. The superset is reduced to $\mathcal{R}^{(2)}$ by testing all its elements for optimality. For clarity, we call these elements *candidate* active sets, since they are not known to be optimal. A candidate active set $\mathcal{A}$ is optimal, if the linear program (LP)

$$\min_{U, x(0), \lambda_{\mathcal{A}}, s_{\mathcal{I}}, t} \quad -t \tag{7a}$$

$$\text{s.t.} \quad F^T x(0) + HU + (G_{\mathcal{A}})^T \lambda_{\mathcal{A}} = 0, \tag{7b}$$

$$te_2 \leq \lambda_{\mathcal{A}}, \tag{7c}$$

$$G_{\mathcal{A}} U - E_{\mathcal{A}} x(0) - w_{\mathcal{A}} = 0, \tag{7d}$$

$$G_{\mathcal{I}} U - E_{\mathcal{I}} x(0) - w_{\mathcal{I}} + s_{\mathcal{I}} = 0, \tag{7e}$$

$$te_1 \leq s_{\mathcal{I}}, \tag{7f}$$

$$t \geq 0, \tag{7g}$$

with column vectors $e_i = (1 \cdots 1)^T$, $i = 1, 2$ of appropriate sizes, Lagrangian multipliers $\lambda_{\mathcal{A}}$, and slack variables $s_{\mathcal{I}}$, has a solution (Gupta et al., 2011, Sect. 3.1). Since all elements in $\mathcal{S}_{N+1}$ and therefore all elements in $\mathcal{R}^{(2)}$ are optimal by definition of $\mathcal{S}_N$, only optimal candidate active sets are added to $\mathcal{S}_{N+1}$ (lines 12,13). Furthermore, candidates that result in $t = 0$ in (7) are collected in the set $\mathcal{S}_N^{\text{degen.}}$. $\mathcal{S}_{N+1}^{\text{degen.}}$ then consists of those $\mathcal{A} \in \mathcal{S}_N^{\text{degen.}}$ that are copied with (5a) (lines 6,7) and candidate active sets such that the solution to (7) is $t = 0$ (lines 14,15). We call an active set $\mathcal{A}$ *feasible* (resp. *infeasible*), if the LP (7) without (7b) and (7c) has a solution (resp. has no solution) (Gupta et al., 2011, Sect. 3.2). If a candidate active set $\mathcal{A}$ is infeasible, every $\mathcal{A}' \supset \mathcal{A}$ is also infeasible (Gupta et al., 2011, Thm. 1). Since (7) without (7b) and (7c) involves a subset of the constraints of (7), an $\mathcal{A}$ that is infeasible is not optimal. Hence, any candidate active set that is a superset of a known infeasible active set can be disregarded (line 10). This is referred to as pruning in the remainder of the paper. Candidate active sets that are not optimal are tested for feasibility (line 17). Infeasible active sets are stored in $\mathcal{S}_N^{\text{pruned}}$ (lines 18,19) in order to be able to dismiss supersets that appear later. Pruning has the greatest impact if candidates are tested in the order of increasing cardinality. The algorithm for determining $\mathcal{S}_1$ is given in Mitze and Mönnigmann (2019, Alg. 3).

The overall dynamic programming algorithm for determining the solution $\mathcal{M}_N$ for a horizon $N_{\max}$ for non-symmetric QP is given in Mitze and Mönnigmann (2019, Alg. 4). It results from Alg. 3 in the present paper when the superscript "red." and lines 13,14 and 17,18 are omitted. The approach terminates if $N_{\max}$ has been reached or if an N such that the solution $\mathcal{S}_N$ equals the infinite-horizon solution $\lim_{N\to\infty} S_N$ has been found. This is the case if $\tilde{\mathcal{S}}_{N+1} = \emptyset$ (lines 5,6) (Mitze and Mönnigmann, 2019, Prop. 1). Algorithm 3 reduces $\mathcal{S}_N$ to $\mathcal{M}_N$ by discarding those active sets $\mathcal{A}$ that define lower-dimensional polytopes and those such that $G_{\mathcal{A}}$ does not have full row rank. Polytopes of active sets such that $G_{\mathcal{A}}$ is not of full row rank are covered by the polytopes defined by the active sets in $\mathcal{M}_N$ (Ahmadi-Moshkenani et al., 2018, Sect. 3), therefore these active sets are not required (line 9). An optimal active set $\mathcal{A}$ that defines a lower dimensional polytope either has $G_{\mathcal{A}}$ that is not of full row rank or $G_{\mathcal{I}}U < w_{\mathcal{I}} + E_{\mathcal{I}}x(0)$ or $\lambda_{\mathcal{A}} > 0$ does not hold (Tøndel et al., 2003, Thm. 2). Active sets such that $G_{\{i\}}U = w_{\{i\}} + E_{\{i\}}x(0)$ holds for an $i \in \mathcal{I}$ or $\lambda_{\{j\}} = 0$ holds for a $j \in \mathcal{A}$ are collected in $\mathcal{S}_N^{\text{degen.}}$. These active sets are only added to $\mathcal{M}_N$ if the polytope defined by them (see e.g. Jost et al. (2015, Lem. 2)) is full-dimensional (lines 10-12).

3. ITERATIVE CONSTRUCTION OF $\mathcal{M}_N$ FOR SYMMETRIC PROBLEMS

We will show in Cor. 4 that a linear-quadratic OCP (1) where $\mathcal{U}$ and $\mathcal{X}$ are point-symmetric with respect to the origin can be reformulated as a symmetric QP. We need Lem. 3 as a preparation.

Lemma 3. If $\mathcal{U}$ and $\mathcal{X}$ in (1) are point-symmetric with respect to the origin, then $\mathcal{T}$ is point-symmetric with respect to the origin.

Proof. The terminal set $\mathcal{T}$ introduced in Sect. 2 is formally defined by

$$\mathcal{T} = \{x \in \mathcal{X} | (A+BK)^k x \in \mathcal{X}_{\mathcal{U}},\, k \geq 0\}, \tag{8}$$

where $\mathcal{X}_{\mathcal{U}} = \{x \in \mathcal{X} | Kx \in \mathcal{U}\}$ (see, e.g., Sznaier and Damborg (1987)). First note $\mathcal{X}_{\mathcal{U}}$ is point-symmetric with respect to the origin (pso), which can be seen as follows. For any $x \in \mathcal{X}_{\mathcal{U}}$ we have $x \in \mathcal{X}$ and $Kx \in \mathcal{U}$ by definition of $\mathcal{X}_{\mathcal{U}}$. Since $\mathcal{X}$ and $\mathcal{U}$ are pso, it follows that $-x \in \mathcal{X}$ and $-Kx = K(-x) \in \mathcal{U}$, which yields $-x \in \mathcal{X}_{\mathcal{U}}$. The point-symmetry of (8) can be established in the same fashion. For any $x \in \mathcal{T}$, we have $x \in \mathcal{X}$ and $(A+BK)^k x \in \mathcal{X}_{\mathcal{U}}$ for all $k \geq 0$ by (8). This implies $-x \in \mathcal{X}$ and $(A+BK)^k(-x) \in \mathcal{X}_{\mathcal{U}}$ since $\mathcal{X}$ and $\mathcal{X}_{\mathcal{U}}$ are pso. The last two relations imply $-x$ respects the conditions that define $\mathcal{T}$ in (8). Since $x \in \mathcal{T}$ was arbitrary, the claim follows. □

Corollary 4. If $\mathcal{U}$ and $\mathcal{X}$ in (1) are point-symmetric with respect to the origin, then the associated QP in (3) is symmetric.

Proof. We need to show that for every constraint i in (3) there exists a constraint j such that $G_i = -G_j$, $E_i = -E_j$ and $w_i = w_j$. Since $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{T}$ are polytopes, they are defined by an intersection of a finite number of halfspaces. Since $\mathcal{X}$ and $\mathcal{U}$ are symmetric by assumption and $\mathcal{T}$ is symmetric according to Lem. 3, there exist defining halfspaces that are symmetric. More precisely, there exist $T^{\mathcal{U}}$, $T^{\mathcal{X}}$, $T^{\mathcal{T}}$ and $d^{\mathcal{U}}$, $d^{\mathcal{X}}$, $d^{\mathcal{T}}$ such that

$$\begin{aligned} \mathcal{U} &= \left\{ u \in \mathbb{R}^m | \begin{pmatrix} T^{\mathcal{U}} \\ -T^{\mathcal{U}} \end{pmatrix} u \leq \begin{pmatrix} d^{\mathcal{U}} \\ d^{\mathcal{U}} \end{pmatrix} \right\}, \\ \mathcal{X} &= \left\{ x \in \mathbb{R}^n | \begin{pmatrix} T^{\mathcal{X}} \\ -T^{\mathcal{X}} \end{pmatrix} x \leq \begin{pmatrix} d^{\mathcal{X}} \\ d^{\mathcal{X}} \end{pmatrix} \right\}, \\ \mathcal{T} &= \left\{ x \in \mathbb{R}^n | \begin{pmatrix} T^{\mathcal{T}} \\ -T^{\mathcal{T}} \end{pmatrix} x \leq \begin{pmatrix} d^{\mathcal{T}} \\ d^{\mathcal{T}} \end{pmatrix} \right\}. \end{aligned} \tag{9}$$

Substituting (1b) into the remaining constraints of (1) yields

$$\begin{aligned} \begin{pmatrix} 0^{m\times km} & I^m & 0^{m\times(N-k-1)m} \end{pmatrix} U &\in \mathcal{U}, \\ x(0) &\in \mathcal{X}, \\ \begin{pmatrix} A^{k-1}B & \cdots & B & 0^{n\times(N-k)n} \end{pmatrix} U + A^k x(0) &\in \mathcal{X}, \\ \begin{pmatrix} A^{N-1}B & \cdots & B \end{pmatrix} U + A^N x(0) &\in \mathcal{T}, \end{aligned} \tag{10}$$

for all $k = 0, \ldots, N-1$, where I and 0 are the identity and zero matrices, respectively, with the obvious dimensions. Substituting (10) into (9) yields

$$\begin{aligned} &\begin{pmatrix} T^{\mathcal{U}} \\ -T^{\mathcal{U}} \end{pmatrix} \begin{pmatrix} 0^{m\times km} & I^m & 0^{m\times(N-k-1)m} \end{pmatrix} U \leq \begin{pmatrix} d^{\mathcal{U}} \\ d^{\mathcal{U}} \end{pmatrix}, \\ &\begin{pmatrix} T^{\mathcal{X}} \\ -T^{\mathcal{X}} \end{pmatrix} x(0) \leq \begin{pmatrix} d^{\mathcal{X}} \\ d^{\mathcal{X}} \end{pmatrix}, \\ &\begin{pmatrix} T^{\mathcal{X}} \\ -T^{\mathcal{X}} \end{pmatrix} \left[\begin{pmatrix} A^{k-1}B & \cdots & B & 0^{n\times(N-k)n} \end{pmatrix} U + A^k x(0) \right] \\ &\qquad \leq \begin{pmatrix} d^{\mathcal{X}} \\ d^{\mathcal{X}} \end{pmatrix}, \\ &\begin{pmatrix} T^{\mathcal{T}} \\ -T^{\mathcal{T}} \end{pmatrix} \left[\begin{pmatrix} A^{N-1}B & \cdots & B \end{pmatrix} U + A^N x(0) \right] \leq \begin{pmatrix} d^{\mathcal{T}} \\ d^{\mathcal{T}} \end{pmatrix}, \end{aligned} \tag{11}$$

for all $k = 0, \ldots, N-1$. By moving all terms that depend on $x(0)$ to the right hand side in (11), the form $GU \leq Ex(0) + w$ can be obtained. Because (11) and $GU \leq Ex(0) + w$ inherit the alternating rows from (9), there exists, for every constraint i, a constraint j such that $G_i = -G_j$, $E_i = -E_j$ and $w_i = w_j$. □

The dynamic programming algorithm for the construction of all active sets can be simplified for symmetric QPs. Essentially, we only need to construct one active set for each pair of two symmetric active sets. Let the reduced solution $\mathcal{S}_N^{\text{red.}}$ contain all $\mathcal{A} \in \mathcal{S}_N$, but only one of two symmetric active sets. Corollary 5 results from applying Lem. 2 to the symmetric case.

Corollary 5. Consider a symmetric OCP (1) and assume its constraints are ordered as in (2). Assume we know $\mathcal{S}_N^{\text{red.}}$. Then

$$\mathcal{S}_{N+1}^{\text{red.}} = \mathcal{R}^{(1),\text{red.}} \cup \mathcal{R}^{(2),\text{red.}},$$

with

$$\mathcal{R}^{(1),\text{red.}} = \{\mathcal{A} \in \mathcal{S}_N^{\text{red.}} | \mathcal{A} \subseteq \{1, ..., Nq_{\mathcal{U}\mathcal{X}}\}\}, \tag{12a}$$

$$\mathcal{R}^{(2),\text{red.}} \subseteq \left\{ \mathcal{A}_j \cup (\mathcal{A}_l \oplus \{q_{\mathcal{U}\mathcal{X}}\} | \mathcal{A}_j \in \mathcal{P}(\mathcal{Q}_0),\, \mathcal{A}_l \in \tilde{\mathcal{S}}_N^{\text{red.}} \right\}, \tag{12b}$$

where $\tilde{\mathcal{S}}_N^{\text{red.}}$ contains all elements of $\mathcal{S}_N^{\text{red.}}$ that have at least one active constraint in stage $k = N-1$ or $k = N$, i.e.,

$$\tilde{\mathcal{S}}_N^{\text{red.}} = \{\mathcal{A} \in \mathcal{S}_N^{\text{red.}} | \mathcal{A} \not\subseteq \{1, ..., (N-1)q_{\mathcal{UX}}\}\}.$$

Proof. $\mathcal{S}_{N+1}$ is the union of the sets $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$ (4), where all $\mathcal{A} \in \mathcal{R}^{(1)}$ have inactive constraints in stages N and $N+1$ (5a) and all $\mathcal{A} \in \mathcal{R}^{(2)}$ have at least one active constraint in stages N or $N+1$ (5b). By definition of the reduced solution, $\mathcal{S}_{N+1}^{\text{red.}}$ contains all elements from $\mathcal{S}_{N+1}$, but only one of two symmetric active sets.

Symmetric constraints appear within the constraints of one stage, see (11). Hence, symmetric active sets are either both part of $\mathcal{R}^{(1)}$ or both part of $\mathcal{R}^{(2)}$. Let $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ contain all elements of $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$, respectively, but only one of two symmetric active sets. It follows that $\mathcal{S}_{N+1}^{\text{red.}} = \mathcal{F}^{(1)} \cup \mathcal{F}^{(2)}$.

It remains to show that $\mathcal{F}^{(1)} = \mathcal{R}^{(1),\text{red.}}$ and $\mathcal{F}^{(1)} = \mathcal{R}^{(2),\text{red.}}$. $\mathcal{F}^{(1)}$ contains all elements of

$$\mathcal{R}^{(1)} = \{\mathcal{A} | \mathcal{A} \subseteq \{1, ..., Nq_{\mathcal{UX}}\}, \mathcal{A} \in \mathcal{S}_N\},$$

but only one of two symmetric active sets. Removing one of two symmetric active sets from $\mathcal{S}_N$, i.e., replacing $\mathcal{S}_N$ by $\mathcal{S}_N^{\text{red.}}$, yields

$$\begin{aligned}\mathcal{F}^{(1)} &= \{\mathcal{A} | \mathcal{A} \subseteq \{1, ..., Nq_{\mathcal{UX}}\}, \mathcal{A} \in \mathcal{S}_N^{\text{red.}}\} \\ &= \mathcal{R}^{(1),\text{red.}}.\end{aligned}$$

Accordingly, $\mathcal{F}^{(2)}$ results from replacing $\mathcal{S}_N$ by $\mathcal{S}_N^{\text{red.}}$ in

$$\begin{aligned}\mathcal{R}^{(2)} = \{\mathcal{A} | \mathcal{A} = \mathcal{A}_j \cup (\mathcal{A}_l \oplus \{q_{\mathcal{UX}}\}), \mathcal{A}_j \in \mathcal{P}(\mathcal{Q}_0), \\ \mathcal{A}_l \not\subseteq \{1, ..., (N-1)q_{\mathcal{UX}}\}, \mathcal{A}_l \in \mathcal{S}_N\},\end{aligned}$$

which yields

$$\begin{aligned}\mathcal{F}^{(2)} = \{\mathcal{A} | \mathcal{A} = \mathcal{A}_j \cup (\mathcal{A}_l \oplus \{q_{\mathcal{UX}}\}), \mathcal{A}_j \in \mathcal{P}(\mathcal{Q}_0), \\ \mathcal{A}_l \not\subseteq \{1, ..., (N-1)q_{\mathcal{UX}}\}, \mathcal{A}_l \in \mathcal{S}_N^{\text{red.}}\}.\end{aligned}$$

Introducing $\tilde{\mathcal{S}}_N^{\text{red.}} = \{\mathcal{A} \in \mathcal{S}_N^{\text{red.}} | \mathcal{A} \not\subseteq \{1, ..., (N-1)q_{\mathcal{UX}}\}\}$, this becomes

$$\begin{aligned}\mathcal{F}^{(2)} &= \{\mathcal{A} | \mathcal{A} = \mathcal{A}_j \cup (\mathcal{A}_l \oplus \{q_{\mathcal{UX}}\}), \mathcal{A}_j \in \mathcal{P}(\mathcal{Q}_0), \\ &\qquad \mathcal{A}_l \in \tilde{\mathcal{S}}_N^{\text{red.}}\} \\ &= \mathcal{R}^{(2),\text{red.}}.\end{aligned}$$

□

3.1 Implementational aspects

The algorithm presented in this paper proceeds analogously to the algorithm presented in Sect. 2.2 from Mitze and Mönnigmann (2019), but determines and stores only one out of each pair of optimal active sets. The set of all optimal active sets is then constructed in the last step only.

In order to compute the reduced solution for the initial horizon $\mathcal{S}_1^{\text{red.}}$, let the reduced power set $\mathcal{P}^{\text{red.}}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$ contain all $\mathcal{A} \in \mathcal{P}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$, but only one of two symmetric active sets. Clearly, $\mathcal{S}_1^{\text{red.}} \subseteq \mathcal{P}^{\text{red.}}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$ and hence, the elements of

$$\mathcal{P}^{\text{red.}}(\{1, \ldots, q_{\mathcal{UX}} + q_{\mathcal{T}}\})$$

are the candidate active sets for the initial horizon (line 2 in Alg. 1). The procedure to test these candidates and develop the reduced solution for the initial horizon is not changed. It is shown in Alg. 1.

Algorithm 1: Determination of $\mathcal{S}_1^{\text{red.}}$

Initialization: set $\mathcal{S}_1^{\text{red.}} = \emptyset$, $\mathcal{S}_1^{\text{degen.}} = \emptyset$ and $S_1^{\text{pruned}} = \emptyset$
for *every* $\mathcal{A}_i \in \mathcal{P}^{\text{red.}}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$ *by incr. cardinality* **do**
  if $\mathcal{A}_i \not\supseteq \tilde{\mathcal{A}}$ *for all* $\tilde{\mathcal{A}} \in \mathcal{S}_1^{\text{pruned}}$ **then**
    solve (7) for QP with horizon 1
    if *solution exists* **then**
      add $\mathcal{A}_i$ to $\mathcal{S}_1^{\text{red.}}$
      if *solution* $t = 0$ **then**
        add $\mathcal{A}_i$ to $\mathcal{S}_1^{\text{degen.}}$
    else
      solve (7) without (7b) and (7c) for QP with horizon 1
      if *no solution exists* **then**
        add $\mathcal{A}_i$ to S_1^{pruned}
Output: $\mathcal{S}_1^{\text{red.}}$, $\mathcal{S}_1^{\text{degen.}}$

We illustrate the power set and the reduced power set with the combinatorial tree in Fig. 1 for an artificial symmetric QP with 4 constraints. The combinatorial tree shown here contains all active sets in $\mathcal{P}(\{1, ..., 4\})$. The constraints are ordered in symmetric pairs such that symmetric constraints are next to each other. The active sets $\{\}$, $\{1, 2\}$, $\{3, 4\}$ and $\{1, 2, 3, 4\}$ are special in the sense that all symmetric constraints are either both active or both inactive therefore they are equal to their symmetric active set. $\mathcal{P}^{\text{red.}}(\{1, ..., 4\})$ can be constructed if every second branch that emanates from an active set that is equal to its symmetric set is omitted. $\mathcal{P}^{\text{red.}}(\{1, ..., 4\})$ then contains all active sets that are shown in black.

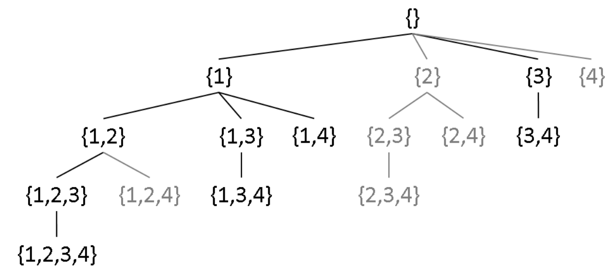


Fig. 1. Combinatorial tree for $q = 4$ constraints. The constraints are ordered in symmetric pairs, hence, constraints 1 and 2 and constraints 3 and 4 are symmetric, respectively.

Corollary 5 describes how the reduced solution for the increased horizon $\mathcal{S}_{N+1}^{\text{red.}}$ is related to the reduced solution for the current horizon $\mathcal{S}_N^{\text{red.}}$. This relationship is used in Alg. 2 to construct $\mathcal{S}_{N+1}^{\text{red.}}$ from $\mathcal{S}_N^{\text{red.}}$.

The overall algorithm is stated in Alg. 3. In proceeds analogously to Alg. 4 in Mitze and Mönnigmann (2019), but adds the symmetric active set whenever an active set is added to $\mathcal{M}_N$ (lines 14,18).

Note that active sets such that all symmetric constraints are either both active or both inactive are equal to their symmetric active set. However, these candidate active sets

Algorithm 2: Determination of $\mathcal{S}_{N+1}^{\text{red.}}$ from $\mathcal{S}_{N}^{\text{red.}}$

Input: $\mathcal{S}_{N}^{\text{red.}}$, $\mathcal{S}_{N}^{\text{degen.}}$
Initialization: set $\mathcal{S}_{N+1}^{\text{red.}} = \emptyset$, $\mathcal{S}_{N+1}^{\text{degen.}}$ and $\mathcal{S}_{N+1}^{\text{pruned}} = \emptyset$
for *every* $\mathcal{A}_l \in \mathcal{S}_{N}^{\text{red.}}$ **do**
  if $\mathcal{A}_l \subseteq \{1, ..., Nq_{\mathcal{UX}}\}$ **then**
    add $\mathcal{A}_l$ to $\mathcal{S}_{N+1}^{\text{red.}}$
    if $\mathcal{A}_l \in \mathcal{S}_{N}^{\text{degen.}}$ **then**
      add $\mathcal{A}_l$ to $\mathcal{S}_{N+1}^{\text{degen.}}$
  if $\mathcal{A}_l \not\subseteq \{1, ..., (N-1)q_{\mathcal{UX}}\}$ **then**
    for *every* $\mathcal{A}_i = \mathcal{A}_j \cup (\mathcal{A}_l \oplus \{q_{\mathcal{UX}}\})$ *with* $\mathcal{A}_j \in \mathcal{P}(\mathcal{Q}_0)$ *by increasing cardinality* **do**
      if $\mathcal{A}_i \not\supseteq \tilde{\mathcal{A}}$ *for all* $\tilde{\mathcal{A}} \in \mathcal{S}_{N+1}^{\text{pruned}}$ **then**
        solve (7) for QP for horizon $N+1$
        if *solution exists* **then**
          add $\mathcal{A}_i$ to $\mathcal{S}_{N+1}^{\text{red.}}$
          if *solution* $t = 0$ **then**
            add $\mathcal{A}_i$ to $\mathcal{S}_{N+1}^{\text{degen.}}$
        else
          solve (7) without (7b) and (7c) for QP for horizon $N+1$
          if *no solution exists* **then**
            add $\mathcal{A}_i$ to $\mathcal{S}_{N+1}^{\text{pruned}}$
Output: $\mathcal{S}_{N+1}^{\text{red.}}$, $\mathcal{S}_{N+1}^{\text{degen.}}$

are never part of $\mathcal{M}_N$, because $G_{\mathcal{A}}$ does not have full row rank. The special case $\mathcal{A} = \{\}$ need not be added. It is treated in lines 13 and 17.

Algorithm 3: Dynamic programming approach to solving constrained linear-quadratic OCPs

Input: $\mathcal{S}_{1}^{\text{red.}}$, $\mathcal{S}_{1}^{\text{degen.}}$ (from Alg. 1), $N_{\max} \geq 1$
Initialization: set $\mathcal{M}_N = \emptyset$
for $N = 1$ *to* $N_{\max} - 1$ **do**
  determine $\mathcal{S}_{N+1}^{\text{red.}}$ and $\mathcal{S}_{N+1}^{\text{degen.}}$ with Alg. 2
  if $\mathcal{S}_{N+1}^{\text{red.}} \subseteq \mathcal{P}(\{1, ..., Nq_{\mathcal{UX}}\})$ **then**
    break
$N = N + 1$
for *every* $\mathcal{A}_k \in \mathcal{S}_{N}^{\text{red.}}$ **do**
  if *rowrank*$(G_{\mathcal{A}_k}) = |\mathcal{A}_k|$ **then**
    if $\mathcal{A}_k \in \mathcal{S}_{N}^{\text{degen.}}$ **then**
      if *polytope def. by* $\mathcal{A}_k$ *full-dim.* **then**
        add $\mathcal{A}_k$ to $\mathcal{M}_N$
        if $\mathcal{A}_k \neq \{\}$ **then**
          add $\mathcal{A}'_k$ that is symmetric to $\mathcal{A}_k$ to $\mathcal{M}_N$
    else
      add $\mathcal{A}_k$ to $\mathcal{M}_N$
      if $\mathcal{A}_k \neq \{\}$ **then**
        add $\mathcal{A}'_k$ that is symmetric to $\mathcal{A}_k$ to $\mathcal{M}_N$
Output: $\mathcal{M}_N$

4. EXAMPLE

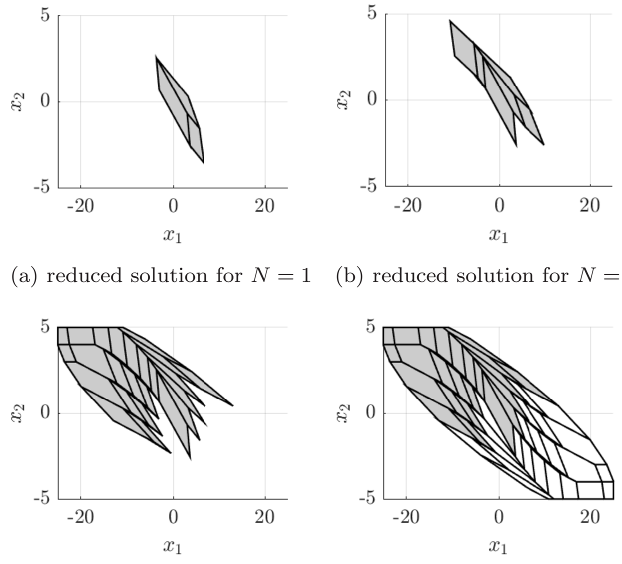
We illustrate how the new approach proceeds and analyze its computational effort with an example. We use the double integrator from Gutman and Cwikel (1987)

$$x(k+1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x(k) + \begin{pmatrix} 0.5 \\ 1 \end{pmatrix} u(k)$$

with symmetric input constraints $|u(k)| \leq 1$, symmetric state constraints $|x_1(k)| \leq 25$, $|x_2(k)| \leq 5$ and cost function matrices $Q = 1 \in \mathbb{R}^{2\times 2}$, $R = 0.1$ as an example. The terminal cost P and set $\mathcal{T}$ are as described in Sect. 2.

We show how the solution develops when the horizon is increased iteratively (Fig. 2) to illustrate how the proposed approach proceeds. We choose $N_{\max} = 5$.

First, we determine the reduced solution for the initial horizon $N = 1$ with Alg. 1. Because the approach considers the reduced power set as candidate active sets, only a subset of the solution, the reduced solution, is determined (Fig. 2a). Then, we apply Alg. 2 to get the reduced solutions for the increased horizons $N = 2, ..., 5$ (Figs. 2b and 2c). Once the target horizon $N = N_{\max}$ has been reached, all symmetric active sets are added with Alg. 3 (Fig. 2d).



(a) reduced solution for $N = 1$ (b) reduced solution for $N = 2$

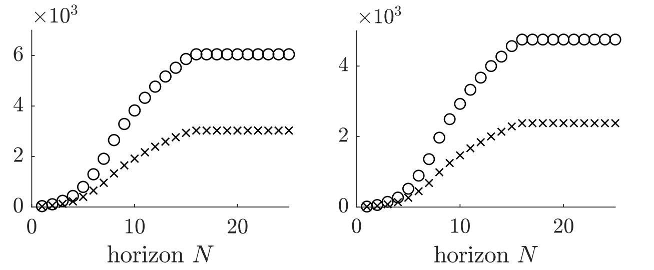
(c) reduced solution for $N = 5$ (d) solution for $N = 5$

Fig. 2. Polytopes defined by active sets that are elements of the reduced solution are shown in gray, polytopes defined by their symmetric active sets are shown in white in Fig. 2d.

4.1 Computational effort

In order to compare the proposed approach to the one from Mitze and Mönnigmann (2019), we compare the numbers of executed optimality tests with (7) and feasibility tests with (7) without (7b) and (7c). The computational effort of solving the LPs is relatively high, therefore these numbers dominate the computational effort of both approaches. Figure 3 shows the numbers for the example stated in Sect. 4 as a function of the horizon N. It is evident that the numbers of optimality and feasibility tests decrease by roughly a factor of 1/2.

The number of candidate active sets when solving an OCP (1) for a horizon $N_{\max}$ equals the number of candidates when determining the solution for the initial horizon



(a) optimality tests (b) feasibility tests

Fig. 3. Number of optimality and feasibility tests for the dynamic programming algorithm presented in Mitze and Mönnigmann (2019) (circles) and for Alg. 3 (crosses) as a function of the horizon N.

(line 2 in Alg. 1) plus the number of candidates when determining the solution for the increased horizons $N = 1, ..., N_{\max} - 1$ (line 9 in Alg. 2). When determining the solution for the initial horizon, the algorithm in Mitze and Mönnigmann (2019) treats all elements in $\mathcal{P}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$ whereas the approach proposed in this paper treats all elements in $\mathcal{P}^{\text{red.}}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$. The number of elements in $\mathcal{P}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$ is $2^{q_{\mathcal{UX}}+q_{\mathcal{T}}}$, the number of elements in $\mathcal{P}^{\text{red.}}(\{1, ..., q_{\mathcal{UX}} + q_{\mathcal{T}}\})$ is $\frac{1}{2}(2^{q_{\mathcal{UX}}+q_{\mathcal{T}}} + 2^{(q_{\mathcal{UX}}+q_{\mathcal{T}})/2})$. Note that the number of elements is not exactly reduced by a factor 1/2, because candidate active sets such that symmetric constraints are both active are equal to their symmetric active sets. When determining the solution for an increased horizon, the algorithm in Mitze and Mönnigmann (2019) treats all elements in the superset (5b) whereas the approach proposed in this paper treats all elements in the superset (12b). The number of elements in the superset (5b) is $2^{q_{\mathcal{UX}}} \cdot |\tilde{\mathcal{S}}_N|$, the number of elements in the superset (12b) is $2^{q_{\mathcal{UX}}} \cdot |\tilde{\mathcal{S}}_N^{\text{red.}}|$. By definition, $\tilde{\mathcal{S}}_N$ does not contain $\mathcal{A} = \{\}$. Active sets with symmetric constraints that are both active are infeasible and therefore not part of $\tilde{\mathcal{S}}_N$. Because $\tilde{\mathcal{S}}_N$ only consists of active sets that have a symmetric active set, $|\tilde{\mathcal{S}}_N^{\text{red.}}| = \frac{1}{2}|\tilde{\mathcal{S}}_N|$ and hence, the number of candidates for the increased horizon is exactly reduced by a factor of 1/2. Therefore, the number of candidate active sets for the approach presented in this paper is reduced by a factor of 1/2 up to $\frac{1}{2}2^{(q_{\mathcal{UX}}+q_{\mathcal{T}})/2}$. This is illustrated in Fig. 4.

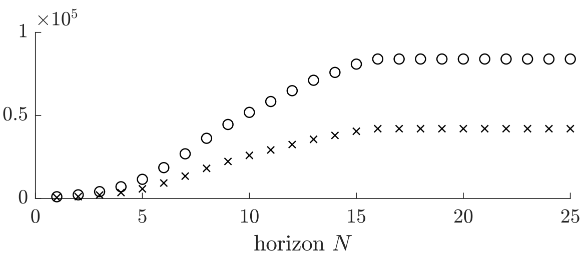


Fig. 4. Number of candidate active sets for the dynamic programming algorithm presented in Mitze and Mönnigmann (2019) (circles) and for Alg. 3 (crosses) over the horizon.

5. CONCLUSION

We improved a recently proposed dynamic programming algorithm solving constrained linear-quadratic OCP with input and state constraints that are point-symmetric with respect to the origin. We showed that the computational effort is almost reduced by a factor of 1/2.

ACKNOWLEDGEMENTS

Support by the German Federal Ministry for Economic Affairs and Energy under grant 0324125C is gratefully acknowledged.

REFERENCES

Ahmadi-Moshkenani, P., Johansen, T.A., and Olaru, S. (2018). Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions. *IEEE Transactions on Automatic Control*, 63(10), 3221–3231.

Baotić, M. (2002). An efficient algorithm for multi-parametric quadratic programming. Technical report, ETH Zürich. AUT02-05.

Bemporad, A. (2004). Hybrid Toolbox - User's Guide. `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox`.

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

Danielson, C.R. (2014). *Symmetric Constrained Optimal Control: Theory, Algorithms, and Applications.* Ph.D. thesis, University of California, Berkeley.

Feller, C., Johanson, T.A., and Olaru, S. (2013). An improved algorithm for combinatorial multi-parametric quadratic programming. *Automatica*, 49, 1370–1376.

Gupta, A., Bhartiya, S., and Nataraj, P. (2011). A novel approach to multiparametric quadratic programming. *Automatica*, 47, 2112–2117.

Gutman, P.O. and Cwikel, M. (1987). An Algorithm to Find Maximal State Constraint Sets for Discrete-Time Linear Dynamical Systems with Bounded Controls and States. *IEEE Transactions on Automatic Control*, 32, 251–254.

Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *European Control Conference*, 502–510.

Herceg, M., Jones, C.N., Kvasnica, M., and Morari, M. (2015). Enumeration-based approach to solving parametric linear complementarity problems. *Automatica*, 62, 243–248.

Jost, M., Schulze Darup, M., and Mönnigmann, M. (2015). Optimal and suboptimal event-triggering in linear model predictive control. In *European Control Conference*, 1147–1152.

Mare, J. and De Dona, J. (2007). Solution of the input-constrained LQR problem using dynamic programming. *Systems and Control Letters*, 56, 342–348.

Mitze, R. and Mönnigmann, M. (2019). A dynamic programming approach to solving constrained linear-quadratic optimal control problems. Submitted to Automatica, `https://arxiv.org/abs/1910.11231`.

Mönnigmann, M. (2019). On the structure of the set of active sets in constrained linear quadratic regulation. *Automatica*, 106, 61–69.

Muñoz de la Peña, D., Alamo, T., Bemporad, A., and Camacho, E.F. (2004). A Dynamic Programming Approach for Determining the Explicit Solution of Linear MPC Controllers. In *43rd IEEE Conference on Decision and Control*, 2479–2484.

Oberdieck, R., Diangelakis, N.A., and Pistikopoulos, E.N. (2017). Explicit model predictive control: A connected-graph approach. *Automatica*, 76, 103–112.

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

Sznaier, M. and Damborg, M. (1987). Suboptimal control of linear systems with state and control inequality constraints. In *Proc. 26th Conf. on Decision and Control*, 761–762.

Tøndel, P., Johanson, T.A., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39, 489–497.