# On degeneracy in exploration of combinatorial tree in multi-parametric quadratic programming

Parisa Ahmadi-Moshkenani, Tor Johansen, Sorin Olaru

# On Degeneracy in Exploration of Combinatorial Tree in Multi-Parametric Quadratic Programming

Parisa Ahmadi-Moshkenani[1], Tor Arne Johansen [1] and Sorin Olaru [2]

***Abstract*— The recently proposed combinatorial approach for multi-parametric quadratic programming (mpQP) has shown to be more efficient than geometric approaches in finding the complete solution when dealing with systems with high dimension of the parameter vector. This method, however, tends to become very slow as the number of constraints increases. Recently, a modification of the combinatorial method is proposed which exploits some of the underlying geometric properties of adjacent critical regions to exclude a noticeable number of feasible but not optimal candidate active sets from the combinatorial tree. This method is followed by a post-processing algorithm based on the geometric operations to assure that the complete solution is found which is time-consuming and prone to numerical errors in high-dimensional systems. In this paper, we characterize degenerate optimal active sets and modify the exploration algorithm such that the complete solution is guaranteed to be found in a general case, which can have degeneracies as well, concurrent with the exploration of combinatorial tree. Simulation results confirm the reliability of the suggested method in finding all critical regions while decreasing the computational time significantly.**

## I. INTRODUCTION

Exploiting the multi-parametric quadratic programming (mpQP) for solving the model predictive control (MPC) problems enables the online computational burden of the problem to be moved offline [1], [2] and [3]. Consequently, application of MPC can be extended to systems with relatively fast dynamics. There are basically two approaches towards solving a mpQP problem. i) Geometric approaches that iteratively build a partition of parameter space using geometric (polyhedral) computations [4]–[8] and [9]. The advantage of these approaches is that mostly optimal combinations of active sets are considered, avoiding unnecessary computations due to the combinatorial number of possible active sets. However, for problems of high dimension of the parameter space, the geometric computations become complex and numerically sensitive and these algorithms, therefore, tend to become slow and unreliable. ii) Computational approaches which are based on implicitly enumerating all possible combinations of active constraints in a combinatorial search tree [10], [11] and [12]. These methods avoid geometric computations and hence deal quite effectively and efficiently with mpQP problems having a higher number of parameters where the geometric methods tend to fail [13]. Another enumeration-based method for solving linear and semi-definite quadratic multi-parametric programs is recently proposed in [14] based on reformulating these problems into parametric linear complementarity problems. This method has shown to be twice as fast as the method of [10]. The pruning criterion in all these methods is to simultaneously cut off branches with infeasible active sets which is crucial for achieving optimal efficiency in enumeration. A drawback of these methods, however, is that the number of possible combination of active constraints increases exponentially with the number of constraints. Therefore, its applications is limited to problems with few constraints [12]. In order to exclude from the combinatorial tree a noticeable number of feasible active constraints that are not optimal, [15] has suggested a downward and upward exploration of combinatorial tree which exploits some of the underlying relationship between two full-dimensional adjacent critical regions. This method is guaranteed to find all critical regions in the non-degenerate cases while reducing the number of LPs that should be solved. Hence the required computational time decreases significantly. For degenerate cases, however, some critical regions may not be found using the algorithm in [15]. Therefore, a post-processing algorithm which is relatively time-consuming and based on the geometric operations (using [16]) is performed in [15] to assure that all critical regions are found. However, the main drawback of this method is that the post-processing algorithm should be performed in both degenerate and non-degenerate cases, while it is not necessary in the later case. This is because the algorithm does not identify the existence of degeneracy on the common facets between adjacent CRs. In order to avoid the unnecessary post-processing, a study about full-dimensional critical regions for which degeneracy occurs on their common facet is presented in this paper. Exploiting these properties for adjacent CRs, one can identify, while exploring the combinatorial tree, whether degeneracy happens for some candidate active sets in the mpQP or not. In addition an alternative method for degenerate cases is suggested to assure finding all optimal active sets without requiring the post-processing check. Hence, this paper is structured as follows. The combinatorial

*The work leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 607957 (TEMPO).

*This work was supported by the Research Council of Norway, Statoil, DNV GL and Sintef through the Centers of Excellence funding scheme, Grant 223254 - Centre for Autonomous Marine Operations and Systems (AMOS).

[1]Center for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, NTNU, O.S. Bragstads plass, 7491 Trondheim, Norway parisa.ahmadi.moshkenani@itk.ntnu.no, tor.arne.johansen@itk.ntnu.no

[2] Laboratory of Signals and Systems, CentraleSupélec-CNRS-Université Paris Sud, Univeristé Paris-Saclay Gif-sur-Yvette, France sorin.olaru@supelec.fr

approach towards mpQP is briefly explained in section II in conjunction with the suggested downward and upward exploration of the combinatorial tree in [15]. The new approach for exploring the combinatorial tree is presented in section III together with a series of theorems describing the theoretical foundation. In section IV the simulation results are presented which show the efficiency of the suggested algorithm, and finally the paper is concluded in section V.

## II. COMBINATORIAL APPROACH TOWARDS MULTI-PARAMETRIC QUADRATIC PROGRAMMING

Consider the standard multi-parametric quadratic program:

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

$$\text{s.t.} \quad Gz \leq Sx + W \tag{1b}$$

where $z \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ denote the optimization variables and parameters, respectively. Assume that the problem is strictly convex, i.e. $H > 0$ and all constraints are irredundant. As shown by [1], the Karush-Kuhn-Tucker (KKT) optimality conditions can be used to characterize the analytic solutions to the mpQP problem:

$$Hz + G^T \lambda = 0, \qquad \lambda \in \mathbb{R}^q, \tag{2a}$$

$$\lambda^i (G^i z - W^i - S^i x) = 0, \qquad i = 1, \ldots, q, \tag{2b}$$

$$\lambda \geq 0, \quad Gz \leq Sx + W \tag{2c}$$

Defining $\mathcal{Q} = \{1, \ldots, q\}$ as the index set of all constraints in (1b), we recall that a constraint among $q$ constraints in (1b) is said to be *active* if it holds with equality for a given $z$ and $x$, and *inactive* if it holds with strict inequality. Thus the active set $\mathcal{A}(z, x)$ can be described as $\mathcal{A}(z, x) := \{i \in \mathcal{Q} \mid G^i z - S^i x - W^i = 0\}$ while the corresponding inactive set $\mathcal{J}(z, x)$ is given by the set difference of $\mathcal{Q}$ and $\mathcal{A}$ i.e. $\mathcal{J}(z, x) := \mathcal{Q} \backslash \mathcal{A}(z, x)$. Before going further, we recall some definitions and theorems.

**Definition 1.** Redundant constraints: Let a polyhedron $X$ be represented by $Ax \leq b$. We say that $A^i x \leq b^i$ is *redundant* if $A^j x \leq b^j, \forall j \neq i \Rightarrow A^i x \leq b^i$ (i.e., it can be removed from the description of the polyhedron).

**Definition 2.** Minimal representation: A representation of a polyhedron is minimal if there are no redundant constraints.

**Definition 3.** Linear Independence Constraints Qualification (LICQ), (Nocedal and Wright, 1999): Given $z^*(x)$ as the optimal solution of (1) at which KKT conditions are satisfied and the corresponding active set $\mathcal{A}$, we say that LICQ holds if the set of active constraint gradients $\{G^i \mid i \in \mathcal{A}(z^*(x), x)\}$ is linearly dependent, i.e., $G^{\mathcal{A}}$ has full row rank.

**Definition 4.** Strict Complementarity Slackness (SCS), (Nocedal and Wright, 1999): Given the pair $(z^*(x), \lambda^*(x))$ satisfying the KKT conditions, SCS holds if exactly one of $\lambda^{i*}(x)$ and $G^i z^*(x) - S^i x - W^i$ is zero for each $i \in \mathcal{Q}$, i.e., $\lambda^{i*}(x) > 0$ for each $i \in \mathcal{A}(z, x)$ and $s^i > 0$ for each $i \in \mathcal{J}(z, x)$ where $s^i$ is the slack variable of inactive constraint $i \in \mathcal{J}$ such that $G^i z^*(x) + s^i = S^i x + W^i$ .
We define as *weakly active constraint* an active constraint with an associated zero Lagrange multiplier $\lambda^{i*}$ and as *weakly inactive constraint* an inactive constraint with an associated zero slack variable $s^i$. Furthermore, an optimization problem for which both the LICQ condition and the SCS condition hold is known to be *non-degenerate* according to the definition of degeneracy in [5].

**Definition 5.** Full-dimensional polyhedron: Let $x_0 \in X \subseteq \mathbb{R}^n$. If $dim(x_0) = n$, we say that $X$ is *full-dimensional*.

***Theorem 1:***
*Consider the problem in* (1) *with* $H > 0$. *Let* $X \subseteq \mathbb{R}^n$ *be a polyhedron, i.e., the problem's feasible set and let* $x \in X$. *Then the solution* $z^*(x)$ *and the Lagrange multipliers* $\lambda^*(x)$ *of a mpQP are piecewise affine functions of the parameter* $x$ *and* $z^*(x)$ *is continuous. Moreover, if LICQ holds for all* $x \in X$, $\lambda^*(x)$ *is also continuous [1].*

Assuming that we know an optimal active set $\mathcal{A}$ and that LICQ holds, we can use (2a) and (2b) to derive the parameter-dependent optimizer [1]:

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

where the existence of $H_{G^{\mathcal{A}}}^{-1} := (G^{\mathcal{A}} H^{-1} (G^{\mathcal{A}})^T)^{-1}$ is guaranteed due to the LICQ and positive definiteness of $H$. The set of inequalities in (2c) characterize the critical region (CR) for the considered optimal active set $\mathcal{A}$. The CR is in the form of a polyhedron in the parameter space defined by the following inequalities:

$$H_{G^{\mathcal{A}}}^{-1}(W^{\mathcal{A}} + S^{\mathcal{A}} x) \leq 0 \tag{4a}$$

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

This polyhedron is the largest set of parameters $x \in X$ for which the combination of active constraints $\mathcal{A}$ at the optimizer remains unchanged.

To determine all optimal active constraints $\mathcal{A}(z^*(x), x), x \in X$, [10] suggests to choose the candidate active sets from $\mathcal{P}'(\mathcal{Q})$ which is a subset of $\mathcal{P}(\mathcal{Q})$ including all the subsets of $\mathcal{Q}$ with maximum $\tilde{m} = min\{m, q\}$ members (note that as pointed out by [10], for a mpQP with $m$ decision variables ($z \in \mathbb{R}^m$) and $q$ constraints, only a maximum of $\tilde{m}$ linearly independent constraints can be strongly active at the optimal solution [17]) in the order of increasing cardinality and use the following LP to check whether the candidate set $\mathcal{A}_i$ is indeed optimal or not:

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

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

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

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

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

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

Here $t$ is a scalar optimization variable and $e_1 = [1, \ldots, 1]^T$ and $e_2 = [1, \ldots, 1]^T$ 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. Inequalities (5b) form an upper bound on the optimization variable $t$ as the minimal value contained in $\lambda^{\mathcal{A}_i}$ and $s^{\mathcal{J}_i}$. This formulation allows the immediate identification of failure of the SCS condition whenever $t = 0$. If the candidate active set is found not to be optimal, i.e., if the optimization problem in (5) is not feasible, another optimization problem should be solved by removing all constraints arising from the optimality condition (all constraints including $\lambda^{\mathcal{A}_i}$ in (5)), to check for the feasibility of the candidate active set. If this optimization problem is not feasible, we can exclude $\mathcal{A}_i$ and all its supersets from the combinatorial tree. This is the only pruning criterion in this method which is based on the infeasibility of a combination of active constraints. A graphical illustration of the combinatorial enumeration strategy and the involved pruning process is given in the form of a combinatorial tree diagram in Fig. 2. As it can be seen from Fig. 2, all feasible combinations of active constraints remain in the combinatorial tree for exploring the levels below while for many cases, none of their supersets become optimal in future. In order to exclude from combinatorial tree a noticeable number of feasible candidate active sets which are not optimal, a joint downward and upward exploration of the combinatorial tree is suggested in [15] based on the following theorem from [5].

***Theorem 2 (No Degeneracy):*** *Consider an optimal active set $\{i_1, i_2, \ldots, i_k\}$ and its corresponding minimal representation of the critical region $CR_0$. Let $CR_i$ be a full-dimensional neighbouring critical region to $CR_0$ and assume LICQ holds on their common facet $\mathscr{F} = CR_0 \cap \mathscr{H}$ where $\mathscr{H}$ is the separating Hyperplane between $CR_0$ and $CR_i$. Moreover, assume that there are no constraints which are weakly active at the optimizer $z^*(x)$ for all $x \in CR_0$. Then:*

*Type I: If $\mathscr{H}$ is given by $G^{i_{k+1}} z_0^*(x) = W^{i_{k+1}} + S^{i_{k+1}} x$, then the optimal active set in $CR_i$ is $\{i_1, i_2, \ldots, i_k, i_{k+1}\}$.*

*Type II: If $\mathscr{H}$ is given by $\lambda_0^{i_k}(x) = 0$, then the optimal active set in $CR_i$ is $\{i_1, i_2, \ldots, i_{k-1}\}$.*

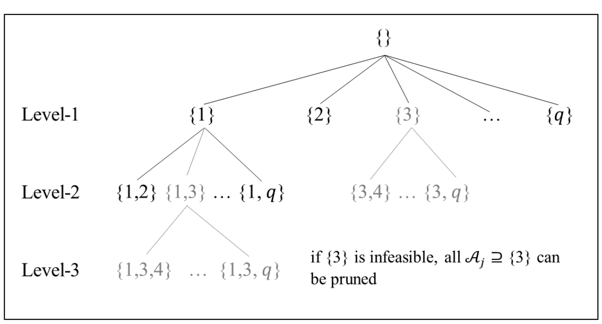


Fig. 1: Combinatorial enumeration strategy used in [10]

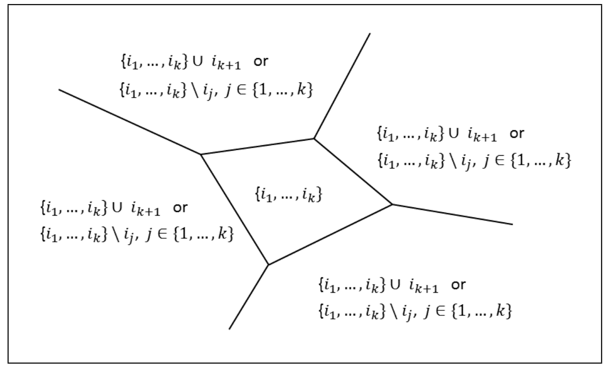


Fig. 2: Combination of optimal active constraints in adjacent critical regions in a non-degenerate system

According to Theorem 2, the combinations of optimal active sets in two adjacent CRs differ only in one constraint in non-degenerate systems. Therefore, one can only keep the track of optimal active sets and for every found optimal active set with a full-dimensional CR, find all optimal active sets corresponding to its adjacent CRs by adding one feasible constraint to or removing one existing constraint from the current optimal active set (See Fig. 2 for illustration). Doing this for all found optimal active sets, guarantees finding the complete solution in non-degenerate cases due to the fully connected critical regions which partition the feasible space. Therefore, this method for finding optimal active constraints requires joint downward and upward exploration of the combinatorial tree. To this aim, one can explore the combinatorial tree as before, in the order of increasing cardinality. The difference is that in this method, we only use the optimal active sets for building the levels below (downward exploration). Hence if a combination of active constraints is not optimal, the feasibility check of LP (5) is not needed any more. For every optimal active set found during downward exploration, we should explore the combinatorial tree upward to check for the optimality of all its subsets with one element less if they are not enumerated yet (upward exploration). Then for every newly found optimal set during upward exploration, we should explore the combinatorial tree downward and upward again, until no new non-enumerated combination is found. For each eliminated feasible but not optimal combination of active constraints, the number of LPs in the form of (5) that should be solved decreases by two (one for checking the optimality and the other for checking the feasibility of the candidate active set). However, when degeneracy happens for some combinations of optimal active constraints, some

CRs may remain unexplored using this procedure. One way to handle this limitation is to do a post-processing, using geometric approaches, to find the regions that maybe are missed as it is suggested in [15]. In the next section, we suggest an alternative approach for handling degenerate cases rather than post processing, which is not based on geometric operations and hence is faster and more reliable when the number of parameter variables and the number of constraints increases.

## III. NEW APPROACH FOR DEGENERATE CASES

Theorem 2 implies that when the optimal active sets in two adjacent full-dimensional CRs differ in more than one constraint, at least one of the LICQ condition or SCS condition is violated. Regarding different possibilities for the combination of optimal active constraints in two adjacent critical regions when the LICQ and/or SCS conditions do not hold on their common facet, we divide them into two categories. By Categ.I, we mean those adjacent CRs in which the combinations of optimal active constraints in two adjacent CRs differ in more than one constraint and each CR has *only* one constraint in addition to a subset of the optimal constraints in the adjacent CR. For example, two neighbouring CRs with the corresponding optimal sets $\mathcal{A}_i = [i_1, \dots i_k, i_{k+1}]$ and $\mathcal{A}_j = [i_1, \dots i_k, i_{k+2}]$ lie in this category. On the other hand, all adjacent CRs for which the combination of optimal active constraints differ in more than one constraint and at least one of the CRs has *more* than one constraint in addition to a subset of the optimal active set in the adjacent CR are classified in Categ.II. For example, adjacent CRs with corresponding optimal sets $\mathcal{A}_i = [i_1, \dots i_k, i_{k+1}, i_{k+2}]$ and $\mathcal{A}_j = [i_1, \dots i_k]$, $\mathcal{A}_i = [i_1, \dots i_k, i_{k+1}, i_{k+2}]$ and $\mathcal{A}_j = [i_1, \dots i_k, i_{k+3}]$, $\mathcal{A}_i = [i_1, \dots i_k, i_{k+1}, i_{k+2}]$ and $\mathcal{A}_j = [i_1, \dots i_k, i_{k+3}, i_{k+4}]$ lie in Categ.II.

For all adjacent CRs that are classified in Categ.I, one of the following two cases can happen on their common facet which is stated as Corollary 1.

***Corollary 1 (Categ.I degeneracy)***
*Let two full-dimensional neighbouring CRs with the minimal representation be classified as Categ.I, i.e., the optimal active sets in these two regions can be defined by* $\mathcal{A}_i = [i_1, \dots i_k, i_{k+1}]$ *and* $\mathcal{A}_j = [i_1, \dots i_k, i_{k+2}]$. *Then one of these conditions holds:*
*a) LICQ is violated for the combination of optimal active constraints on their common facet.*
*b) LICQ holds on the common facet and SCS is violated there, which implies that some of the constraints are weakly active or weakly inactive.*

***Proof:*** The proof follows directly from Theorem 2, i.e., since the combinations of the optimal active constraints in two adjacent CRs differ in more than one constraint, then at least one of the LICQ or SCS conditions is violated on their common facet. Hence either LICQ condition is violated on the common facet (a), or if LICQ condition holds there, then SCS condition is violated (b). ■

Theorem 3 describes the characteristic of combinations of active constraints on the common facet between two CRs that are classified as Categ.II.

***Theorem 3 (Categ.II degeneracy)***
*Let two full-dimensional neighbouring CRs be classified as Categ. II, i.e., the optimal active set in one of the regions have at least two constraints in addition to a subset of optimal active constraints in the adjacent CR. Then the SCS condition is violated on the common facet* $\mathcal{F}$ *between these two critical regions.*

***Proof:*** Let us denote the critical region which has at least two constraints in addition to a subset of the optimal active set in the neighbouring critical region as $CR_i$, those two additional constraints as $i_{k+1}$ and $i_{k+2}$ and $\mathcal{A}_j$ as the optimal active set in the adjacent critical region $CR_j$. It can be proved that $\mathcal{A}_{\mathcal{F}_1} \triangleq \mathcal{A}_j \cup i_{k+1}$ is an optimal active set on the common facet with the associated critical region $CR_{\mathcal{F}_1}$ due to feasibility of the LP in (5) with $\mathcal{A}_j$ for all $x \in \mathcal{F}$ and the trivial value for $\lambda^{i_{k+1}}$ equal to zero (Note that $\lambda^{i_{k+1}} = 0$ gives a feasible point for LP in (5) with $\mathcal{A}_{\mathcal{F}_1} \triangleq \mathcal{A}_j \cup i_{k+1}$ which guarantees the optimality of $\mathcal{A}_{\mathcal{F}_1}$ there. But it doesn't mean that the obtained optimal value for $\lambda^{i_{k+1}}$ should be necessarily zero). Similarly it can be proved that $\mathcal{A}_{\mathcal{F}_2} \triangleq \mathcal{A}_j \cup i_{k+1} \cup i_{k+2}$ is an optimal active set on $\mathcal{F}$ with the trivial values $\lambda^{i_{k+1}} = \lambda^{i_{k+2}} = 0$ in (5) and the corresponding critical region $CR_{\mathcal{F}_2}$. Since the optimizer $z^*(x)$ is unique due to positive definiteness of $H$, for all $x \in \mathcal{F}$ we have that $G^{k+2}z^*(x) + s^{k+2} = S^{k+2}x + W^{k+2}$ with some $s^{k+2} \geq 0$ as $x \in CR_{\mathcal{F}_1}$ and simultaneously we have $G^{k+2}z^*(x) = S^{k+2}x + W^{k+2}$ as $x \in CR_{\mathcal{F}_2}$. This means that $s^{k+2} = 0$ for all $x \in CR_{\mathcal{F}_1}$, which means that $i_{k+2}$ is weakly active on $\mathcal{F}$. ■

**Remark 1.** Whenever the facet-to-facet property [18] does not hold for two adjacent critical regions, the same results as in Corollary 1 and Theorem 3 still holds by substituting $\mathcal{F}$ with the part of the facet that is common between $CR_i$ and $CR_j$ in the proofs.

Exploiting the results in Corollary 1 and Theorem 3, we can now modify the downward-upward algorithm in [15] such that the degenerate cases are explicitly considered and therefore, all critical regions are found during exploration of the combinatorial tree while the number of LPs needed to be solved reduces significantly. To this aim, in the downward-upward exploration we also consider combinations of active constraints for which either LICQ condition or SCS condition is violated. If in the exploration of entire tree, no combination of active constraints with failure in SCS condition is found, then due to Theorem 3, no adjacent CRs which can be classified as Categ.II exists in the whole partitioned feasible parameter domain. The only possibility for the combinations of optimal active sets in two adjacent CRs, apart from the non-degenerate cases, is due to Corollary 1-a. Hence if we explore the combinatorial tree up to level-$(\tilde{m}+1)$ where $\tilde{m} = min\{m, q\}$ (as such case may happen in the last level of the combinatorial tree ($\tilde{m}$) with the violation of LICQ condition in the optimal active set in level-$(\tilde{m}+1)$ which forms the common facet between these two adjacent critical regions), and consider combinations of optimal active constraint for which LICQ is violated. Then for all such cases explore their subsets which have one constraint less and are not explored yet, all critical regions will be found.

If the SCS condition fails for some combinations of active constraints in a full-dimensional CR or in a low-dimensional CR which corresponds to the common facet between full-dimensional CRs, identifying the combination of optimal active constraints in the adjacent CR is not straightforward. Specially because of the possibility of many overlapping low-dimensional CRs which leads to a significantly different combination of active constraints in the full-dimensional adjacent CR. To deal with such situations, a similar method to what is suggested in [19] can be exploited. To this aim, we can determine the set including all constraints that are active or weakly inactive for each optimal active set with violation of SCS condition and then explore all its unexplored full row rank subsets which have at most $\tilde{m}$ elements since potentially every combination of these constraints can appear in a full-dimensional adjacent critical region depending on which low-dimensional critical regions with violation of SCS overlap on the common facet. Note that the indices of all weakly inactive constraints can be simply obtained by identifying all slack variables equal to zero.

The following theorem shows that the optimal active sets for which LICQ is violated need not to be considered in the downward exploration of the combinatorial tree.

***Theorem 4***
*If a superset $\mathcal{A}_l$ of an optimal active set $\mathcal{A}_j$ for which LICQ is violated, is also optimal, then the SCS condition is violated for the optimal active set $\mathcal{A}_j$.*

Theorem 4 guarantees the corresponding optimality check of $\mathcal{A}_l$ via solving the LP in (5) to be performed when dealing with optimal sets with violation in SCS condition. Hence it preserves us from solving the optimization problem (5) for candidate active sets which can arise from exploring the supersets of optimal sets with LICQ violation if it is not needed. Before proceeding with the proof of Theorem 4, we state the following lemma.

***Lemma 1.***
*If the LICQ condition fails for the optimal active set $\mathcal{A}_i$ in a full-dimensional critical region, then all its subsets $\mathcal{A}_j \subset \mathcal{A}_i$ with $G^{\mathcal{A}_j}$ having full row rank, are optimal active sets that are degenerate in the sense of violation of the SCS condition.*

***Proof.*** Assume the full-dimensional critical region $CR_i$ with corresponding optimal set $\mathcal{A}_i = [i_1, \ldots, i_{k-1}, i_k]$ and the Lagrange multipliers $\{\lambda^1, \ldots, \lambda^k\}$ for which LICQ is violated. Assume further that $\mathcal{A}_j = [i_1, \ldots, i_{k-1}]$ is one of its full row rank subsets. This means that the $k^{th}$ row of matrix $G^{\mathcal{A}_i}$ can be written as $G^{\mathcal{A}_i,k} = c_1 G^{\mathcal{A}_i,1} + \ldots + c_{k-1} G^{\mathcal{A}_i,k-1}$ where $G^{\mathcal{A}_i,j}$ represents the $j^{th}$ row of matrix $G^{\mathcal{A}_i}$. Let $x_0$ be a point in the interior of $CR_i$. Then it can be easily proved that $\mathcal{A}_j$ is also the optimal active set at $x_0$ with Lagrange multipliers $\overline{\lambda}^l = \lambda^l + c_l \lambda^k, \forall l \in \{1, \ldots, k-1\}$ and the slack variable corresponding to the $k^{th}$ constraint is equal to zero ($s^k = 0$). Hence $\mathcal{A}_j = [i_1, \ldots, i_{k-1}]$ is an optimal set for which SCS does not hold. ■

Using Lemma 1 we can now prove Theorem 4 as follows.

***Proof:*** Assume that $\mathcal{A}_i = [i_1, \ldots, i_k]$ is an optimal active set with a full-dimensional critical region $CR_i$ where both LICQ and SCS conditions hold for that. Further assume that its superset $\mathcal{A}_j = [i_1, \ldots, i_k, i_{k+1}]$ is an optimal active set with violation of the LICQ condition. By Lemma 1 it is clear that the corresponding critical region $CR_j$ cannot be full-dimensional since otherwise, SCS condition should fail for $\mathcal{A}_i$. Then if $\mathcal{A}_l = [i_1, \ldots, i_{k+1}, i_{k+2}]$ which is built by adding the feasible constraint $i_{k+2}$ to $\mathcal{A}_j$ is also optimal with $CR_l$, two different situations may happen. i) $CR_l$ is low-dimensional: This means that two low-dimensional critical region $CR_j$ and $CR_l$ are neighbouring. Therefore they must overlap. Hence $i_{k+2}$ is weakly inactive for $\mathcal{A}_j$. ii) $CR_l$ is full-dimensional: This means that $CR_i$ and $CR_l$ are two full-dimensional CRs which are adjacent. Therefore they lie in the Categ.II and the SCS condition fails on their common facet (with $\mathcal{A}_j$) as a result of Theorem 3. ■

Based on the above explanations, the modified downward-upward algorithm suggested in [15] can be summarized as in Algorithm 1.

## IV. SIMULATION RESULTS

In this section, the results of the combinatorial approach using the suggested method in Algorithm 1 are shown for the four tank system with 4 state variables and 2 inputs and the fuel cell breathing control system with 8 state variables and 1 input which are used in [15]. These examples do not have conditions in which SCS fails. However, the condition in Corollary 1-a can occur, as it happens in the example of fuel cell system with $N = 6$ in which $\tilde{m} = 3$ and $\mathcal{A}_i = [3, 11, 13]$ and $\mathcal{A}_j = [11, 13, 16]$ are the optimal sets in two full-dimensional adjacent CRs and $\mathcal{A}_{\mathcal{F}} = [3, 11, 13, 16]$ is the optimal set on their common facet with the violation of the LICQ condition as $|\mathcal{A}_{\mathcal{F}}| > \tilde{m}$. The simulation results using the algorithm in [10], implemented in MPT3, and using the method which is suggested here for a four tank system and a fuel cell breathing system on a 3.2 GHz core i5 CPU

**Algorithm 1** Downward-upward exploration strategy of the combinatorial tree

**Phase I (Initialization):**

1) $i = 1$, Explore the entire level-1, use (5) to check the optimality of each constraint. For each optimal constraint with violation of the SCS condition, create its superset including the active and all weakly inactive constraints and store it in "*SCS Set*". If the constraint is not optimal, use (5) without optimality conditions to check the feasibility of that constraint. Store all optimal constraints for which the SCS condition holds in "*Optimal Set*" and all feasible constraints, whether they are optimal or not, in "*Feasible Set*";
   - ∟ if no constraint is found to be optimal without violation of SCS condition in 1), then:
     - ∟ while *Optimal Set* is empty, explore the entire level-$(i+1)$, check *only* for optimality of the generated combinations. For each found optimal set with violation in SCS condition, create its superset including all active and weakly inactive constraints and store them in *SCS Set*;
     - ∟ $i := i + 1$;

**Phase II (Recursive Exploration):**

2) *(Downward Exploration)* Construct level-$(i + 1)$ by adding one feasible constraint from level-1 to all sets in *Optimal Set* which are found in level-$i$ and check *only* for the optimality of new combinations whether LICQ holds for them or not. For each found optimal active set:
   - ∟ if both LICQ and SCS hold
     compute control law and critical region and add the combination to *Optimal Set*;
   - ∟ elseif SCS fails
     compute the superset including all active and weakly inactive constraints and add it to *SCS Set*;
   - ∟ elseif LICQ fails
     add it to *LICQ Set* to check its subsets with one element less to find possibly missed CRs as in Corollary 1-a;

   $i := i + 1$;
   ∟ if $i < \tilde{m} = min\{m, q\}$ then go to 2), else go 3);
3) *(Upward Exploration)* For all found optimal active sets in *Optimal Set* and *LICQ Set*, check the optimality of all its subsets with one element less that have not been enumerated yet. Store all newly found optimal sets in "*New Set*";
4) For each optimal set $\mathcal{A}_i \in$ *New Set*:
   *New Set* := *New Set* $\backslash \mathcal{A}_i$
   - ∟ if both LICQ and SCS hold
     add $\mathcal{A}_i$ to *Optimal Set* and compute the corresponding critical region and control law, check the optimality of all its subsets with one element less and supersets with one element more that have not been enumerated yet (joint upward and downward exploration of the tree for a newly found non-degenerate optimal set). Add all found optimal sets to *New Set*;
   - ∟ elseif SCS fails
     compute the superset including all active and weakly inactive constraints and add it to *SCS Set*;
   - ∟ esleif LICQ fails
     add it to *LICQ Set* and explore only its subsets with one element less and check for the optimality, add all found optimal sets to *New Set*;
   - ∟ if *New Set* is empty then go to 5), else go to 4);

**Phase III (Handling Cases with SCS Violation):**

5) Compute the union of all sets in *SCS Set*. Explore all its full row rank subsets with cardinality less than or equal to $\tilde{m}$ if it is not enumerated yet;
   - ∟ add all newly found optimal active set for which SCS holds to *Optimal Set* and go to 4);
   - ∟ if no new set for which SCS holds is found, stop

running MATLAB 2014a are shown in Table I and Table II, respectively, where $N$, $n_{CR}$ and $n_{LP}$ represent the prediction (and control) horizon, number of found CRs and number of solved LPs. The LP solver in the suggested method, i.e. Alg. 2, is chosen to be GLPK which is intended for solving large-scale linear programmings. The last column shows the ratio of the computational time using the suggested algorithm to the computational time using algorithm in [10]. It can be seen that as the prediction horizon increases, this ratio decreases dramatically which indicates the superiority of the suggested algorithm for systems with a large number of constraints.

As an example for cases with violation of SCS condition, we augmented example 1 from [18] by adding random matrices to $G$, $S$ and $W$ such that the number of inputs and the number of constraints are increased in the problem. Table III shows the comparison for four different randomly augmented examples for which SCS condition fails for some

TABLE I: Comparison between different algorithms for four tank system from [19]

| Method | $N$ | $n_{CR}$ | $n_{LP}$ | $t_{comp}[s]$ | $\frac{t_{Alg.2}}{t_{Alg.1}}$ |
|---|---|---|---|---|---|
| Alg. 1 | 2 | 62 | 679 | 1.5600 | |
| Alg. 2 | | 62 | 456 | 1.7743 | 1.1373 |
| Alg. 1 | 3 | 221 | 17282 | 19.1400 | |
| Alg. 2 | | 221 | 2709 | 6.9700 | 0.3642 |
| Alg. 1 | 4 | 605 | 374470 | 415.8050 | |
| Alg. 2 | | 605 | 10540 | 25.1268 | 0.0604 |
| Alg. 1 | 5 | 1393 | 7193402 | 11185 | |
| Alg. 2 | | 1393 | 31639 | 80.0446 | 0.0072 |
| Alg. 1 | 6 | 1432* | 25699816 | 24h* | |
| Alg. 2 | | 2744 | 77579 | 239.6348 | - |

* The code execution is manually stopped after 24 hours. Only 1432 CRs are found by solving 25699816 LPs while the actual number of LPs to be solved for the complete solution will be substantially larger.
Alg.1: Algorithm by Gupta/Feller
Alg.2: Algorithm suggested here

TABLE II: Comparison between different algorithms for fuel cell breathing system

| Method | $N$ | $n_{CR}$ | $n_{LP}$ | $t_{comp}[s]$ | $\frac{t_{Alg.2}}{t_{Alg.1}}$ |
|---|---|---|---|---|---|
| Alg. 1 | 3 | 71 | 287 | 2.0150 | |
| Alg. 2 | | 71 | 574 | 2.7608 | 1.3701 |
| Alg. 1 | 4 | 133 | 1701 | 5.0730 | |
| Alg. 2 | | 133 | 1551 | 5.9922 | 1.1812 |
| Alg. 1 | 5 | 191 | 6001 | 11.4970 | |
| Alg. 2 | | 191 | 2653 | 9.4219 | 0.8195 |
| Alg. 1 | 6 | 241 | 18561 | 29.5160 | |
| Alg. 2 | | 241 | 3888 | 13.2536 | 0.4491 |
| Alg. 1 | 7 | 279 | 47017 | 69.8420 | |
| Alg. 2 | | 279 | 4622 | 16.3629 | 0.2343 |
| Alg. 1 | 8 | 307 | 149319 | 230.5860 | |
| Alg. 2 | | 307 | 5840 | 19.6120 | 0.0851 |

Alg.1: Algorithm by Gupta/Feller
Alg.2: Algorithm suggested here

TABLE III: Comparison between different algorithms for the system with violation in the SCS condition

| Method | $n_z$ | $q$ | $n_{CR}$ | $n_{LP}$ | $t_{comp}[s]$ | $\frac{t_{Alg.2}}{t_{Alg.1}}$ |
|---|---|---|---|---|---|---|
| Alg. 1 | 4 | 10 | 34 | 452 | 0.8500 | |
| Alg. 2 | | | 34 | 263 | 2.0614 | 2.4252 |
| Alg. 1 | 6 | 20 | 54 | 14376 | 16.6200 | |
| Alg. 2 | | | 54 | 925 | 3.7631 | 0.2264 |
| Alg. 1 | 8 | 30 | 70 | 223211 | 423.8800 | |
| Alg. 2 | | | 70 | 1763 | 7.3106 | 0.0172 |
| Alg. 1 | 10 | 40 | - | 6439332* | 5h* | |
| Alg. 2 | | | 79 | 2835 | 9.6660 | - |

* Matlab ran out of memory in the ninth-level, after approximately 5 hours of execution and solving 6439332 LPs.
Alg.1: Algorithm by Gupta/Feller
Alg.2: Algorithm suggested here

of the combinations of active constraints. Here $n_z$, $q$, $n_{CR}$, $n_{LP}$, and $t_{comp}$ represent the number of control variables, number of constraints, number of found CRs, number of solved LPs, and the computational time required by different algorithms, respectively. It can be seen that the suggested algorithm has a significant reduction of computational time in comparison with the algorithm in [10] for the combinatorial approach and as the the number of control variables and the number of constraints increase, the superiority of the suggested algorithm becomes significantly noticeable.

## V. Conclusion

In this paper, a new method for exploration of the combinatorial tree in combinatorial mpQP was suggested which is based on exploiting the information about full-dimensional adjacent critical regions. By excluding a great number of feasible but not optimal combination of active constraints from the combinatorial tree, the computational time decreases dramatically. All critical regions in both non-degenerate and degenerate cases are guaranteed to be found without requiring a post-processing algorithm which is time-consuming and may cause numerical problems in high-dimensional parameter spaces. Therefore the suggested method is well-suited for explicit MPC of high order systems with a large number of constraints.

## References


[1] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[2] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear model predictive control*. Springer, 2009, pp. 345–369.

[3] E. Pistikopoulos, "Perspectives in multiparametric programming and explicit model predictive control," *AIChE journal*, vol. 55, no. 8, pp. 1918–1925, 2009.

[4] M. Baotić, "An efficient algorithm for multiparametric quadratic programming," *Technical Report AUT02-05, ETH Zürich, Institute für Automatik, Physikstraße 3, CH-8092, Switzerland*, 2002.

[5] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit mpc solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[6] ——, "Further results on multiparametric quadratic programming," in *Proc. 42nd IEEE Conf. on Decision and Control*, vol. 3. Citeseer, 2003, pp. 3173–3178.

[7] M. M. Seron, G. C. Goodwin, and J. A. Doná, "Characterisation of receding horizon control for constrained linear systems," *Asian Journal of Control*, vol. 5, no. 2, pp. 271–286, 2003.

[8] S. B. Olaru and D. Dumur, "A parameterized polyhedra approach for explicit constrained predictive control," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2. IEEE, 2004, pp. 1580–1585.

[9] C. N. Jones and M. Morrari, "Multiparametric linear complementarity problems," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 5687–5692.

[10] A. Gupta, S. Bhartiya, and P. Nataraj, "A novel approach to multiparametric quadratic programming," *Automatica*, vol. 47, no. 9, pp. 2112–2117, 2011.

[11] C. Feller, T. A. Johansen, and S. Olaru, "An improved algorithm for combinatorial multi-parametric quadratic programming," *Automatica*, vol. 49, no. 5, pp. 1370–1376, 2013.

[12] ——, "Combinatorial multi-parametric quadratic programming with saturation matrix based pruning," in *IEEE 51st Annual Conference on Decision and Control (CDC), 2012*, pp. 4562–4567.

[13] C. Feller and T. A. Johansen, "Explicit mpc of higher-order linear processes via combinatorial multi-parametric quadratic programming," in *European Control Conference (ECC), 2013*, pp. 536–541.

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

[15] P. Ahmadi-Moshkenani, S. Olaru, and T. A. Johansen, "Further results on the exploration of combinatorial tree in multi-parametric quadratic programming," in *Proceedings of European Control Conference (ECC), 2016*.

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

[17] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[18] 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*, vol. 42, no. 12, pp. 2209–2214, 2006.

[19] C. Feller, "A non-geometric approach to multi-parametric programming," Master's thesis, Diploma Thesis IST-80, University of Stuttgart, 2011.