Reduced Memory Footprint in Multiparametric Quadratic Programming by Exploiting Low Rank Structure

Isak Nielsen and Daniel Axehill

Conference Publication

N.B.: When citing this work, cite the original article.

Original Publication:

ISBN: 9781509018376 (online), 9781509018444 (DVD) and 9781509018383 (PoD)

Copyright: Institute of Electrical and Electronics Engineers (IEEE)

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Postprint available at: Linköping University Electronic Press

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-133641
Reduced Memory Footprint in Multiparametric Quadratic Programming by Exploiting Low Rank Structure

Isak Nielsen and Daniel Axehill

Abstract—In multiparametric programming an optimization problem which is dependent on a parameter vector is solved parametrically. In control, multiparametric quadratic programming (mp-QP) problems have become increasingly important since the optimization problem arising in Model Predictive Control (MPC) can be cast as an mp-QP problem, which is referred to as explicit MPC. One of the main limitations with mp-QP and explicit MPC is the amount of memory required to store the parametric solution and the critical regions. In this paper, a method for exploiting low rank structure in the parametric solution of an mp-QP problem in order to reduce the required memory is introduced. The method is based on ideas similar to what is done to exploit low rank modifications in generic QP solvers, but is here applied to mp-QP problems to save memory. The proposed method has been evaluated experimentally, and for some examples of relevant problems the relative memory reduction is an order of magnitude compared to storing the full parametric solution and critical regions.

I. INTRODUCTION

In parametric programming the optimization problem is dependent on a parameter which can be thought of as input data to the optimization problem [1]. When the optimization problem is dependent on several parameters it is referred to as multiparametric programming, and one class of such problems that has proven to be important is multiparametric quadratic programming (mp-QP) problems. See, e.g., [1] for a survey on parametric programming. In control, the importance of mp-QP has increased since it was shown in [2] that the optimization problem in Model Predictive Control (MPC) can be cast as an mp-QP problem and solved explicitly.

MPC is a control strategy where the control input in each sample is computed as the solution to a constrained finite-time optimal control (CFTOC) problem, [3]. The CFTOC problem is solved on-line in each sample of the control loop, which requires efficient algorithms for solving the optimization problem. Examples of algorithms where the special structure in MPC problems is exploited are [4]–[6].

Solving the mp-QP problem that corresponds to the CFTOC problem prior to the on-line execution is referred to as explicit MPC, and the solution is explicitly given as a function of the parameter. For a strictly convex mp-QP problem, the parametric solution is a piecewise affine (PWA) function of the parameters over polyhedral critical regions, [2]. In [2], an algorithm for computing the solution to the mp-QP problem is presented. The on-line computational effort consists of evaluating the PWA function for a given parameter [2], which allows for a division free implementation of the control law that can be computed within an a priori known worst case execution time [7].

However, there are limitations with mp-QP and explicit MPC, and much focus in research has been spent to overcome them. The main limitations are the computation of the PWA function and the critical regions, the computation of a data structure which provides efficient lookup, the memory requirement to store the parametric solution and the critical regions, and the time consumed to determine which critical region the parameter belongs to, [2], [7]–[9]. In [2] and [10] the critical regions and the corresponding optimal active sets are determined by geometric approaches for exploring the parameter space. The algorithm in [10] exploits the relation between neighboring critical regions and the optimal active sets, and it is reported to avoid unnecessary partitioning. In [11] an approach to solve mp-QP problems by using an implicit enumeration of all possible optimal active sets prior to the construction of the critical regions is proposed. The algorithm provides a partition of the full parametric space without unnecessary partitioning. In [12] a method for solving multiparametric linear complementarity problems is presented. This class of problems include mp-QP problems, but also extends to more general problems.

An algorithm which combines explicit MPC and online MPC is proposed in [13]. Here, the main algorithm is similar to a standard active set method such as the one presented in, e.g., [14], but the search directions are computed offline for all optimal active sets. In [15] a PWA function, which is only defined over the regions with non-saturated control inputs, is combined with a projection onto a non-convex set to reduce the memory footprint in explicit MPC. The method of implicitly enumerating all optimal active sets proposed in [11] and the semi-explicit approach in [13] is combined in [7], where the authors propose an algorithm that reduces the memory footprint in explicit MPC.

A commonly used algorithm for improving the on-line process of evaluating the PWA function is given in [8]. The authors propose an algorithm based on a binary search tree, which provides evaluation times that are logarithmic in the number of regions. In [16] a graph traversal algorithm is used to evaluate PWA functions and the graph is constructed while solving the mp-QP problem. In [9] the point location problem is solved by the use of linear decision functions, and significantly better performance in terms of computational time at a small cost of increased memory has been reported.

Furthermore, the Multiparametric Toolbox (MPT) is an open source MATLAB-based toolbox for multiparametric optimization problems, [17].

I. Nielsen and D. Axehill are with the Division of Automatic Control, Linköping University, SE-58183 Linköping, Sweden, isak.nielsen@liu.se, daniel.axehill@liu.se.
The main contribution in this paper is the introduction of theory and algorithms for exploiting low rank structure in the parametric solution between neighboring critical regions for an mp-QP problem. The proposed method can significantly reduce the amount of memory required to store the solution and critical regions by exploiting that only low rank modifications of the parametric solution is obtained when making minor changes to the optimal active sets. In methods for solving general QP problems, exploiting low rank structure has been a crucial approach to improve the performance [14]. However, to the authors' knowledge, this structure has not yet been known to exist also in the parametric solution of an mp-QP problem. The method stores the solution in a tree structure and can be incorporated directly in already existing mp-QP solvers, or be applied as a post-processing step to an already existing solution in order to reduce the required memory. Hence, the approach presented here can be interpreted as a data compression algorithm. The problem of solving the mp-QP problem has been considered in previous work by other authors and is outside the scope of this paper.

In this paper $S_+^{n 	imes n}$ denotes symmetric positive (semi) definite matrices with $n$ columns, $Z_{a,j} \triangleq \{i, i+1, \ldots, j\}$ and symbols in sans-serif font (e.g. $x$) denote matrices of stacked components. $|S|$ denotes the cardinality of a set $S$.

II. MULTIPARAMETRIC QUADRATIC PROGRAMMING

In this section the basics of mp-QP are surveyed, and notation that will be used in the following sections is introduced. Consider an optimization problem in the form

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} u^T H u + \theta^T gu \\
\text{subject to} & \quad Gu \leq b + E\theta, \quad \theta \in \Theta,
\end{align*}$$

(1)

where $u \in \mathbb{R}^m$ is the optimization variable, $\theta \in \mathbb{R}^n$ is the parameter, the cost function is determined by $H \in S_+^{n \times n}$ and $g \in \mathbb{R}^{n \times m}$, the inequality constraints are given by $G \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ and $E \in \mathbb{R}^{p \times n}$, and $\Theta$ is a polyhedral set.

The problem (1) is an mp-QP problem with parameter $\theta$, see, e.g., [1], [2]. By introducing the change of variables $z \triangleq u + H^{-1}g^T \theta$, the problem (1) can be transformed into the equivalent mp-QP problem

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} z^T Hz \\
\text{subject to} & \quad Gz \leq b + S\theta, \quad \theta \in \Theta,
\end{align*}$$

(2)

where $S \triangleq E + GH^{-1}g^T \in \mathbb{R}^{p \times n}$. For any choice of parameter $\theta \in \Theta$, the problem (2) is a strictly convex QP problem, and the necessary and sufficient optimality conditions are given by the Karush-Kuhn-Tucker (KKT) conditions

$$\begin{align*}
H z + G^T \lambda &= 0, \quad (3a) \\
G z &\leq b + S\theta, \quad (3b) \\
\lambda &\geq 0, \quad (3c) \\
\lambda_k (G_k z - b_k - S_k \theta) &= 0, \quad k \in \mathbb{Z}_{1,p}, \quad (3d)
\end{align*}$$

where $\lambda \in \mathbb{R}^p$ are the Lagrange multipliers associated with the inequality constraints, [14].

It is shown, e.g., [2] that the solution to the mp-QP problem (2) is given by the PWA function

$$z^*(\theta) = k_i + K_i \theta \quad \text{if} \quad \theta \in \mathcal{P}_i, \ i \in \mathbb{Z}_{1,R},$$

(4)

where $k_i \in \mathbb{R}^m$, $K_i \in \mathbb{R}^{m \times n}$ define the parametric solution in the polyhedral critical region $\mathcal{P}_i$ for $i \in \mathbb{Z}_{1,R}$. The PWA function (4) and the $R$ critical regions are computed by solving the mp-QP problem (2) parametrically, basically by considering all possible combinations of optimal active constraints, which is explained in, e.g., [2], [10], [11].

A. Computing the solution for an optimal active set

For any feasible choice of the parameter $\theta \in \Theta$, the set of indices of constraints that hold with equality at the optimum is called the optimal active set. Assume that no constraints in the optimal active set are weakly active, i.e., no constraints $k$ in the optimal active set have $\lambda_k = 0$. How to choose the optimal active sets in the case of weakly active constraints can be seen in, e.g., [10]. Let the set of all optimal active sets be denoted $\mathcal{A}$ and let the elements in $\mathcal{A}$ be denoted $\mathcal{A}_i$, for $i \in \mathbb{Z}_{1,R}$. Each optimal active set $\mathcal{A}_i$ then corresponds to a critical region $\mathcal{P}_i$, [10], [11]. Let $\mathcal{N}_i$ be the set of inactive constraints, satisfying $\mathcal{A}_i \cup \mathcal{N}_i = \mathcal{Z}_{1,p}$ and $\mathcal{A}_i \cap \mathcal{N}_i = \emptyset$.

To clarify the relation between the optimal active set and the corresponding solution and critical region, let $G_{\mathcal{A}_i}$ and $S_{\mathcal{N}_i}$ be the matrices consisting of the rows in $S$ indexed by $\mathcal{A}_i$ and $\mathcal{N}_i$, respectively, and let the same hold for $S_{\mathcal{A}_i}$, $S_{\mathcal{N}_i}$, $b_{\mathcal{A}_i}$ and $b_{\mathcal{N}_i}$. Furthermore, let $G_{\mathcal{A}_i}$ have full row rank, i.e., the linear independence constraint qualification (LICQ) holds for $\mathcal{A}_i$, [10]. Violation of LICQ is referred to as primal degeneracy, and how to handle this can be seen in, e.g., [2]. If LICQ is violated but $G_{\mathcal{A}_i}^T z = b_{\mathcal{A}_i} + S_{\mathcal{A}_i} \theta$ are linearly independent, it results in a non full dimensional critical region that in general is a facet between full dimensional regions and need not be considered here, [2], [10]. For any parameter $\theta \in \mathcal{P}_i$, where $\mathcal{P}_i$ corresponds to $\mathcal{A}_i$, the solution to the mp-QP problem (2) can be computed by parametrically solving the equality constrained mp-QP problem

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} z^T Hz \\
\text{subject to} & \quad G_{\mathcal{A}_i} z = b_{\mathcal{A}_i} + S_{\mathcal{A}_i} \theta.
\end{align*}$$

(5)

The parametric solution to (5) is an affine function in the parameter $\theta$, [2]. Now define $q_{i} \in \mathbb{R}^p$ and $Q_{i} \in \mathbb{R}^{p \times n}$ as

$$q_{i} \triangleq \begin{cases} q_{i,\mathcal{N}_i} &= 0, \\ q_{i,\mathcal{A}_i} &= q_{i}, \end{cases}$$

(6)

and

$$Q_{i} \triangleq \begin{cases} Q_{i,\mathcal{N}_i} &= 0, \\ Q_{i,\mathcal{A}_i} &= Q_{i}, \end{cases}$$

where $q_{i} \in \mathbb{R}^{n_{\mathcal{A}_i}}$ and $Q_{i} \in \mathbb{R}^{n_{\mathcal{A}_i} \times n}$ are given by

$$q_{i} + Q_{i} \theta = - (G_{\mathcal{A}_i} H^{-1} G_{\mathcal{A}_i}^T)^{-1} (b_{\mathcal{A}_i} + S_{\mathcal{A}_i} \theta).$$

(7)

See, e.g., [2] for further details. The boldface notation, introduced in (6), is used to denote a variable that is related to all the $p$ constraints in the mp-QP problem (2), but with some components trivially equal to zero. A double index as in, e.g., $Q_{i,\mathcal{A}_i}$ denotes the rows in $Q_{i}$ indexed by $\mathcal{A}_i$. The choice of
this notation will later become clear. The parametric primal and dual solution to (5) for \( \theta \in P_i \) is then given by
\[
\lambda^*(\theta) = q_i + Q_i \theta, \tag{8a}
\]
\[
z^*(\theta) = k_i + K_i \theta \triangleq -H^{-1}G_{Ai}^T (q_{i,Ai} + Q_{i,Ai} \theta). \tag{8b}
\]

B. Computing the critical region for an optimal active set

The critical region \( P_i \) is the set of parameters for which the active set \( A_i \) is optimal, i.e., all parameters \( \theta \in \Theta \) such that primal and dual feasibility given by (3b) and (3c), respectively, is retained. By inserting the parametric solution (8) in (3b) and (3c), the critical region \( P_i \) is defined by
\[
P_i = \{ \theta \in \Theta | H_{N_i}^T (k_i + K_i \theta) \leq b_{N_i} + S_{N_i} \theta, \ q_{i,Ai} + Q_{i,Ai} \theta \geq 0 \}, \tag{9}
\]
where the primal and dual feasibility conditions are given by the first and second inequalities in (9), respectively. \( P_i \) is a polyhedron with \( p \) hyperplanes, [2]. All hyperplanes in (9) which are redundant can be removed to obtain a description of \( P_i \), consisting of the minimal number of describing hyperplanes, [2], [10]. Let \( H_{i}^T \subseteq N_i \) and \( H_{i}^T \subseteq A_i \) be the indices of the describing hyperplanes related to the primal and dual feasibility conditions in (9), respectively. Then the critical region \( P_i \) in (9) is equivalent to
\[
P_i = \{ \theta \in \Theta | H_{N_i}^T (k_i + K_i \theta) \leq b_{N_i} + S_{N_i} \theta, \ q_{i,N_i} + Q_{i,N_i} \theta \geq 0 \}. \tag{10}
\]

Storing the parametric solution and the minimal description of the critical regions requires \( \mathbb{M} \) real numbers, where
\[
\mathbb{M} = \mathcal{R} m (n+1) + \mathcal{M}^T, \quad \mathcal{M}^T = \sum_{i=1}^{R} (H_{i}^T + H_{i}^T) (n+1). \tag{11}
\]

III. LOW RANK CHANGES OF PARAMETRIC SOLUTION

In this section it will be shown how the parametric solution in a neighboring region can efficiently be described by small, structured, modifications of the solution in the first region. Stepping over a facet between two neighboring regions corresponds to adding or removing constraints to the optimal active set, [10]. Hence, an equivalent interpretation is that the parametric solution for one set of optimal active constraints can be used to describe the solution for an optimal active set where constraints have been added to, or removed from, the first one. For notational convenience the case when only one constraint is added or removed is presented in this paper. The case for \( k \) constraints can be shown analogously.

A. Adding one constraint to the optimal active set

Let the solution corresponding to an optimal active set \( A_i \subseteq A \) be given by (8). Consider the case when a constraint \( l \in N_i \) is added such that the active set \( A_j = A_i \cup l \) is also optimal, and hence corresponds to the critical region \( P_j \).

**Theorem 1:** Let the parametric solution of (2) for the optimal active set \( A_i \) be given by (8). Then, the solution for an optimal \( A_j = A_i \cup l \) with \( l \in N_i \) is given by
\[
\lambda^*(\theta) = q_i + Q_i \theta \triangleq q_i + Q_i \theta - d_j (c_j + T_j \theta), \tag{12a}
\]
\[
z^*(\theta) = k_j + K_j \theta \triangleq k_i + K_i \theta + f_j (c_j + T_j \theta), \tag{12b}
\]
where \( c_j \in \mathbb{R}, v_j \in \mathbb{R}^n, f_j \in \mathbb{R}^m \) and \( d_j \in \mathbb{R}^p \) with \( d_{j,N_j} = 0, d_{j,l} = -1 \) and \( d_{j,A_i} \) possibly non-zero.

**Proof:** For \( A_j \), the solution to the mp-qp problem (2) is given by (8), but with index \( j \) instead of \( i \). Without loss of generality, let \( G_L \) be the last row in \( G_{A_i} \). To compute (7), the matrix inversion lemma (38) in App. A is applied to
\[
\begin{bmatrix}
W & w \\
W^T & \end{bmatrix} \triangleq G_{A_i} H^{-1} G_{A_j}^T = \begin{bmatrix}
G_{A_i} H^{-1} G_{A_i}^T & G_{A_i} H^{-1} G_{A_j}^T \\
G_{A_j} H^{-1} G_{A_i}^T & G_{A_j} H^{-1} G_{A_j}^T
\end{bmatrix}. \tag{13}
\]
The dual parametric solution (8a) for \( A_j \) is then given by
\[
\lambda_{A_j}^*(\theta) = \begin{bmatrix}
-W^{-1}(b_{A_i} + S_{A_i} \theta) \\
0
\end{bmatrix} - \begin{bmatrix}
-W^{-1}(c_j + T_j \theta) \\
-(c_j + T_j \theta)
\end{bmatrix}, \tag{14}
\]
where \( c_j \) and \( v_j \) are defined as (with \( C \) defined as in (38))
\[
c_j \triangleq C^{-1} (w^T \begin{bmatrix}
\theta \\
\theta
\end{bmatrix} - b_i) \in \mathbb{R}, \tag{15a}
\]
\[
v_j \triangleq C^{-1} (\begin{bmatrix}
S_{A_i}^T w - S_{A_i}^T \theta
\end{bmatrix} + \begin{bmatrix}
\theta
\end{bmatrix}). \tag{15b}
\]
Furthermore, from (7) and the definition of \( W \) in (13), it is clear that \( -W^{-1}(b_{A_i} + S_{A_i} \theta) = q_i + Q_i \theta \). Hence, by defining \( d_{j,A_i} \triangleq W^{-1}(b_i) \in \mathbb{R}^{A_i}, d_{j,l} \triangleq -1 \) and \( d_{j,N_j} \triangleq 0 \), the dual solution can be written compactly as (12a).

From (8b) it can be seen that the primal solution for the optimal active set \( A_j \) is \( z^*(\theta) = -H^{-1} G_{A_j}^T \lambda_{A_j}^*(\theta) \), giving
\[
z^*(\theta) = -H^{-1} G_{A_j}^T (q_{i,A_i} + Q_{i,A_i} \theta - d_{j,A_i} (c_j + T_j \theta)) = -H^{-1} G_{A_j}^T (q_{i,A_i} + Q_{i,A_i} \theta + H^{-1} G_{A_j}^T d_{j,A_i} (c_j + T_j \theta)). \tag{16}
\]
In the second equality, \( d_{i,l} = 0 \) and \( Q_{i,l} = 0 \) by definition are used. Using \( -H^{-1} G_{A_j}^T (q_{i,A_i} + Q_{i,A_i} \theta) = k_i + K_i \theta \) from (8b) and defining \( f_j \triangleq H^{-1} G_{A_j}^T d_{j,A_i} \in \mathbb{R}^m \), the primal solution (12b) is obtained from (16).

**Remark 1:** From Thm. 1 it can be seen that the parametric solution in \( P_j \) can be computed as a rank one modification of the parametric solution in \( P_i \).

**Corollary 1:** Let the parametric solution to (2) for an optimal active set \( A_j = A_i \cup l \) be given by Thm. 1. Then the corresponding critical region \( P_j \) is given by (9), but where the primal and dual feasibility conditions are instead
\[
G_{N_j} (k_i + K_i \theta) + f_j, N_j (c_j + T_j \theta) \leq b_{N_j} + S_{N_j} \theta, \tag{17a}
\]
\[
q_{i,A_i} + Q_{i,A_i} \theta - d_{j,A_i} (c_j + T_j \theta) \geq 0, \tag{17b}
\]
where \( f_j \triangleq G_{f_j} \in \mathbb{R}^p \) with \( f_j,A_i = 0 \).

**Proof:** The critical region \( P_j \) is given by (9) but with index \( j \) instead of \( i \). Hence, the dual feasibility conditions (17b) for \( P_j \) follow directly from the dual solution (12a).

By inserting the primal solution (12b) into the inequality constraints of (2) and defining \( f_j \triangleq G_{f_j} \) gives
\[
G_{A_j} (k_i + K_i \theta) + G_{f_j} (c_j + T_j \theta) \leq b_i + S_{A} \theta \iff G_{N_j} (k_j + K_i \theta) + f_j, N_j (c_j + T_j \theta) \leq b_{N_j} + S_{N_j} \theta. \tag{18}
\]
Here it is used that \( G_{A_i} (k_i + K_i \theta) + G_{A_j} f_j (c_j + T_j \theta) = G_{A_j} (k_i + K_j \theta) = b_{A_j} + S_{A_j} \theta \) by the definition of \( A_j \).

\[ f_j, A_i = 0 \] follows from the definition of \( f_j, d_j, W \) and \( w \).
Remark 2: $G_{N_1}(k_i + K_0 \theta) \preceq b_{N_1} + S_{N_1} \theta$ are a subset of the primal feasibility conditions in $P_i$, and $q_{i,l} = 0$ and $Q_{i,l} = 0$. Hence, the new information in the description of $P_i$ is contained in $f_{i,N_1} \in R^{|N_1|}$ and $d_{j,A_i} \in R^{|A_i|}$. Note that $c_j$ and $v_j$ are already computed for $\lambda^*(\theta)$ and $z^*(\theta)$.

B. Removing one constraint from the optimal active set

When a constraint is removed from the optimal active set, the parametric solution and the description of the critical region change in a similar way as in Sec. III-A.

Theorem 2: Let the solution for the optimal active set $A_i$ be given by (8) but with index $j$ instead of $i$. Then, the parametric solution for the optimal active set $A_i = A_j \setminus l$ with $l \in A_j$ is given by

$$
\lambda^*(\theta) = q_i + Q_i \theta \triangleq q_j + Q_j \theta - d_i (c_j + v_i^T \theta),
$$

$$(19a)
$$

$$
z^*(\theta) = k_i + K_0 \theta \triangleq k_j + K_0 \theta + f_i (c_j + v_i^T \theta),
$$

$$(19b)
$$

where $c_j \triangleq c_j$, $v_i \triangleq v_j$, $d_i \triangleq -d_j$ and $f_i \triangleq -f_j$. Here the variables with index $j$ are computed as in Thm. 1.

Proof: First, note that $A_i = A_j \setminus l \iff A_j = A_i \cup l$. Hence, Thm. 1 applies for the optimal active set $A_j$, and by re-ordering the terms in (12a) it can be seen that

$$
q_i + Q_i \theta = q_j + Q_j \theta + d_j (c_j + v_j^T \theta),
$$

$$(20)
$$

which, by defining the variables $c_i \triangleq c_j$, $v_i \triangleq v_j$ and $d_i \triangleq -d_j$ gives (19a). Furthermore, by re-ordering the terms in (12b) it follows that

$$
k_i + K_0 \theta = k_j + K_0 \theta - f_i (c_j + v_j^T \theta),
$$

$$(21)
$$

which, using $c_i$, $v_i$ and defining $f_i \triangleq -f_j$, gives (19b).

Remark 3: The parametric solution in $P_i$ is a rank one modification of the parametric solution in $P_j$.

Corollary 2: Let the parametric solution to (2) with $A_i = A_j \setminus l$ be given by Thm. 2. Then the corresponding critical region $P_i$ is given by (9), but where the primal and dual feasibility conditions are instead given by

$$
G_{N_i}(k_j + K_0 \theta) + f_{i,N_1} (c_i + v_i^T \theta) \preceq b_{N_i} + S_{N_i} \theta,
$$

$$(22a)
$$

$$
q_{i,A_i} + Q_{j,A_i} \theta - d_{i,A_i} (c_i + v_i^T \theta) \succeq 0,
$$

$$(22b)
$$

where $f_i \triangleq G f_i \in R^p$ with $f_{i,A_i} = 0$.

Proof: The dual feasibility conditions (22b) follow directly from (19a). Furthermore, inserting the primal parametric solution (19b) into the inequality constraints of the mp-QP problem (2) gives

$$
G_{N_i}(k_j + K_0 \theta) + G_{N_i} f_i (c_i + v_i^T \theta) \preceq b_{N_i} + S_{N_i} \theta.
$$

$$(23)
$$

By using the definition $f_i \triangleq G f_i$, (23) gives (22a). $f_{i,A_i} = 0$ follows by definition, which concludes the proof.

Remark 4: Similar to Rem. 2, the description of $P_j$ is reused in $P_i$. Furthermore, $G_i (k_j + K_0 \theta) = b_i + S_i \theta$ since $l \in A_j$. Hence, the new information in the description of $P_i$ is contained in $f_{i,N_1} \in R^{|N_1|}$ and $d_{i,A_i} \in R^{|A_i|}$.

IV. MEMORY EFFICIENT STORAGE TREE

In this section it will be shown how the theory presented in Sec. III can be utilized repeatedly to store the parametric solution and critical regions in a memory efficient manner. The storage of the parametric solution is arranged into a tree structure, henceforth denoted as the storage tree. The tree structure is related to the tree in [11] and the graph in [16].

Remark 5: The results in Sec. III are not dependent on the tree structure and hold also for more general graph structures.

The set of all optimal active sets $A$ can be arranged in a tree structure by choosing the root node $r \in Z_i$ to correspond to $A_i \in A$. To simplify the notation, let $p(i)$, $ch(i)$ and $anc(i)$ denote the parent, the set of children and the ordered set of ancestors of node $i$ in the tree, respectively. Furthermore, let desc$(i)$ denote the descendants of node $i$, $A \cup (\text{desc}(i))$ be the set of all nodes in the tree and the ancestor nodes except the root, and let $D$ be the maximum depth in the tree.

Definition 1: A storage tree of a set of optimal active sets $A$ is denoted $T(A, r)$, where node $r$ is the root node.

Assumption 1: For all nodes $i \in T(A, r)$ only one constraint is added to, or removed from, the optimal active set of the parent node $p(i)$.

The nodes in a storage tree from Def. 1 correspond to the optimal active sets $A_i \in A$, and when Ass. 1 holds, they are arranged such that either one constraint is added or removed in the optimal active set from a parent to a child. Moving between a parent and one of its children hence corresponds to stepping over a facet between adjacent critical regions [10].

Remark 6: For the case when $k$ constraints are added or removed, Ass. 1 does not hold. However, this case can be derived analogously but is omitted here for notational brevity.

An example with two parameters and a partitioning consisting of 6 critical regions $P_i$ for $i \in Z_{1,6}$ is seen in Fig. 1, and a corresponding storage tree $T(A, 2)$ is seen in Fig. 2. Each critical region $P_i$ corresponds to an optimal active set $A_i \in A$, where $A = \{\{\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}\}$. Hence, $P_2$ is the critical region for the optimal active set $A_2 = \{1\}$ etc. In Fig. 1, the hyperplane between, e.g., $P_2$ and $P_6$ corresponds to constraint 3. Hence, moving from region $P_2$ to $P_6$ by stepping over the shared facet corresponds to adding constraint 3 to the optimal active set in $P_2$, i.e., $A_6 = A_2 \cup \{3\}$. In the tree in Fig. 2 the optimal active set $A_2 = \{1\}$ is chosen as root node, and for this tree it can be seen that, for example, $p(3) = 1$, $\text{anc}(3) = \{1, 2\}$, $P(3) = \{3, 1\}$ and $\text{ch}(2) = \{1, 5, 6\}$. The maximum depth is $D = 2$. The transition from $P_2$ to $P_6$ corresponds to moving from node 2 to node 6 in the tree by adding constraint 3.

Note that the storage tree is not unique. Here, e.g., $T(A, 1)$ could also be used, i.e., having the optimal active set $\{\}$ in the root instead. The choice of tree structure will affect the maximum depth of the tree, and hence also the online performance. How to choose the tree to obtain maximum performance is outside the scope of this work.

Remark 7: The storage tree could either be constructed after all optimal active sets have been determined, or while building the solution to the mp-QP problem in the solver.
Furthermore, the critical region corresponds to constraints depending on the type of edge between pa nodes. When moving between adjacent critical regions, the constraint corresponding to the facet is either added or removed from the optimal active set. An example of a storage tree for this partition is seen in Fig. 2.

Fig. 1. An example with critical regions \( \mathcal{P}_i \) for \( i \in \mathbb{Z}_{1,6} \) in two dimensions. The three separating hyperplanes correspond to constraints 1, 2, 3, respectively. When moving between adjacent critical regions, the constraint corresponding to the facet is either added or removed from the optimal active set. An example of a storage tree for this partition is seen in Fig. 2.

Fig. 2. An example of a storage tree for the partitioning in Fig. 1, where each \( \mathcal{P}_i \) for \( i \in \mathbb{Z}_{1,6} \) corresponds to an optimal active set in \( \mathcal{A} = \{\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\} \). The number in front of the colon in each node corresponds to the index \( i \) in \( \mathcal{A} \). A plus sign at an edge corresponds to adding a constraint to the child, and a minus sign corresponds to removing a constraint. Note that the tree is not unique.

### A. Computing the parametric solution and critical region

By storing the full description of the critical region \( \mathcal{P}_r \) and the parametric solution \( k_r \) and \( K_r \) in the root node and repeating the theory in Sec. III, it can be shown that the parametric solution and description of the critical region in each node \( i \) can be described by a number of low rank modifications of the solution in the root node. From the definition of \( \mathcal{T(\mathcal{A}, r)} \) it follows that the modifications which are used to obtain the solution and critical region in node \( i \) are stored in the nodes \( j \in P(i) \), i.e., the nodes along the path from the root node to node \( i \).

**Theorem 3:** Let \( \mathcal{A} \) be the set of optimal active sets for the mp-QP problem (2), and let \( \mathcal{T(\mathcal{A}, r)} \) be a storage tree for which Ass. 1 holds. Then the parametric solution for \( \theta \in \mathcal{P}_i \) for \( i \in T(\mathcal{A}, r) \) is given by

\[
z^*(\theta) = k_r + K_r \theta + \sum_{j \in \mathcal{P}(i)} f_j (c_j + v_j^T \theta),
\]

where \( c_j, v_j \) and \( f_j \) are defined as in Thm. 1 or Thm. 2 depending on the type of edge between \( \text{pa}(i) \) and node \( i \). Furthermore, the critical region \( \mathcal{P}_i \) is described by the set of parameters \( \theta \in \Theta \) that satisfy the inequalities

\[
b_{\mathcal{N}_i} + A_{\mathcal{N}_i} \theta + \sum_{j \in \mathcal{P}(i)} f_j A_{\mathcal{N}_i} (c_j + v_j^T \theta) \preceq 0, \tag{25a}
\]

\[
-q_{r,\mathcal{N}_i} - Q_{r,\mathcal{N}_i} \theta + \sum_{j \in \mathcal{P}(i)} d_j A_{\mathcal{N}_i} (c_j + v_j^T \theta) \preceq 0, \tag{25b}
\]

where \( b \triangleq Gk_r - b \) and \( A \triangleq G K_r - S \).

**Proof:** Assume that (24) holds for some \( n \in \mathcal{T(\mathcal{A}, r)} \) such that \( \text{ch}(n) \neq \emptyset \). Take an arbitrary node \( i \in \text{ch}(n) \) where \( \mathcal{A}_i = \mathcal{A}_n \cup l \) (or \( \mathcal{A}_i = \mathcal{A}_n \setminus l \)). Then it follows from Thm. 1 (or Thm. 2) that the parametric solution is

\[
z^*(\theta) = k_r + K_r \theta + \sum_{j \in \mathcal{P}(n)} f_j (c_j + v_j^T \theta) + \sum_{j \in \mathcal{P}(i)} f_j (c_j + v_j^T \theta).
\]

Here it has been used that \( P(i) = i \cup P(n) \) since \( i \in \text{ch}(n) \). Since (24) holds for the root \( r \), and \( n \) and \( i \in \text{ch}(n) \) were chosen arbitrarily, the relation (24) follows from induction.

The relations (25) can be shown analogously from Cor. 1 (or Cor. 2) by utilizing that, for the root node \( r \), the dual feasibility conditions (3c) are \(-q_r - Q_r \theta \preceq 0 \) and the primal feasibility conditions (3b) are described by

\[
G (k_r + K_r \theta) \preceq b + S \theta \iff b + A \theta \preceq 0, \tag{27}
\]

**Remark 8:** The full parametric solution and description of the critical region is only stored for the root node \( r \). For the rest of the nodes, only the low rank modifications are stored.

The method to compute the solution as in Thm. 3 is implemented in Alg. 1, where the parameter \( \theta \) and the node \( i \) are inputs, and the optimal solution \( z^*(\theta) \) is returned.

**Algorithm 1** On-line evaluation of parametric solution

1: \textbf{input} \( \theta \) and \( i \).
2: Initialize \( z = k_r + K_r \theta \).
3: for \( j \in P(i) \) do
4: \( z := z + f_j (c_j + v_j^T \theta) \).
5: end for
6: \( z^* := z \).

**Corollary 3:** To obtain the minimal representation of the critical region \( \mathcal{P}_i \), only the describing hyperplanes \( \mathcal{H}^+_{r} \) and \( \mathcal{H}^-_{i} \) need to be used, i.e., \( \mathcal{P}_i \) can be described by (10) but where the primal and dual feasibility conditions are given by

\[
b_{\mathcal{H}^+_{r}} + A_{\mathcal{H}^+_{r}} \theta + \sum_{j \in \mathcal{P}(i)} f_j A_{\mathcal{H}^+_{r}} (c_j + v_j^T \theta) \preceq 0, \tag{28a}
\]

\[
-q_{r,\mathcal{H}^+_{r}} - Q_{r,\mathcal{H}^+_{r}} \theta + \sum_{j \in \mathcal{P}(i)} d_j A_{\mathcal{H}^+_{r}} (c_j + v_j^T \theta) \preceq 0. \tag{28b}
\]

**Proof:** The corollary follows directly from Thm. 3 since each hyperplane \( n \) in (25) is described only by the components and rows indexed by \( n \) in (25).

In Alg. 2 the evaluation of a hyperplane using Cor. 3 is implemented. The parameter \( \theta \), the node \( i \) and the hyperplane
Algorithm 2 On-line evaluation of a hyperplane

1: input $\theta$, $i$ and $n$.
2: if $n \in \mathcal{H}_{i}^{P}$ then
3: Initialize $s := b_{n}+A_{n}\theta$.
4: for $j \in P(i)$ do
5: $s := s + f_{j,n}(c_{j}+v_{j}^{T}\theta)$.
6: end for
7: else if $n \in \mathcal{H}_{i}^{R}$ then
8: Initialize $s := -q_{r,n} - Q_{r,n}\theta$.
9: for $j \in P(i)$ do
10: $s := s + d_{j,n}(c_{j}+v_{j}^{T}\theta)$.
11: end for
12: end if

parameter $\theta$ the value of $c_{j}+v_{j}^{T}\theta$ only has to be computed once for each $P_{j}$ when evaluating hyperplanes, and $b_{n}+A_{n}\theta$ and $-q_{r,n} - Q_{r,n}\theta$ only need to be computed once for each parameter and can be re-used by all nodes.

Remark 9: In algs. 1 and 2 the order of summation can be chosen to facilitate on-line performance by taking, e.g., memory access into consideration.

B. Storing the parametric solution and critical region

From Thm. 3 it follows that computing the parametric solution in node $i$ only requires the storage of $c_{j}+v_{j}^{T}\theta$ for each $j \in P(i)$. However, in the root node $r$ the full parametric solution defined by $k_{r} \in \mathbb{R}^{m}$ and $K_{r} \in \mathbb{R}^{m \times n}$ needs to be stored. In Cor. 3 it is shown that the description of the critical region in node $i$ only requires the describing hyperplanes. Hence, the full vectors and matrices in (28) do not need to be stored, but only the components and rows of $b$, $q_{r}$, $A$, $Q_{r}$, $f_{j}$ and $d_{j}$ for $j \in P(i)$ that correspond to the describing hyperplanes.

Furthermore, from Cor. 3 it can be seen that the low rank modification in node $i$ is also used by all descendants desc($i$) $\subseteq T(A,r)$. Hence, the components and rows with indices corresponding to hyperplanes for $P_{j}$ with $j \in $desc($i$) also need to be stored in node $i$. Let $S_{i}^{p}$ and $S_{i}^{d}$ be the indices of the hyperplanes corresponding to primal and dual feasibility conditions in the description of $P_{i}$ that need to be stored in node $i$. Then these sets of indices are

$$S_{i}^{p} \triangleq \bigcup_{j \in $desc($i$) \cup i} \mathcal{H}_{j}^{p}, \quad S_{i}^{d} \triangleq \bigcup_{j \in $desc($i$) \cup i} \mathcal{H}_{j}^{d}. \quad (29)$$

Here the trivial zeros in $f_{i}$ and $d_{i}$ (by definition) should not be taken into consideration when computing $S_{i}^{p}$ and $S_{i}^{d}$ in (29). In each node $i \in T(A,r)$, $c_{i} \in \mathbb{R}$, $v_{i} \in \mathbb{R}^{n}$, $f_{i} \in \mathbb{R}^{m}$, $f_{i}^{S_{p}} \in \mathbb{R}^{S_{i}^{p}}$ and $d_{i}^{S_{d}} \in \mathbb{R}^{S_{i}^{d}}$ need to be stored. Storing them for node $i$ requires $1 + n + m + |S_{i}^{p}| + |S_{i}^{d}|$ real numbers. For the root node $r$, the vectors $k_{r} \in \mathbb{R}^{m}$, $b_{S_{p}} \in \mathbb{R}^{S_{i}^{p}}$, $q_{r} \in \mathbb{R}^{S_{i}^{p}}$ and $K_{r} \in \mathbb{R}^{m \times n}$, $A_{S_{d}} \in \mathbb{R}^{S_{i}^{d} \times n}$ and $Q_{r} \in \mathbb{R}^{S_{i}^{d} \times n}$ need to be stored. This requires $m(n+1)+(|S_{i}^{p}|+|S_{i}^{d}|)(n+1)$ real numbers to be stored. Hence, the total number $M_{LR}$ of stored real numbers for $T(A,r)$ is

$$M_{LR} = m(n+R) + M_{LR}^{p}, \quad M_{LR}^{p} \triangleq \left(|S_{i}^{p}| + |S_{i}^{d}|\right)(n+1) + (R-1)(1+n) + \sum_{i \in T(A,r) \cup r} \left(|S_{i}^{p}| + |S_{i}^{d}|\right). \quad (30)$$

Note that the storage of $c_{i}$ and $v_{i}$ is included in $M_{LR}^{p}$.

V. EXPLICIT MODEL PREDICTIVE CONTROL

In linear MPC, the input is computed by solving a CFTOC problem. A common formulation of the CFTOC problem is

$$\begin{align*}
\min_{x,u} & \quad \frac{1}{2} \sum_{t=0}^{N-1} (x_{t}^{T}Q_{x}x_{t} + u_{t}^{T}Q_{u}u_{t}) + \frac{1}{2}x_{N}^{T}P_{N}x_{N} \\
\text{subject to} & \quad x_{0} = \bar{x} \\
& \quad x_{t+1} = Ax_{t} + Bu_{t}, \quad t \in \mathbb{Z}_{0,N-1} \\
& \quad H_{x}x_{t} + Hu_{t} + h \leq 0, \quad t \in \mathbb{Z}_{0,N-1} \\
& \quad H_{x}x_{N} + h \leq 0,
\end{align*} \quad (31)$$

where $x \in \mathbb{R}^{n_{x}}$ is the state vector, $u_{t} \in \mathbb{R}^{n_{u}}$ is the control input, $\bar{x}$ is the initial state and $N$ is the prediction horizon. The cost function is given by $Q_{x} \in \mathbb{R}^{n_{x} \times n_{x}}$, $Q_{u} \in \mathbb{R}^{n_{u} \times n_{u}}$ and $P_{N} \in \mathbb{R}^{n_{x} \times n_{x}}$. The equality constraints are the dynamics equations of the controlled system, and the inequality constraints are constraints on the states and control inputs.

Similarly to what is shown in [2], the CFTOC problem (31) can equivalently be written in the form of an mp-QP problem (1) by defining the matrices

$$H \triangleq Q_{u} + B^{T}Q_{B}B, \quad g \triangleq A^{T}Q_{u}B, \quad (32a)$$

$$G \triangleq H_{x}B + Hu, \quad h \triangleq -h, \quad E \triangleq -H_{A}, \quad \theta \triangleq \bar{x}, \quad (32b)$$

where $Q_{u}$, $Q_{B}$, $A$, $B$, $H_{x}$, $H_{u}$ and $h$ are all defined in (39) in App. A. By re-writing the mp-QP problem (1) into (2) and solving it parametrically, the optimal solution to the CFTOC problem is given by $u^{*}(\theta) = z^{*}(\theta) - H^{-1}g^{T}\theta$ where $z^{*}(\theta)$ is given by the PWA function (4), [2].

Since the CFTOC problem is equivalent to an mp-QP problem in the form (2), the storage of the explicit solution of (31) can be done using the theory presented in this paper. However, only the first control input $u_{0}^{*}$ is used as input to the plant in the MPC control loop [3], and hence the full parametric solution does not need to be stored in the case of explicit MPC. Here, instead only the first $n_{u}$ rows of $k_{i}$ and $K_{i}$, respectively, are stored. For the traditional, non-compressed, solution this results in that

$$M_{LR}^{mpc} = Rn_{u}(n+1) \sum_{i=1}^{R} \left(\mathcal{H}_{i}^{p} + \mathcal{H}_{i}^{d}\right)(n+1), \quad (33)$$

real numbers are stored. This is a slightly modified version of $M_{LR}$ in (11). Similarly for the compressed solution, only the low rank modifications affecting the first $n_{u}$ rows need to be stored in $T(A,r)$. Hence, for the explicit MPC solution, the number of stored real numbers is

$$M_{LR}^{mpc} = n_{u}(n+R) + M_{LR}^{p}, \quad (34)$$

which is a slightly modified version of $M_{LR}$ in (30).
VI. EXPERIMENTAL EVALUATION

In this section, the memory requirement for storing the solution and the critical regions for the proposed method is compared to storing the full solution and critical regions. The mp-QP problems have been solved using MPT (version: 3.1.2 (R2011a) 28.10.2015) in MATLAB (version: 8.4.0.150421 (R2014b)). Although the optimal active sets are computed in the mp-QP solver, it was not possible to access them in the solution returned by the solver. Hence, they have instead been retrieved by solving the corresponding QP problem with the Chebychev center of each critical region as parameter.

A. Defining the problems

The comparison has been made for three different examples where explicit MPC controllers are applied to stable LTI systems. In the first example, referred to as Problem 1, the continuous system $1/(s+1)^{n_x}$ which is used in [7] is studied. This system has $n_x$ states and $n_u = 1$ control inputs. The transfer function has been discretized using a unit sampling time, the weight matrices are $Q_x = I$ and $Q_u = 1$, and the terminal cost $P_N$ is chosen as the discrete time LQ cost. The states and control inputs are subject to the constraints

$$-10 \leq x_t \leq 10, \quad t \in \mathbb{Z}_{0,N}, \quad -1 \leq u_t \leq 1, \quad t \in \mathbb{Z}_{0,N-1}. \quad (35)$$

After re-writing this explicit MPC problem into an equivalent mp-QP problem as in Section V, the problem has $n = n_x$ parameters and $m = Nn_u = N$ variables.

Both the second and third problems use a system which is similar to the one used in, e.g., [18], [19]. It consists of $n_M$ unit masses which are coupled with springs and dampers. The spring constants are chosen as 1, the damping constants as 0, the weight matrices to $Q_x = 100 \cdot I$ and $Q_u = I$ and the terminal cost $P_N$ is chosen as the discrete time LQ cost. The continuous system is discretized using the sampling time 0.5 seconds. Two different cases have been studied, referred to as Problem 2 and Problem 3. In Problem 2, the control input is a force acting between terra firma and the first mass, and in Problem 3 there is also an extra control input acting as a force applied between the first two masses. In both problems, the states and control inputs are subject to the constraints

$$-4 \leq x_t \leq 4, \quad t \in \mathbb{Z}_{0,N}, \quad -0.5 \leq u_t \leq 0.5, \quad t \in \mathbb{Z}_{0,N-1}. \quad (36)$$

Each mass introduces 2 states, and $n_u = 1$ in Problem 2 and $n_u = 2$ in Problem 3 by construction. Hence, by re-writing the MPC problem into the equivalent mp-QP problem as in Section V, the corresponding mp-QP problem has $n = 2n_M$ parameters and $m = Nn_u$ variables.

B. Experimental results

The relative memory reduction has been computed for the three problems for different parameter dimensions and prediction horizons, and the results are summarized in tables I-III. The storage tree $T(A, r)$ is chosen such that the root node corresponds to the optimal active set $A_r = \emptyset$, i.e., the unconstrained minimum, and whenever it is possible only one constraint is added to each child of the nodes. When the mp-QP problem is defined, all redundant constraints are removed, giving $p$ number of non-redundant constraints. Furthermore, $R$ is the number of regions, $D$ is the maximum depth of $T(A, r)$, and

$$\Delta^{\text{cr}} \triangleq \frac{M_{\text{cr}}^F}{M_F}, \quad \Delta \triangleq \frac{M_{\text{L}}^F}{M_F}, \quad \Delta^{\text{mpc}} \triangleq \frac{M_{\text{mpc}}^F}{M_F}, \quad (37)$$

are the relative reductions in the number of stored real numbers for only the critical regions, the full solution and the critical regions, and for storing the first $n_u$ control inputs and the critical regions, respectively. Hence, for the case of explicit MPC, $\Delta^{\text{mpc}}$ determines the total relative memory reduction, whereas for a full mp-QP problem it is given by $\Delta$. No symmetry or other properties which are inherited from the explicit MPC problems are exploited in the comparison. In some cases, for large problems with many parameters and variables, the solution in a few regions are numerically bad. This is probably a consequence of difficulties with finding the correct optimal active set given the critical region.

In Table I the result for Problem 1 is seen, and it is clear that the relative memory reduction becomes increasingly beneficial with the parameter dimension. For $N = 2$ and $N = 3$, the memory is reduced by approximately an order of magnitude for the parameter dimensions $n = 12$ and $n = 14$. The result for Problem 2 is presented in Table II, and for this problem the relative memory reduction is an order of magnitude for problems with $n_M \geq 6$. Table III contains the numerical results for Problem 3, and it can be seen that also for this problem the relative memory reduction is increased for larger parameter dimensions. Note that for all evaluated problem dimensions the memory required for the storage tree is lower than for storing the full solution.

<table>
<thead>
<tr>
<th>$n/N$</th>
<th>$p$</th>
<th>$R$</th>
<th>$D$</th>
<th>$\Delta^{\text{cr}}$</th>
<th>$\Delta$</th>
<th>$\Delta^{\text{mpc}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/2</td>
<td>10</td>
<td>5</td>
<td>2</td>
<td>0.909</td>
<td>0.729</td>
<td>0.827</td>
</tr>
<tr>
<td>4/2</td>
<td>20</td>
<td>11</td>
<td>2</td>
<td>0.446</td>
<td>0.403</td>
<td>0.431</td>
</tr>
<tr>
<td>6/2</td>
<td>28</td>
<td>45</td>
<td>2</td>
<td>0.258</td>
<td>0.239</td>
<td>0.252</td>
</tr>
<tr>
<td>8/2</td>
<td>44</td>
<td>153</td>
<td>2</td>
<td>0.176</td>
<td>0.167</td>
<td>0.173</td>
</tr>
<tr>
<td>10/2</td>
<td>52</td>
<td>192</td>
<td>2</td>
<td>0.130</td>
<td>0.126</td>
<td>0.129</td>
</tr>
<tr>
<td>12/2</td>
<td>66</td>
<td>255</td>
<td>2</td>
<td>0.107</td>
<td>0.105</td>
<td>0.107</td>
</tr>
<tr>
<td>14/2</td>
<td>80</td>
<td>336</td>
<td>2</td>
<td>0.090</td>
<td>0.088</td>
<td>0.090</td>
</tr>
<tr>
<td>2/3</td>
<td>12</td>
<td>5</td>
<td>2</td>
<td>0.909</td>
<td>0.694</td>
<td>0.827</td>
</tr>
<tr>
<td>4/3</td>
<td>24</td>
<td>13</td>
<td>3</td>
<td>0.476</td>
<td>0.411</td>
<td>0.456</td>
</tr>
<tr>
<td>6/3</td>
<td>36</td>
<td>89</td>
<td>3</td>
<td>0.258</td>
<td>0.234</td>
<td>0.252</td>
</tr>
<tr>
<td>8/3</td>
<td>56</td>
<td>575</td>
<td>3</td>
<td>0.187</td>
<td>0.175</td>
<td>0.184</td>
</tr>
<tr>
<td>10/3</td>
<td>66</td>
<td>1186</td>
<td>3</td>
<td>0.139</td>
<td>0.133</td>
<td>0.138</td>
</tr>
<tr>
<td>12/3</td>
<td>86</td>
<td>1679</td>
<td>3</td>
<td>0.124</td>
<td>0.119</td>
<td>0.123</td>
</tr>
<tr>
<td>14/3</td>
<td>102</td>
<td>2664</td>
<td>3</td>
<td>0.115</td>
<td>0.111</td>
<td>0.114</td>
</tr>
<tr>
<td>2/4</td>
<td>14</td>
<td>5</td>
<td>2</td>
<td>0.909</td>
<td>0.669</td>
<td>0.827</td>
</tr>
<tr>
<td>4/4</td>
<td>26</td>
<td>13</td>
<td>3</td>
<td>0.476</td>
<td>0.400</td>
<td>0.456</td>
</tr>
<tr>
<td>6/4</td>
<td>42</td>
<td>129</td>
<td>4</td>
<td>0.252</td>
<td>0.226</td>
<td>0.246</td>
</tr>
<tr>
<td>8/4</td>
<td>66</td>
<td>1222</td>
<td>4</td>
<td>0.207</td>
<td>0.189</td>
<td>0.203</td>
</tr>
<tr>
<td>10/4</td>
<td>80</td>
<td>4300</td>
<td>4</td>
<td>0.161</td>
<td>0.151</td>
<td>0.159</td>
</tr>
<tr>
<td>12/4</td>
<td>104</td>
<td>5408</td>
<td>4</td>
<td>0.181</td>
<td>0.172</td>
<td>0.179</td>
</tr>
</tbody>
</table>

VII. CONCLUSIONS AND FUTURE WORK

In this paper, theory and algorithms for reducing the memory footprint when storing parametric solutions to mp-QP problems are introduced. This is performed by exploiting low rank structure in the parametric solution between the neighboring critical regions. This is similar to how low
rank modifications is used as a tool to increase on-line performance in popular state-of-the-art QP methods, but here it is applied to mp-QP problems to reduce the memory required to store the solutions. The proposed method stores the solution in a tree structure and can be implemented in already existing solvers for mp-QP problems, or be considered as a post-processing data compression step. For future work, an extension to other problem classes such as, e.g., multiparametric linear programming will be investigated. Moreover, it will be studied which point location algorithm that benefits most from using the storage tree introduced in this paper. Furthermore, it can be investigated how to exploit low rank modifications of, e.g., Cholesky factorizations to improve the numerical computations of the low rank modifications.

**APPENDIX**

A. Linear algebra and definitions

Consider the symmetric positive definite matrix \( \begin{bmatrix} W & w \\ w^T & w_0 \end{bmatrix} \).

By using the matrix inversion lemma, the inverse is given by

\[
\begin{bmatrix} W & w \\ w^T & w_0 \end{bmatrix}^{-1} = \begin{bmatrix} W^{-1} + W^{-1}wC^{-1}w^TW^{-1} & -W^{-1}wC^{-1} \\ C^{-1}w^TW^{-1} & C^{-1} \end{bmatrix},
\]

where \( C \triangleq w_0 - w^TW^{-1}w \in \mathbb{S}_++ \).

The matrices in the CFTOC problem are defined by

\[
A \triangleq \begin{bmatrix} I \\ A \end{bmatrix}, \quad B \triangleq \begin{bmatrix} 0 \\ AB \\ A^{n-1}B \\ A^{n-2}B \ldots B \end{bmatrix},
\]

\[
Q_x \triangleq \text{diag}(Q_x, \ldots, Q_x, P_N), \quad Q_u \triangleq \text{diag}(Q_u, \ldots, Q_u),
\]

\[
H_x \triangleq \begin{bmatrix} H_x \\ \vdots \\ H_x \end{bmatrix}, \quad H_u \triangleq \begin{bmatrix} H_u \\ \vdots \\ H_u \end{bmatrix}, \quad h \triangleq \begin{bmatrix} h \\ \vdots \\ h \end{bmatrix}.
\]

**REFERENCES**


