2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Explicit MPC of higher-order linear processes via combinatorial multi-parametric quadratic programming

Christian Feller[1], Tor Arne Johansen[2]

Abstract— **Parameter-dependent constrained optimization problems like they occur in the context of model predictive control (MPC) can be solved explicitly by means of multi-parametric quadratic programming (mpQP) techniques. We present a complexity analysis for a recently proposed combinatorial mpQP algorithm and discuss its advantages over existing geometric approaches concerning off-line explicit MPC computations for higher-order linear systems. The results are accompanied by numerical benchmark results for two suitable example problems from the area of process control.**

I. Introduction

For a given system state, MPC computes the control action in a receding horizon fashion by solving on-line repeatedly a finite horizon open-loop optimal control problem in which both system dynamics and constraints are taken into account. Seeing the evolving system state as a parameter and using multi-parametric programming (mpP) techniques, the on-line optimization involved in the MPC approach can be moved off-line by computing an explicit optimal control law a priori. In this paper, we will focus on strictly convex multi-parametric quadratic programming (mpQP) problems, which are related to linear MPC based on a quadratic cost function, i.e., the constrained finite-horizon LQR problem. In general, the solution to this problem has the form of a piecewise affine function that is defined over a polyhedral partition of the parameter space into so-called critical regions, where each region corresponds to a set of optimal active constraints.

Many 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, e.g., [1], [2] and [3], [4]. Due to the involved geometric operations in the parameter space, these algorithms scale, in general, rather badly with the complexity of the resulting polyhedral partition, i.e., the number of full-dimensional critical regions and the dimension of the parameter (or state) space. Thus, MPC problems that can be solved explicitly by means of geometric mpQP algorithms are usually limited to lower-dimensional system models. In [5] and [6], it is shown that a significant complexity reduction can be achieved by using model reduction techniques. However, in this case, the obtained explicit solution is only approximate, and neither good performance nor stability of the closed loop can be guaranteed in general.

[1] Institute for Systems Theory and Automatic Control, University of Stuttgart, Pfaffenwaldring 9, 70550 Stuttgart, christian.feller@ist.uni-stuttgart.de

[2] Department of Engineering Cybernetics, NTNU, O.S. Bragstads plass, 7491 Trondheim, Norway, tor.arne.johansen@itk.ntnu.no

Recently, a new combinatorial mpQP approach has been proposed in [7] that is based on an implicit enumeration of all possible constraint combinations in form of candidate active sets. In this approach, the candidate active sets are tested for optimality by solving a linear program (LP), while several pruning criteria are used to reduce the number of constraint combinations that need to be considered in the enumeration process. In contrast to most of the existing geometric approaches, the combinatorial approach does not rely on an explicit geometric exploration strategy and guarantees partitioning of the complete parameter space. Despite the pruning criteria, one disadvantage of the approach is the combinatorial complexity with respect to the number of possible candidate active sets, i.e., the number of LPs that need to be solved in the solution process. However, since it requires no geometric operations in the parameter space, it might be expected that the combinatorial approach scales rather nicely with the number of full-dimensional critical regions and the order of the involved system model.

In this paper, we give a qualitative complexity analysis for the combinatorial mpQP algorithm from [7] and discuss its advantages over existing geometric algorithms when considering mpQP problems with higher dimensional parameter spaces. In addition, we present numerical case studies for two problems from the area of process control: an 8-state-1-input fuel cell breathing control system and a binary distillation column with two inputs and up to 82 system states. In order to demonstrate the benefits of the combinatorial approach, its performance with respect to the off-line computation of the explicit MPC law is compared to the geometric mpQP algorithm that is implemented in the Multi-Parametric Toolbox ($\mathbb{MPT}$) [8]. Note that some techniques to achieve further reduction in the complexity of the discussed combinatorial mpQP approach [7] are proposed in [9] and [10]. However, these are not considered in the following for the sake of simplicity.

II. Multi-parametric quadratic programming

Consider a linear time-invariant system of the form

$$x(k+1) = Ax(k) + Bu(k) \tag{1a}$$

$$y(k) = Cx(k)\ , \tag{1b}$$

with the state, input, and output vectors $x(k)$, $u(k)$, $y(k)$, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, (A, B) stabilizable. The goal is to regulate the system state $x(k)$ to the origin while satifsying output and input constraints of the form $y_{\min} \leq y(k) \leq y_{\max}$, $u_{\min} \leq u(k) \leq u_{\max}$ for all $k \geq 0$. In MPC,

this goal is usually achieved by solving repeatedly an open-loop optimal control problem with quadratic cost over a finite prediction horizon N. The resulting constrained optimization problem can be represented as an mpQP of the form [1]

$$V_z^*(x) = \min_z \frac{1}{2} z^T H z \tag{2a}$$

$$\text{s. t. } Gz \leq W + Sx \ , \tag{2b}$$

where $z = U + H^{-1}F^T x \in \mathbb{R}^{Nm}$ denotes the vector of optimization variables and $H \succ 0 \in \mathbb{R}^{Nm \times Nm}$, $G \in \mathbb{R}^{q \times Nm}$, $W \in \mathbb{R}^{q \times 1}$, $S \in \mathbb{R}^{q \times n}$, and $F \in \mathbb{R}^{n \times Nm}$ are real matrices. Here, $U = [u_{0|k}^T, u_{1|k}^T, \ldots, u_{N-1|k}^T]^T$ is the stacked vector of optimal inputs that minimize the corresponding quadratic cost of the receding horizon problem at sampling instant k.

A. Analytic solutions to mpQP problems

As shown in [1], we can solve (2) by applying the corresponding Karush-Kuhn-Tucker (KKT) conditions, which are given by

$$Hz + G^T \lambda = 0, \quad \lambda \in \mathbb{R}^q, \tag{3a}$$

$$\lambda^i \left(G^i z - W^i - S^i x \right) = 0, \quad i = 1, \ldots, q, \tag{3b}$$

$$\lambda \geq 0, \tag{3c}$$

$$Gz \leq W + Sx, \tag{3d}$$

where the superscript index i denotes the i^{th} row of a matrix or vector and λ refers to the vector of Lagrangian multipliers.

Assuming that we know the optimal active set $\mathcal{A} \subseteq Q = \{1, 2, \ldots, q\}$, $\mathcal{A} := \{i \in \mathcal{Q} \mid G^i z^*(\bar{x}) - W^i - S^i \bar{x} = 0\}$ for a given $\bar{x}$ and corresponding optimizer $z^*(\bar{x})$, we can form matrices the $G^{\mathcal{A}}$, $W^{\mathcal{A}}$, and $S^{\mathcal{A}}$, containing the rows of G, W, and S associated to the indices in $\mathcal{A}$. In [1], it is shown that if $G^{\mathcal{A}}$ has full row rank, the KKT conditions (3) can be used to derive the parameter-dependent optimizer

$$z_{\mathcal{A}}(x) = H^{-1}(G^{\mathcal{A}})^T H_{G^{\mathcal{A}}}^{-1} \left(W^{\mathcal{A}} + S^{\mathcal{A}} x \right) \tag{4}$$

which solves the mpQP problem for all parameters x in a so-called *critical region* $CR_{\mathcal{A}} \subset \mathbb{R}^n$ around $\bar{x}$, where $CR_{\mathcal{A}}$ is defined by the inequalities

$$H_{G^{\mathcal{A}}}^{-1} \left(W^{\mathcal{A}} + S^{\mathcal{A}} x \right) \leq 0, \tag{5a}$$

$$GH^{-1}(G^{\mathcal{A}})^T H_{G^{\mathcal{A}}}^{-1} \left(W^{\mathcal{A}} + S^{\mathcal{A}} x \right) \leq W + Sx \tag{5b}$$

with $H_{G^{\mathcal{A}}}^{-1} := \left(G^{\mathcal{A}} H^{-1} (G^{\mathcal{A}})^T \right)^{-1}$. This polyhedral region 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 $\mathcal{A}_i$, $i = 1, \ldots, n_r$, the parameter space is implicitly partitioned into n_r critical regions $CR_{\mathcal{A}_i}$, and the optimizer can be represented as a continuous piecewise affine function of the parameter x [1]. Note that $H_{G^{\mathcal{A}}}^{-1}$ may not exist if the rows of the constraint matrix $G^{\mathcal{A}}$ are linearly dependent, in which case further methods have to be applied in order to obtain a representation of $z_{\mathcal{A}}$ and $CR_{\mathcal{A}}$ [1], [9].

B. Geometric mpQP algorithms

In [1], the authors propose 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. A different approach for the exploration of the parameter space is used in the algorithm presented by [2], which is summarized in the following pseudo-code.

Algorithm 1 Geometric mpQP algorithm from [2]

1. solve an LP to find an x_0 for which the QP (2) is feasible;
2. identify the current optimal active set $\mathcal{A}_0$ by solving the QP (2) with $x = x_0$; if feasible, construct the optimizer $z_{\mathcal{A}_0}$ and the critical region $CR_{\mathcal{A}_0}$ by using (4) and (5) ;
3. for every facet $\mathscr{F}_i$ of $CR_{\mathcal{A}_0}$, choose the center $\hat{x}_i \in \mathscr{F}_i$ and step over the boundary of $CR^{\mathcal{A}_0}$ by constructing a new point $x_i = \hat{x}_i + \varepsilon \left(A_0^i \right)^T$, where $\varepsilon > 0$ is a scalar and $A_0^i x \leq b_0^i$ is the constraint defining the hyperplane that contains $\mathscr{F}_i$;
4. repeat step 2. and step 3. with $x = x_i$ until no new critical regions are found, i.e., $\nexists$ new feasible $x_j \notin \cup_i CR_{\mathcal{A}_i}$;

Algorithm 1 is essentially implemented in the mpQP solver of the Multi-Parametric Toolbox ($\mathbb{MPT}$) [8], which we will use in Section III as a reference for the geometric mpQP algorithms. Different geometric algorithms are presented in [3], [4] and (with some combinatorial elements) in [11] and [12]. However, all these algorithms rely on geometric operations like facet-crossing or polytope projections when constructing the explicit mpQP solution.

C. 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 in [7] operates directly on the level of possible optimal active sets. Particularly, the main idea of the approach is the implicit enumeration of all possible combinations of active constraints.

Consider again the set $\mathcal{Q} = \{1, \ldots, q\}$ referring to the constraint indices in (2b). Then, the active set $\mathcal{A}(z, x)$ can be described as $\mathcal{A}(z, x) := \{i \in \mathcal{Q} \mid G^i z - W^i - S^i x = 0\}$, while the corresponding set of inactive constraints $\mathcal{J}(z, x)$ is given by $\mathcal{J}(z, x) := \mathcal{Q} \setminus \mathcal{A}(z, x)$. It can be shown that all possible optimal active sets are included in the set

$$\mathcal{P}'(\mathcal{Q}) := \Big\{ \mathcal{A}_1 = \{\ \}, \mathcal{A}_2 = \{1\}, \ldots, \mathcal{A}_{q+1} = \{q\}, \ldots$$

$$\mathcal{A}_{q+2} = \{1, 2\}, \ldots, \mathcal{A}_{n_{\mathcal{A}}} = \{\tilde{q} - Nm + 1, \ldots, q\} \Big\}, \tag{6}$$

which is a subset of the power set $\mathcal{P}(\mathcal{Q})$ and consists of $n_{\mathcal{A}} = \sum_{l=0}^{\tilde{m}} \binom{q}{l} \leq 2^q$ index sets. Here, $\tilde{q}$ and $\tilde{m}$ are defined as $\tilde{q} = \max\{Nm, q\}$ and $\tilde{m} = \min\{Nm, q\}$, respectively [7], [10]. In order to identify all optimal active sets, the authors of [7] suggest to choose candidate active

sets $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$ in the order of increasing cardinality and use the LP

$$\max_{z,x,\lambda^{\mathcal{A}_i},s^{\mathcal{J}_i}} \; t \tag{7a}$$

$$\text{s. t.} \;\; te_1 \leq \lambda^{\mathcal{A}_i},\; te_2 \leq s^{\mathcal{J}_i} \tag{7b}$$

$$t \geq 0,\; \lambda^{\mathcal{A}_i} \geq 0,\; s^{\mathcal{J}_i} \geq 0 \tag{7c}$$

$$Hz + (G^{\mathcal{A}_i})^T \lambda^{\mathcal{A}_i} = 0 \tag{7d}$$

$$G^{\mathcal{A}_i} z - S^{\mathcal{A}_i} x - W^{\mathcal{A}_i} = 0 \tag{7e}$$

$$G^{\mathcal{J}_i} z - S^{\mathcal{J}_i} x - W^{\mathcal{J}_i} + s^{\mathcal{J}_i} = 0 \tag{7f}$$

to check whether $\mathcal{A}_i$ is indeed an 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 (7) has a feasible solution, there exist feasible $z_{\mathcal{A}_i}, x_{\mathcal{A}_i}, \lambda^{\mathcal{A}_i}, s^{\mathcal{J}_i}$ satisfying the KKT conditions (3), and $\mathcal{A}_i$ is an optimal active set [7]. In this case, (4) and (5) can be used to compute the affine solution and the corresponding critical region. On the other hand, infeasibility of the LP (7) 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 (7) 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 [7]. The following criterion reduces the number of candidate sets and renders the enumeration of $\mathcal{P}'(\mathcal{Q})$ implicit.

Criterion 1 (Pruning of candidate active sets [7])
If a candidate active set $\mathcal{A}_i \in \mathcal{P}'(\mathcal{Q})$
***(i)** leads to a row rank deficient matrix $G^{\mathcal{A}_i}$, 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 $\mathcal{P}'(\mathcal{Q})$.

While the first pruning condition results from the hierarchical structure of the combinatorial enumeration process, the second one follows directly from the fact that an infeasible system of equality and inequality constraints, i.e., here the LP (7), will stay infeasible when some of the inequalities are treated as equalities. With this, the combinatorial mpQP algorithm proposed in [7] can be summarized as follows.

Algorithm 2 Combinatorial mpQP algorithm from [7]

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 (7)
  ⌞ if feasible, use (4) and (5) to construct $z_{\mathcal{A}_i}$ and $CR_{\mathcal{A}_i}$
  ⌞ if infeasible, solve (7) 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.

A graphical illustration of the combinatorial enumeration strategy and the involved pruning process is given in Fig. 1 in form of a tree diagram. More details on the properties of the combinatorial approach and aspects concerning its implementation can be found in [7] and [9].

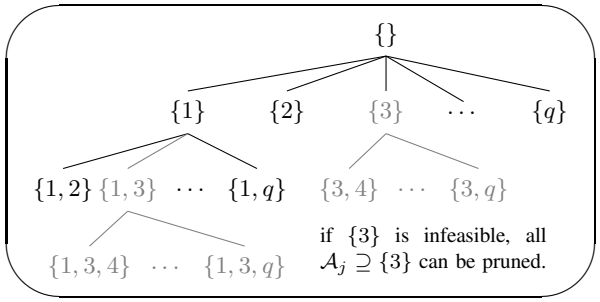

Fig. 1. Combinatorial enumeration strategy used in [7].

D. Complexity analysis

In the following, we present a qualitative complexity analysis for the discussed combinatorial mpQP algorithm Algorithm 2 and compare it with the complexity of the geometric exploration used in Algorithm 1. The goal is to point out structural differences between the two appoaches and to provide a theoretical basis for the numerical case studies that will be discussed in Section III.

The main operations when executing Algorithm 2 are

(i) solving the LP (7) (with/without optimality constraints),
(ii) constructing candidate sets & pruning infeasible constraint combinations,
(iii) removing redundant constraints from polyhedra and checking critical regions for full-dimensionality .

It is obvious that the effort for the first two tasks is closely related to the number of maximal possible constraint combinations, i.e., the number of elements in $\mathcal{P}'(\mathcal{Q})$. Since this number is equal to the maximal number of optimization problems that need to be solved, we will denote it by $n_{LP,\max}$. As stated above, $n_{LP,\max}$ is given by $n_{LP,\max} = n_{\mathcal{A}}$. By introducing a *pruning rate* γ, we can express the number of optimization problems that are effectively solved in step (i) of the exploration process as $n_{LP,(i)} = (1-\gamma)\; n_{LP,\max}$, $\gamma < 1$. Note that the pruning rate γ is problem dependent and therefore, in general, not known a priori. The computational burden that is caused in step (ii) by constructing the candidate active sets and pruning infeasible computation can not be expressed directly in terms of optimization problems. However, since it will also depend on the number of maximal possible combinations, it could be included in our estimate by introducing an additional factor $\kappa > 0$, i.e., $n_{LP,(ii)} \approx \kappa\, n_{LP,\max}$. Thus, we get the following cumulative estimate for the tasks (i) and (ii):

$$n_{LP,(i+ii)} = (1-\gamma+\kappa)\; n_{LP,\max} = \tilde{\gamma} \sum_{l=0}^{\tilde{m}} \binom{q}{l} \tag{8}$$

with $\tilde{\gamma} \in \mathbb{Q}^+$ and, usually, $\tilde{\gamma} < 1$ since $\kappa \ll \gamma$. Now let us turn to the last part (iii). Redundant constraints can be removed from the polyhedral description of each critical region by solving one linear optimization problem for each defining hyperplane. Moreover, whether a critical region is full-dimensional can be tested by checking if the Chebyshev radius of the corresponding polytope is positive (respectively

above a certain positive theshold). Since the Chebyshev radius of a polytope can be computed by solving one linear optimization problem, the computational effort for task (iii) can be estimated as

$$n_{LP,(iii)} = n_r \times \begin{pmatrix} \# \text{ LPs for redundancy check per region} \\ +1 \text{ LP for checking full-dimensionality} \end{pmatrix}, \tag{9}$$

where n_r is the total number of identified critical regions in the parameter space partition. By combining Equations (8) and (9), we can finally give an estimate for the computational effort of the combinatorial approach based on the number of LPs that need to be solved:

$$\tilde{\gamma} \sum_{l=0}^{\tilde{m}} \binom{q}{l} + n_r \times \begin{pmatrix} \# \text{ LPs for redundancy check per region} \\ + \text{ 1 LP for checking full-dimensionality} \end{pmatrix}. \tag{10}$$

On the other hand, the main cost of the geometric approach that is used in Algorithm 1 and also in the mpQP solver of the $\mathbb{MPT}$ can be estimated by

$$\begin{aligned} & n_r \times \begin{pmatrix} \# \text{ LPs for stepping over the facets per region} \\ + \text{ 1 QP for identifying the optimal active set} \end{pmatrix} \\ & + n_r \times \begin{pmatrix} \#\text{LPs for redundancy check per region} \\ +1 \text{ LP for checking full-dimensionality} \end{pmatrix}, \end{aligned} \tag{11}$$

where we used the fact that one LP has to be solved for each facet of a critical region in order to compute the corresponding center $\hat{x}_i \in \mathscr{F}_i$. A similar cost estimate for the geometric algorithms from [1] and [3] is given in [3].

By comparing Equation (10) and Equation (11), we see that the complexities of the combinatorial and the geometric algorithm differ only in the first term. This is plausible insofar that in both cases the first term describes the cost of the operations which are characteristic for the particular algorithm. We see that while the computational burden for the combinatorial approach is mainly affected by the maximal number of constraint combinations $n_{LP,\max}$ and the overall pruning rate $\tilde{\gamma}$, the complexity of the geometric algorithm is closely related to the number of regions n_r. Moreover, since both the number of critical regions, the number of facets per region, and the size of the additional LPs and QPs will generally grow exponentially, the additional geometric operations will become more and more time consuming with increasing dimension of the parameter space. In fact, our observations showed that computing the centers $\hat{x}_i \in \mathscr{F}_i$ when stepping over the increasing number of facets is the most time consuming operation. Of course, this may be influenced by the numerical implementation, but we consider the code of the $\mathbb{MPT}$ to be rather mature and efficient. In contrast to this, the combinatorial approach scales rather well with the dimension of the parameter space, but rather badly with the numbers of optimization variables and constraints in the mpQP problem. Hence, for problems where the parameter space partition consists of a large number of regions in a higher-dimensional parameter space, it may be advantageous to use the combinatorial approach.

III. Numerical Case Studies

In this section we want to illustrate the results of the above complexity analysis on two numerical examples from the area of process control. Both control problem formulations are based on linear system models with a relatively large number of system states and a rather small number of constrained in- and outputs. Hence, the corresponding mpQP problems involve a higher-dimensional parameter vector, while the number of possible candidate active sets is relatively small. Only the basic principles of the underlying industrial processes and their system models are outlined in the following; for more details the reader is referred to the respective references in the literature. All computations are performed on a 3 GHz Dual Core PC with 8 GB RAM, running MATLAB 7.11, $\mathbb{MPT}$ version 2.6.3 and NAG Toolbox Mark 22.1 (64 bit).

A. Fuel Cell Breathing Control

As the first example, we consider a fuel cell breathing control system that has already been discussed in the context of explicit linear MPC in [5]. A description of the considered fuel cell process and a derivation of the associated nonlinear and linear system models is given in [13]. In short, the goal is to control the air flow inside a fuel cell stack system in order to avoid oxygen starvation, a phenomenon which can result in a rapid cell voltage decrease and the destruction of the fuel cell system membranes. In the following, we consider a linear system model of the form

$$\dot{x} = A_c x + B_c u, \; y = C_c x \tag{12}$$

for which the system matrices $A_c \in \mathbb{R}^{8\times 8}$, $B_c \in \mathbb{R}^{8\times 1}$, $C_c \in \mathbb{R}^{2\times 8}$ can be found in [13]. The model involves eight system states, which refer to the oxygen, hydrogen and nitrogen masses, the pressures in the return and supply manifold, the total mass in the supply manifold, the mass of water on the fuel cell anode, and the rortational speed of the compressor. The only input to the system is the compressor motor voltage, while the outputs are given by the net power and the oxygen excess ratio. As in [5], we discretize the linear model (12) with a sampling time of $T_d = 1$ ms and formulate the open-loop optimal control problem of linear MPC for varying prediction horizon N, using the input and output constraints $|u| \leq 5$, $|y_1| \leq 0.03$, $|y_2| \leq 0.2$ together with a quadratic performance index based on $Q = 10^3 \cdot {C_d}^T C_d$, $R = 1$, where C_d denotes the output matrix of the disretized system model.

In Table I, we give a comparison of the computation times that were needed by the combinatorial and the geometric mpQP algorithm for computing the explicit MPC solution. As we can see, the combinatorial mpQP algorithm is indeed more efficient than the $\mathbb{MPT}$ algorithm, at least for a prediction horizon of $N \leq 6$. However, the advantages in the computational speed decline for increasing horizon, and for a horizon of $N = 7$ the computation time of the combinatorial algorithm increases dramatically. In fact, due to the exponentially increasing number of possible candidate

TABLE I

PERFORMANCE RESULTS FOR THE 8TH-ORDER FUEL CELL PROCESS.

N	1	2	3	4	5	6	7
n_r	5	27	123	449	1 261	2 335	3 996
$t_{\text{Alg.1}}[s]$	0.18	1.04	7.54	54.31	251.93	942.37	3 088.38
$t_{\text{Alg.2}}[s]$	0.05	0.21	1.23	6.63	52.67	553.36	-

active sets ($n_{LP,\max} > 33 \times 10^6$), the first term of the complexity estimate in (10) explodes, and the algorithm is not able to compute the explicit solution in a computation time of four hours. Nevertheless, the results for $N \leq 6$ confirm in principle that the combinatorial mpQP approach may have significant advantages over existing geometric approaches for the discussed class of mpQP problems.

B. Binary distillation column

As another example for explicit linear MPC based on large-scale system models we want to consider the well-known "Column A" [14], which is a well-studied example for a binary distillation column model. All data and MATLAB models used in the following are taken from the "Column A" website [15], to which the interested reader is referred for more details on the column model. The model includes 41 column stages and the following model assumptions are made: binary mixture, constant pressure, constant relative volatility, constant molar flows, no vapor holdup, linear liquid dynamics, equilibrium on all stages, total condenser. The system states are given by the liquid composition x_i and the liquid hold-up m_i at each stage. In this work, we consider the so called LV-configuration of the system, in which the top and bottom product flow rates are controlled by the reboiler and condenser hold-up via two internal P-controllers. As a result, the model of the distillation column in LV-configuration consists of 82 states, two inputs (reflux and boil-up flow) and two outputs (top and bottom composition x_1, x_{41}), see Fig. 2. Furthermore, model reduction techniques can be employed to reduce the order of the linear model, i.e., the number of considered system states. In [14], it is shown that a second-order model is actually sufficient to represent the fundamental I/O-behavior of the distillation column. In the following, we want to exploit this scalability of the system model size by formulating mpQP problems for a varying dimension of the state/parameter space. Following [16], we discretize the continuous-time model from [15] with $T_s = 4$ min and use the input and output constraints $|u_i| \leq 1$, $|y_i| \leq 0.005$, $i = 1, 2$ together with a quadratic cost function based on $Q = 100 \cdot I_{n_{\text{red}}}$, $R = 0.2 \cdot I_2$, in the formulation of the linear MPC open-loop optimal control problem, where n_{red} denotes the number of state variables in the reduced order system model.

We tested both the combinatorial algorithm and the geometric algorithm of the MPT for the distillation column example while varying both the prediction horizon N and the number of state variables in the underlying system model. The model order was varied from simple reduced order models with $n_{\text{red}} = 4$ to the original full-order model with $n_{\text{red}} = 82$. For each model order, the open-loop optimal control problems of linear MPC were formulated for the prediction horizons $N = 1, 2, 3$. The resulting computation times are presented in Fig. 3.

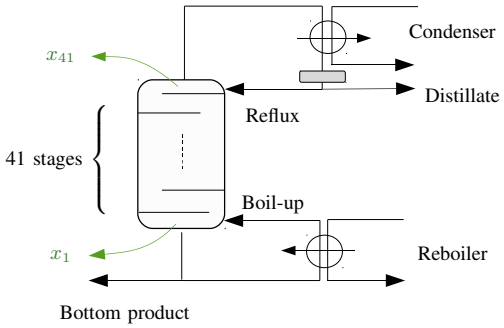

Fig. 2. Schematic distillation column process.

For a horizon of $N = 1$, both algorithms succeed to compute the solution within the imposed time limit of four hours for all considered model orders. As can be seen from the different time curves, the combinatorial algorithm is always faster in computing the explicit solution than the geometric MPT algorithm. In fact, the speed advantage increases from a factor of 3-4 to a factor of around 84 with increasing model order n_{red}. Similar results were obtained for a horizon of $N = 2$. However, two important differences can be observed in Fig. 3. On the one hand, the MPT algorithm is not able to compute the solution for $n_{\text{red}} \geq 54$ within the time limit of four hours. In addition, the algorithm fails completely for $n_{\text{red}} = 12$ and $n_{\text{red}} = 16$. For this cases, only the critical regions corresponding to $\mathcal{A}_0 = \{\}$ are detected, which is caused by numerical problems in the geometric state space exploration process. Again, the results also show that the speed advantage of the combinatorial algorithm tends to grow for increasing model order: while the combinatorial algorithm is two times faster than the geometric algorithm for $n_{\text{red}} = 4$, it is more than 180 times faster for $n_{\text{red}} = 46$. Finally, the bottom plot of Fig. 3 presents the computation times for prediction horizon $N = 3$. Here, the computation time of the geometric MPT algorithm exceeds the time limit of four hours already for $n_{\text{red}} \geq 14$. The computation times for the combinatorial algorithm are also increased significantly, and for $n_{x,\text{red}} = 4$ it is even slower than the geometric algorithm. However, for increasing model order, the combinatorial algorithm is much faster than the geometric one and succeeds in computing all explicit solutions.

To the knowledge of the authors, there exists no example of explicit MPC for such a high-dimensional system in the literature. By taking a look at the corresponding state space partitions, it is easy to comprehend why a geometric exploration strategy may be problematic in the context of the discussed problem setup: the resulting state space partition for the full-order model and horizon $N = 3$ consists of more than 15 000 regions in a 82-dimensional parameter space. Here, the combinatorial approach is much more efficient since it avoids time-demanding and numerically

challenging geometric operations in the high-dimensional parameter space. Furthermore, in all performed computations the combinatorial algorithm produced very reliable results and detected in many cases more correct critical regions than its geometric counterpart. More details concerning the numerical results can be found in [9]. For such complex state space partitions as discussed above, identification of the critical region that contains the current system state x is of course not a trivial task. However, there exist efficient techniques to address this so-called point-location problem, e.g., by making use of multiway trees [17], hash tables [18], or truncated binary trees [19].

IV. Conclusion

In this paper, we have analyzed and demonstrated the applicability of combinatorial multi-parametric quadratic programming to the off-line computation of explicit MPC laws for problem setups involving higher-dimensional system models. The advantages of the combinatorial active set enumeration over geometric exploration strategies have been discussed both by a qualitative complexity analysis and by numerical case studies for suitable example systems. For moderate prediction horizons and simple input and output constraints, the combinatorial algorithm even allows to compute the explicit MPC law for a distillation column process with 82 system states.

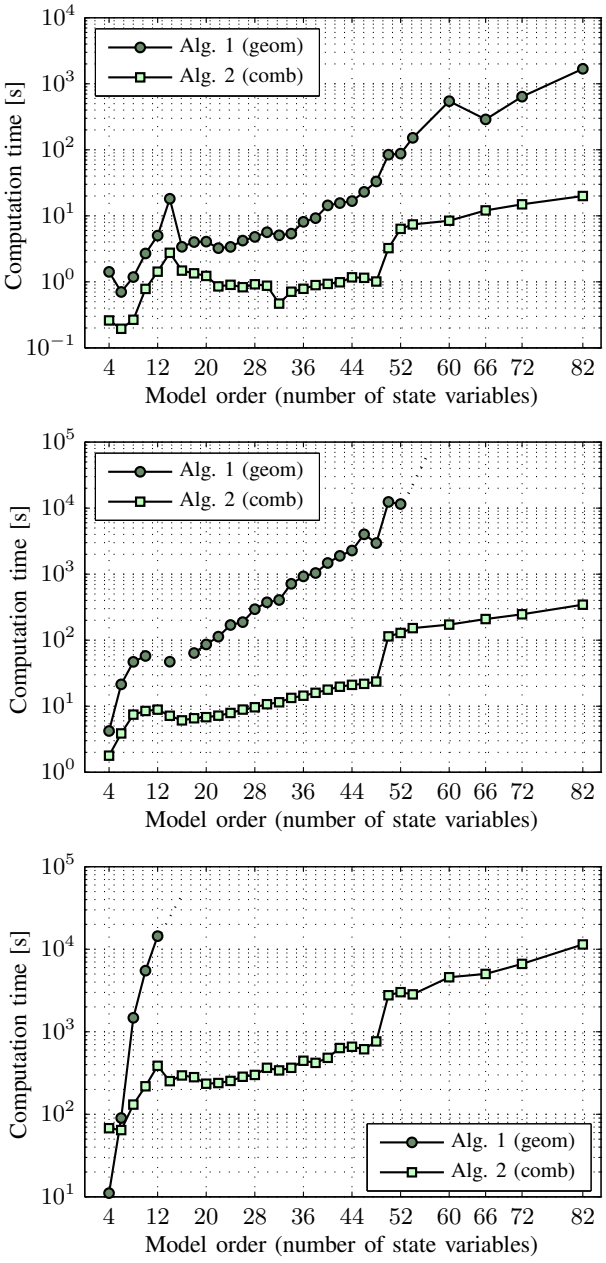

Fig. 3. MpQP computation times for the distillation column example with varying model order for $N = 1$ (top), $N = 2$ (middle), $N = 3$ (bottom). The jump in the computation times at $n_{\mathrm{red}} = 50$ is caused by an abrupt change in the number of critical regions.

References

[1] 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.

[2] M. Baotić, "An Efficient Algorithm for Multi-Parametric Quadratic Programming," ETH Zürich, Tech. Rep. AUT02-05, 2002.

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

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

[5] S. Hovland and J. Gravdahl, "Complexity Reduction in Explicit MPC through Model Reduction," in *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 2008.

[6] T. A. Johansen, "Reduced explicit constrained linear quadratic regulators," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 823–829, 2003.

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

[8] M. Kvasnica, P. Grieder, M. Baotić, and F. J. Christophersen, *Multi-Parametric Toolbox (MPT)*, ETH Zürich, 2006.

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

[10] C. Feller, T. A. Johansen, and S. Olaru, "An improved algorithm for combinatorial multi-parametric quadratic programming," *Automatica*, accepted for publication December 2012.

[11] 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, no. 2, pp. 271–286, 2003.

[12] 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*, vol. 2, 2004, pp. 1580–1585.

[13] J. T. Pukrushpan, "Modeling and Control of Fuel Cell Systems and Fuel Processors," Ph.D. dissertation, University of Michigan, 2003.

[14] S. Skogestad and M. Morari, "Understanding the Dynamic Behavior of Distillation Columns," *Industrial & Engineering Chemistry Research*, vol. 27, no. 10, pp. 1848–1862, 1988.

[15] S. Skogestad, "MATLAB Distillation column model ("Column A")," Website: www.nt.ntnu.no/users/skoge/book/matlab_m/cola/cola.html.

[16] N. L. Ricker, "Model Predictive Control with State Estimation," *Industrial & Eng. Chemistry Research*, vol. 29, pp. 374–382, 1990.

[17] M. Mönnigmann and M. Kastsian, "Fast Explicit Model Predictive Control with Multiway Trees," in *Proceedings of the 18th IFAC World Congress*, Milan, 2011.

[18] F. Bayat, T. A. Johansen, and A. A. Jalali, "Using hash tables to manage time-storage complexity in a point location problem: Application to explicit model predictive control," *Automatica*, vol. 47, pp. 571–577, 2011.

[19] ——, "Flexible piecewise function evaluation methods based on truncated binary search trees and lattice representation in explicit MPC," *IEEE Trans. Control Systems Technology*, vol. 20, pp. 632–640, 2012.