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

Accelerating online MPC with partial explicit information and linear storage complexity in the number of constraints

Michael Jost and Martin Mönnigmann

***Abstract*— Model predictive control (MPC) is an acclaimed method for the control of constrained systems. Since a constrained optimization problem has to be solved in every time step, the online computational effort of MPC is high. Explicit MPC provides an analytical solution to the same optimization problem, but explicit MPC is only useful for small systems, since the storage requirements for the explicit control law grow exponentially in the number of constraints of the optimization problem.**

We show that *online* MPC can be accelerated with information on the structure of the control law, where this structural information is calculated *offline* with techniques from explicit MPC. Our two main contributions are as follows: We demonstrate that online MPC can be sped up significantly if only q state space regions, the regions of activity, are stored, where q is the number of constraints. Note that this linear growth in q is obviously very different from the exponential growth in q of the number of polytopes that need to be stored in explicit MPC. Secondly, we claim that the proposed method is a variant of a family of methods, which comprises online MPC and explicit MPC as two limiting cases.

I. Introduction

Explicit model predictive control is usually thought of as a method to compute analytical solutions to MPC problems. On a more abstract level, however, explicit MPC provides insight into the structure of the solution of MPC problems. In particular, the solution to the constrained LQR problem is known to be a piecewise affine function on a partition of the state space into convex polytopes [1], [2]. Some authors have used this insight into the structure of the solution to improve *online* MPC methods. Ferreau and Diehl [3] proposed a homotopy method to detect active set changes to speed up online MPC. Pannocchia and Rawlings [4], [5] proposed to enumerate, or partially enumerate, active sets in online MPC. This approach builds up a frequency distribution of active sets and uses this information to anticipate the new active set when an active set change occurs.

We propose to compute information on inactive and active sets *before* runtime, to store this information in the form of regions of activity (see (11)), and to use the regions of activity in an *online* MPC method to speed up the online optimization step. Essentially, the online optimization problem is simplified by removing constraints that are known to be *inactive* for the current state, and by turning inequality constraints that are known to be *active* for the current state into equality constraints. It is the central aspect of the proposed method that only q state space regions, the regions of activity, need to be found and stored, where q is the number of constraints of the quadratic program (QP) solved in each online MPC step. In contrast, the number of polytopes needed to store the explicit MPC solution grows exponentially in q.

M. Jost and M. Mönnigmann are with Automatic Control and Systems Theory, Department of Mechanical Engineering, Ruhr-Universität Bochum, Germany. E-mail: `michael.s.jost@rub.de`, `martin.moennigmann@rub.de`.

Several variants of the proposed method can be constructed, which differ with respect to their offline computational efforts, their online storage requirements, and the degree of reduction of the online QP size that results. We introduce and investigate two variants that work with q regions as outlined above. Other variants can be considered in which the online storage can be tuned to either a smaller, or larger, number of regions than q. In fact, the explicit MPC solutions can be conceived as a limiting case of the proposed method.

Section II states the problem class. The central idea is introduced in Sect. III, followed by some details on its implementation in Sect. IV. An example is presented in Sect. V. Conclusions and an outlook are stated in Sect. VI.

II. MPC problem statement

Consider a linear discrete-time system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

subject to input and output constraints

$$u_{\min} \le u(t) \le u_{\max},\ y_{\min} \le y(t) \le y_{\max} \text{ for all } t \tag{2}$$

with system matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, states $x(t) \in \mathbb{R}^n$, inputs $u(t) \in \mathbb{R}^m$, outputs $y(t) \in \mathbb{R}^p$, and bounds $u_{\min}$, $u_{\max}$, $y_{\min}$, $y_{\max}$ of appropriate dimensions. We assume the pair (A, B) to be stabilizable. In order to regulate system (1), (2) to the origin by MPC, the following optimal control problem needs to be solved. We state this problem for completeness and to introduce some notation:

$$\begin{aligned} \min_{U,X} \quad & x'_{t+N_y|t} P\, x_{t+N_y|t} + \\ & \sum_{k=0}^{N_y-1} \left(x'_{t+k|t}\, Q\, x_{t+k|t} + u'_{t+k}\, R\, u_{t+k} \right) \\ \text{s. t.} \quad & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k}, \quad k = 0, \dots, N_y - 1, \\ & y_{t+k|t} = Cx_{t+k|t}, \quad k = 1, \dots, N_y, \\ & x_{t|t} = x(t), \\ & u_{t+k} = Kx_{t+k|t}, \quad N_u \le k < N_y, \\ & y_{\min} \le y_{t+k} \le y_{\max}, \quad k = 1, \dots, N_c, \\ & u_{\min} \le u_{t+k} \le u_{\max}, \quad k = 0, \dots, N_c, \end{aligned} \tag{3}$$

where $U' = (u'_t, ..., u'_{t+N_u-1})$, $X' = (x'_{t+1|t}, \ldots, x'_{t+N_y|t})$ are introduced for convenience, $x(t+k|t)$ is the state at time $t+k$ predicted at time t, $Q \in \mathbb{R}^{n\times n}$, $Q \succeq 0$, $R \in \mathbb{R}^{m\times m}$, $R \succ 0$, $P \in \mathbb{R}^{n\times n}$, $P \succeq 0$ are the usual weighting matrices. The pair $(Q^{1/2}, A)$ is assumed to be detectable, $K \in \mathbb{R}^{m\times n}$ is a feedback gain applied to (1) for all $N_u \leq t < N_y$, and N_y, N_c and N_u are the output, constraint and input horizon, respectively. Closed-loop stability has been discussed elsewhere (see [1] and references therein). We choose P and K to be the solution of the unconstrained infinite horizon LQR problem

$$\begin{aligned} P &= Q + A'PA - A'PB(R + B'PB)^{-1}B'PA, \\ K &= (R + B'PB)^{-1}B'PA. \end{aligned} \tag{4}$$

Problem (3) can be solved analytically by transforming it into a parametric quadratic program [1], [2] of the form

$$\min_z \left\{ \frac{1}{2} z'Hz \middle| Gz \leq w + Sx \right\} \tag{5}$$

with

$$z = U + H^{-1}\tilde{F}'x, \tag{6}$$

$n_z = mN_u$, $H \in \mathbb{R}^{n_z\times n_z}$, $w \in \mathbb{R}^q$, $G \in \mathbb{R}^{q\times n_z}$, $S \in \mathbb{R}^{q\times n}$, where q denotes the number of constraints. It can be shown that H is positive definite under the assumptions stated for (3). Consequently, H is invertible, and therefore the transformation (6) exists. Bemporad et al. [1] showed that (5) is solved by a continuous piecewise affine function. More precisely, there exist a finite number $n_{\mathcal{P}}$ of convex polytopes $\mathcal{P}_i$ with pairwise disjoint interiors, gains $\hat{K}_i \in \mathbb{R}^{n_z\times n}$, and biases $\hat{b}_i \in \mathbb{R}^{n_z}$, $i = 1, \ldots, n_{\mathcal{P}}$ such that $\cup_i^{n_{\mathcal{P}}} \mathcal{P}_i = \mathcal{X}$ and

$$z(x) = \begin{cases} \hat{K}_1 x + \hat{b}_1 & \text{if } x \in \mathcal{P}_1, \\ \quad\vdots & \quad\vdots \\ \hat{K}_{n_{\mathcal{P}}} x + \hat{b}_{n_{\mathcal{P}}} & \text{if } x \in \mathcal{P}_{n_{\mathcal{P}}}. \end{cases} \tag{7}$$

This $z(x)$ is the solution to the parametric program (5) in the sense that

$$z(x) = \arg\min_z \left\{ \frac{1}{2} z'Hz \middle| Gz \leq w + Sx \right\}. \tag{8}$$

It is unique for every $x \in \mathcal{X}$, since the optimization problem is strictly convex. Once $z(x)$ has been found, the control law $u : \mathcal{X} \to \mathbb{R}^m$ that solves (3) can be calculated with (6). [1]

A. Notation

For an arbitrary matrix $M \in \mathbb{R}^{a\times b}$, $M^{\mathcal{S},\mathcal{T}}$ with $\mathcal{S} \subseteq \{1, ..., a\}$ and $\mathcal{T} \subseteq \{1, ..., b\}$ denotes a submatrix with the rows indicated by $\mathcal{S}$ and the columns indicated by $\mathcal{T}$. For any set S denote the convex hull of S by $\mathrm{conv}(S)$. Let the q constraints form the index set

$$\mathcal{Q} = \{1, ..., q\}.$$

[1]Note that it is an abuse of notation to denote both the optimization variable z in (5) and the solution to the optimization problem $z(x)$ by the same symbol (also cf. (6)). This notation is used, however, to avoid cumbersome technicalities.

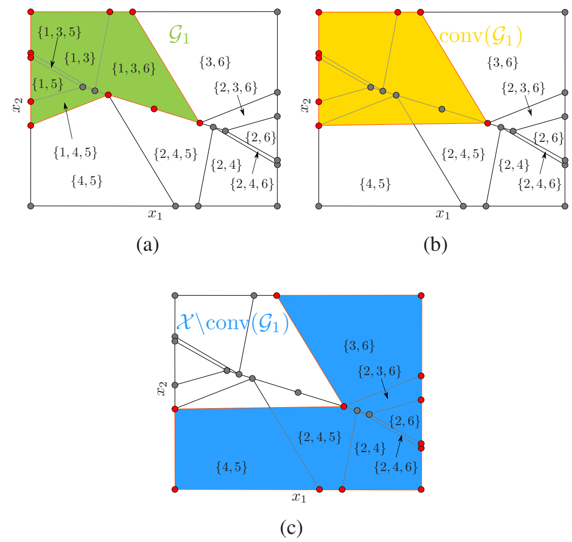


Fig. 1. Illustration of the main idea. Sets $\{1, 3, 5\}$, $\{1, 3\}$ etc. indicate the active set $\mathcal{A}$ of the respective region. The region of activity $\mathcal{G}_1 := \{x \in \mathcal{X} | \text{ constraint 1 active}\}$ (cf. the definition in (11)), is highlighted in (a). $\mathcal{G}_1$ is the union of some of the convex polytopes $\mathcal{P}_j$ by definition, but $\mathcal{G}_1$ itself is not convex. The convex hull of $\mathcal{G}_1$ is highlighted in (b). Since every $\mathcal{G}_i$ is the union of convex polytopes, the convex hull $\mathrm{conv}(\mathcal{G}_i)$ is a polytope (as opposed to a more general convex set than a convex polytope). Part (c) illustrates the implication (14). Since constraint 1 is not active outside of $\mathrm{conv}(\mathcal{G}_1)$, we can infer that for every $x \in \mathcal{X}\backslash\mathrm{conv}(\mathcal{G}_i)$, or equivalently for all $x \notin \mathrm{conv}(\mathcal{G}_1)$, constraint 1 is inactive. (Sample system taken from Spjotvold et al. [9].)

A constraint i is called *active* at x if $G^i z(x) - w^i - S^i x = 0$ and *inactive* if $G^i z(x) - w^i - S^i x < 0$, where $x \in \mathcal{X}$ is an arbitrary current state and $z(x)$ is as in (8). By

$$\begin{aligned} \mathcal{A}(x) &= \left\{ i \in \mathcal{Q} \middle| G^i z(x) - w^i - S^i x = 0 \right\}, \\ \mathcal{I}(x) &= \left\{ i \in \mathcal{Q} \middle| G^i z(x) - w^i - S^i x < 0 \right\}, \end{aligned} \tag{9}$$

denote the sets of active and inactive constraints, respectively, for the current state x. The symbol $\tilde{\mathcal{A}}(x)$ (resp. $\tilde{\mathcal{I}}(x)$) refers to subsets of the active (inactive) constraints, $\tilde{\mathcal{A}}(x) \subseteq \mathcal{A}(x)$ ($\tilde{\mathcal{I}}(x) \subseteq \mathcal{I}(x)$).

We recall that, for every $\mathcal{P}_j$ of the control law (7), the set of active and inactive constraints remains constant in the interior of $\mathcal{P}_j$ [6]–[8]. More specifically, let $\mathcal{P}_j$ be an arbitrary polytope of a control law (7), then $\mathcal{A}(x) = \mathcal{A}(\bar{x})$ (resp. $\mathcal{I}(x) = \mathcal{I}(\bar{x})$) for all $x \in \mathcal{P}_j$ and $\bar{x} \in \mathcal{P}_j$. This implies it is meaningful to introduce the set of active (inactive) constraints $\mathcal{A}_j$ ($\mathcal{I}_j$) of a polytope $\mathcal{P}_j$

$$\begin{aligned} \mathcal{A}_j &= \{i \in \mathcal{Q} | i \in \mathcal{A}(x) \text{ for all } x \in \mathcal{P}_j\}, \\ \mathcal{I}_j &= \{i \in \mathcal{Q} | i \in \mathcal{I}(x) \text{ for all } x \in \mathcal{P}_j\}. \end{aligned} \tag{10}$$

III. Central idea

The central idea can be introduced as follows. For every constraint $i \in \mathcal{Q} = \{1, \ldots, q\}$ define the region of activity

$$\mathcal{G}_i := \{x \in \mathcal{X} | \text{ constraint } i \text{ active}\}. \tag{11}$$

Then, for any $x \in \mathcal{X}$, constraint i is active if and only if $x \in \mathcal{G}_i$ by definition of $\mathcal{G}_i$. In other words

$$x \in \mathcal{G}_i \Leftrightarrow i \in \mathcal{A}(x) \tag{12}$$

for all $i = 1, \ldots, q$. Note that there exist only q regions of activity, where q is the number of constraints in (5). In contrast, the partition needed to define the explicit solution (7) consists of $n_{\mathcal{P}}$ polytopes, and typically $q \ll n_{\mathcal{P}}$. In fact the number of polytopes $n_{\mathcal{P}}$ may grow exponentially in q.

While there generally exist by far fewer regions $\mathcal{G}_i$ than polytopes $\mathcal{P}_j$, the $\mathcal{G}_i$ are in general not convex (see Fig. 1a for an example). Consequently, it is computationally hard to check whether $x \in \mathcal{G}_i$ for some or all i, and it may require too much computation time to test whether $x \in \mathcal{G}_i$ or $x \notin \mathcal{G}_i$ for all $i = 1, \ldots, q$ in an online control algorithm. The computational effort can be reduced, however, by carrying out a weaker test with the convex hulls $\mathrm{conv}(\mathcal{G}_i)$ as follows (see Figs. 1b, c for an illustration). Since $\mathrm{conv}(\mathcal{G}_i) \supseteq \mathcal{G}_i$, we have to replace the equivalence in (12) by the implication $i \in \mathcal{A}(x) \Rightarrow x \in \mathrm{conv}(\mathcal{G}_i)$. It is straightforward to show by contradiction that

$$x \notin \mathrm{conv}(\mathcal{G}_i) \Rightarrow i \in \mathcal{I}(x) \tag{14}$$

holds. We claim without proof that $\mathrm{conv}(\mathcal{G}_i)$ is a polytope (rather than a more general convex set) for every $i \in \mathcal{Q}$, since $\mathcal{G}_i$ is the union of polytopes. Therefore, the computational effort to test whether $x \in \mathrm{conv}(\mathcal{G}_i)$ is of the same type as testing $x \in \mathcal{P}_j$.

Relation (14) can be used to construct a subset $\tilde{\mathcal{I}}(x)$ of the true inactive set $\mathcal{I}(x)$ for the current state x. We stress again that only q tests $x \in \mathrm{conv}(\mathcal{G}_i)$, $i = 1, \ldots, q$ must be carried out to find $\tilde{\mathcal{I}}(x) \subseteq \mathcal{I}(x)$. For the current state x the constraints $i \in \tilde{\mathcal{I}}(x)$ can be removed from the optimization problem (5) when solving it online, since these constraints cannot become active. If some of the regions $\mathcal{G}_i$ are convex, the stronger condition (12) can be used to infer $i \in \mathcal{A}(x)$. Let the set of these conditions be denoted by $\tilde{\mathcal{A}}(x)$. For the given state x the constraints $i \in \tilde{\mathcal{A}}(x)$ can be treated as *equality* rather than inequality constraints, thus further simplifying the online optimization problem.

Figure 2 juxtaposes the QP (5) and the reduced QPs that result from exploiting information on active and inactive constraints. Problems (13a), (13b) and (13c) in Fig. 2 are equivalent in the sense that the optimal z is unique due to the strict convexity of (5), and that z is a solution to all of the problems if it is the solution to one of them.

IV. Implementation

A simple algorithm to construct the regions of activity is introduced in Sect. IV-A. We stress this algorithm is not efficient but only introduced to allow a proof of concept for the use of the regions of activity. We refer to Sect. VI for further remarks. Section IV-B describes how to carry out the simplification of the QP described in Sect. III and Fig. 2 in each step of the online MPC.

A. Offline construction of the regions of activity

The index set $\mathcal{A}_j$ introduced in (9) denotes, for every $j \in \{1, \ldots, n_{\mathcal{P}}\}$, the set of constraints that are active on polytope $\mathcal{P}_j$. Based on the $\mathcal{A}_j$ we can find the subset of all polytopes on which a specific constraint $i \in \{1, \ldots, q\}$ is active. Specifically, constraint i is active for all $x \in \mathcal{P}_j$ with $j \in \mathcal{J}_i$, where

$$\mathcal{J}_i := \{ j \in \{1, ..., n_{\mathcal{P}}\} \,|\, i \in \mathcal{A}_j \}$$

for $i = 1, \ldots, q$. The region of activity of constraint i defined in (11) can now be constructed from the union of all polytopes on which constraint i is active, i.e.

$$\mathcal{G}_i = \bigcup_{k \in \mathcal{J}_i} \mathcal{P}_k$$

for $i = 1, \ldots, q$.

Algorithm 1 summarizes how the sets $\mathcal{J}_i$, $i = 1, \ldots, q$ and the regions of activity $\mathcal{G}_i$, $i = 1, \ldots, q$ can be constructed from the explicit solution (7). In addition to the sets $\mathcal{J}_i$ and the regions of activity $\mathcal{G}_i$, Alg. 1 provides $\bar{\mathcal{I}}$, the set of constraints that are never active on $\mathcal{X}$. These constraints $i \in \bar{\mathcal{I}}$ can be removed from the parametric program (5), since they are not active for any $x \in \mathcal{X}$.

We pointed out in Sect. III that the regions of activity $\mathcal{G}_i$ are in general not convex. Some $\mathcal{G}_i$ may be convex, however. If $\mathcal{G}_i$ is known to be convex for some i, we would like to use the stronger relation (12) instead of (14) for this region $\mathcal{G}_i$. Without restriction we assume there exists a $\bar{q} \leq q$ such that $\mathcal{G}_i$ are nonconvex or not known to be convex for all $i = 1, \ldots, \bar{q}$ and known to be convex for all $i = \bar{q}+1, \ldots, q$, where $\bar{q} = q$ is understood to mean that none of the $\mathcal{G}_i$ are known to be convex. Note that the proposed approach also works, if some or all convex $\mathcal{G}_i$ are treated like nonconvex regions. In this sense it is not crucial to determine if the $\mathcal{G}_i$ are convex or not.

Algorithm 1 is a brute-force approach that constructs the $q \ll n_{\mathcal{P}}$ regions of activity from the $n_{\mathcal{P}}$ polytopes of the explicit MPC law. It is obviously of interest to construct the regions of activity without having to compute the explicit MPC law first. Here, however, we are interested in constructing the regions of activity in a reliable, simple fashion in order to investigate their use for accelerating online MPC. Some remarks on better approaches are given in Sect. VI.

B. Online reduction of the MPC quadratic program

This section describes how to simplify the quadratic program (5) to be solved in time step t of the MPC algorithm. We stress that these simplifications are carried out *online*, i.e. once for every time t. The simplifications are based on $\mathcal{G}_i$, $i = 1, \ldots, q$, $\bar{\mathcal{I}}$, and $\bar{q}$, which are constructed as described in Sect. IV-A *offline*, i.e. once before the online MPC is started. Assume all constraints $i \in \bar{\mathcal{I}}$, where $\bar{\mathcal{I}}$ is defined as in Sect. IV-A, have been removed from (5). The index sets $\tilde{\mathcal{I}}(x)$ and $\tilde{\mathcal{A}}(x)$ can be constructed as summarized in Alg. 2.

The implication (14) is used to construct the subset $\tilde{\mathcal{I}}(x)$ of the inactive constraints for all $i = 1, \ldots, \bar{q}$. The stronger relation (12) is used to construct the subset $\tilde{\mathcal{A}}(x)$ of the active constraints for all regions of activity $\mathcal{G}_i$ that are known to be convex, i.e. for all $i = \bar{q} + 1, \ldots, q$. After determining $\tilde{\mathcal{I}}(x)$ and $\tilde{\mathcal{A}}(x)$, the MPC optimization problem (5), or equivalently (13a), can be replaced by the reduced quadratic program (13c). Note that the reduced QP (13c) can be solved analytically if $\mathcal{Q} = \tilde{\mathcal{I}}(x) \cup \tilde{\mathcal{A}}(x)$, i.e. all constraints

$$
\begin{array}{ll|ll|ll}
\min\limits_{z} & \frac{1}{2}z'Hz & \min\limits_{z} & \frac{1}{2}z'Hz & \min\limits_{z} & \frac{1}{2}z'Hz \\
\text{s.t.} & Gz \le w + Sx & \text{s.t.} & G^{\tilde{\mathcal{Q}}}z \le w^{\tilde{\mathcal{Q}}} + S^{\tilde{\mathcal{Q}}}x & \text{s.t.} & G^{\tilde{\mathcal{Q}}}z \le w^{\tilde{\mathcal{Q}}} + S^{\tilde{\mathcal{Q}}}x \\
 & & & G^{\tilde{\mathcal{A}}}z \le w^{\tilde{\mathcal{A}}} + S^{\tilde{\mathcal{A}}}x & & G^{\tilde{\mathcal{A}}}z = w^{\tilde{\mathcal{A}}} + S^{\tilde{\mathcal{A}}}x \\
 & & & G^{\tilde{\mathcal{I}}}z \le w^{\tilde{\mathcal{I}}} + S^{\tilde{\mathcal{I}}}x & & \cancel{G^{\tilde{\mathcal{I}}}z \le w^{\tilde{\mathcal{I}}} + S^{\tilde{\mathcal{I}}}x} \\
 & \text{(a)} & & \text{(b)} & & \text{(c)}
\end{array}
\tag{13}
$$

Fig. 2. The optimization problems (a), (b), (c) are equivalent. Problems (a) and (b) are equivalent, since they only differ with respect to notation. Problem (c) exploits that equality holds in $G^i z \le w^i + S^i x$ for every state x with $i \in \tilde{\mathcal{A}}(x)$. Similarly, $i \in \tilde{\mathcal{I}}(x)$ implies equality cannot hold. Since $G^i z \le w^i + S^i x$ cannot be active, constraints $i \in \tilde{\mathcal{I}}$ can be omitted in (c) for the given x while constraints $i \in \tilde{\mathcal{A}}$ can be treated as equality conditions. $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{A}}$ are short for $\tilde{\mathcal{I}}(x)$ and $\tilde{\mathcal{A}}(x)$, respectively. $\tilde{\mathcal{Q}}$ denotes $\tilde{\mathcal{Q}} = \mathcal{Q}\backslash(\tilde{\mathcal{A}} \cup \tilde{\mathcal{I}})$.

are known to be either active or inactive at the current state x.

C. Some remarks on the reduced QP (13c)

If the index set $\tilde{\mathcal{A}}(x)$ is nonempty for the current state x, some of the inequality constraints in the original quadratic program (13a) become equality constraints in the reduced problem (13c). This simplifies the quadratic program, because, loosely speaking, it has to be solved on a subspace only. On the other hand, an additional QR decomposition is required, which has to be accounted for in the comparisons in Sect. V. We briefly summarize the necessary computations for completeness (see [7] for similar calculations in a different context, also see [10]). We assume that $\mathcal{G}^{\tilde{\mathcal{A}}}$ has full rank. If this is not the case, an arbitrary subset of the rows of $\mathcal{G}^{\tilde{\mathcal{A}}}$ can be chosen such that this assumption holds. Let $G_{\mathcal{N}}^{\tilde{\mathcal{A}}}$ be a matrix of column vectors that span the null space and let $G_{\mathcal{R}}^{\tilde{\mathcal{A}}}$ be a matrix of column vectors that span the range space of $G^{\tilde{\mathcal{A}}}$, where $\tilde{\mathcal{A}} = \tilde{\mathcal{A}}(x)$ for short in the remainder of the section. We claim without proof that the choice of $G_{\mathcal{R}}^{\tilde{\mathcal{A}}}$ implies the matrix $[G_{\mathcal{N}}^{\tilde{\mathcal{A}}} \quad G_{\mathcal{R}}^{\tilde{\mathcal{A}}}]$ has full rank, $G^{\tilde{\mathcal{A}}}G_{\mathcal{N}}^{\tilde{\mathcal{A}}} = 0$, and $(G^{\tilde{\mathcal{A}}}G_{\mathcal{R}}^{\tilde{\mathcal{A}}})^{-1}$ exists. The equality constraints in (13c) define a $\operatorname{rank}(G^{\tilde{\mathcal{A}}})$-dimensional subspace in the n_z-dimensional space of the generalized inputs z. The optimal solution can uniquely be decomposed into

$$z = G_{\mathcal{R}}^{\tilde{\mathcal{A}}} z_{\mathcal{R}} + G_{\mathcal{N}}^{\tilde{\mathcal{A}}} z_{\mathcal{N}}, \tag{15}$$

Algorithm 1 Offline construction of the regions of activity $\mathcal{G}_i$ from the explicit solution (7).

init: $\bar{\mathcal{I}} \leftarrow \mathcal{Q}$, $\mathcal{J}_i \leftarrow \emptyset$ for all $i = 1, ..., q$
for all $i = 1, ..., q$ **do**
  for all $j = 1, ..., n_{\mathcal{P}}$ **do**
    if constraints i is in active set of $\mathcal{P}_j$ **then**
      $\mathcal{J}_i \leftarrow \mathcal{J}_i \cup \{j\}$
      $\bar{\mathcal{I}} \leftarrow \bar{\mathcal{I}} \backslash \{i\}$
    end if
  end for
  $\mathcal{G}_i = \bigcup\limits_{j \in \mathcal{J}_i} \mathcal{P}_j$
end for
return: $\mathcal{G}_i, i = 1, ..., q$

where uniqueness of the coefficients $z_{\mathcal{R}}$ and $z_{\mathcal{N}}$ follows from the full rank of $[G_{\mathcal{N}}^{\tilde{\mathcal{A}}} \quad G_{\mathcal{R}}^{\tilde{\mathcal{A}}}]$. Using (15) the equality constraints can be rewritten as

$$G^{\tilde{\mathcal{A}}}(G_{\mathcal{R}}^{\tilde{\mathcal{A}}} z_{\mathcal{R}} + G_{\mathcal{N}}^{\tilde{\mathcal{A}}} z_{\mathcal{N}}) = w^{\tilde{\mathcal{A}}} + S^{\tilde{\mathcal{A}}}x.$$

Since $G_{\mathcal{N}}^{\tilde{\mathcal{A}}}$ is in the null space of $G^{\tilde{\mathcal{A}}}$ by definition, this yields

$$z_{\mathcal{R}} = (G^{\tilde{\mathcal{A}}}G_{\mathcal{R}}^{\tilde{\mathcal{A}}})^{-1}(w^{\tilde{\mathcal{A}}} + S^{\tilde{\mathcal{A}}}x).$$

Replacing z by (15) and using the resulting solution $z_{\mathcal{R}}$, the QP in (13) can be stated as

$$\begin{array}{ll} \min\limits_{z_{\mathcal{N}}} & \frac{1}{2} z_{\mathcal{N}}' \hat{H} z_{\mathcal{N}} + \hat{F} z_{\mathcal{N}} \\ \text{s. t.} & \hat{G}^{\tilde{\mathcal{Q}}} z_{\mathcal{N}} \le \hat{w}^{\tilde{\mathcal{Q}}} + \hat{S}^{\tilde{\mathcal{Q}}}x, \end{array} \tag{16}$$

where $\hat{H} = (G_{\mathcal{N}}^{\tilde{\mathcal{A}}})' H G_{\mathcal{N}}^{\tilde{\mathcal{A}}}$, $\hat{G}^{\tilde{\mathcal{Q}}} = G^{\tilde{\mathcal{Q}}} G_{\mathcal{N}}^{\tilde{\mathcal{A}}}$, $\hat{F} = z_{\mathcal{R}}'(G_{\mathcal{R}}^{\tilde{\mathcal{A}}})' H G_{\mathcal{N}}^{\tilde{\mathcal{A}}}$, $\hat{w}^{\tilde{\mathcal{Q}}} = w^{\tilde{\mathcal{Q}}} - G^{\tilde{\mathcal{Q}}} G_{\mathcal{R}}^{\tilde{\mathcal{A}}} (G^{\tilde{\mathcal{A}}} G_{\mathcal{R}}^{\tilde{\mathcal{A}}})^{-1} w^{\tilde{\mathcal{A}}}$ and $\hat{S}^{\tilde{\mathcal{Q}}} = S^{\tilde{\mathcal{Q}}} - G^{\tilde{\mathcal{Q}}} G_{\mathcal{R}}^{\tilde{\mathcal{A}}} (G^{\tilde{\mathcal{A}}} G_{\mathcal{R}}^{\tilde{\mathcal{A}}})^{-1} S^{\tilde{\mathcal{A}}}$. The reduced QP is solved over the $q - \operatorname{rank}(G^{\tilde{\mathcal{A}}})$ dimensional space of the $z_{\mathcal{N}}$. If $\operatorname{rank}(G^{\tilde{\mathcal{A}}}) = n_z$, the solution of (16) is determined by the equality constraints alone. In this case, the null space is empty, the solution is given by $z = z_{\mathcal{R}}$ and the online optimization step can be omitted. The computational effort for the construction of the matrices $G_{\mathcal{R}}^{\tilde{\mathcal{A}}}$ and $G_{\mathcal{N}}^{\tilde{\mathcal{A}}}$ is taken into account in the numerical experiments reported in

Algorithm 2 Online construction of $\tilde{\mathcal{I}}(x)$ and $\tilde{\mathcal{A}}(x)$ for current state x.

init: $\tilde{\mathcal{I}}(x) \leftarrow \emptyset$, $\tilde{\mathcal{A}}(x) \leftarrow \emptyset$
for all $i = 1, \ldots, \bar{q}, i \notin \bar{\mathcal{I}}$ **do**
  if $x \notin \mathcal{G}_i$ **then**
    $\tilde{\mathcal{I}}(x) \leftarrow \tilde{\mathcal{I}}(x) \cup \{i\}$
  end if
end for
for all $i = \bar{q}+1, \ldots, q, i \notin \bar{\mathcal{I}}$ **do**
  if $x \notin \mathcal{G}_i$ **then**
    $\tilde{\mathcal{I}}(x) \leftarrow \tilde{\mathcal{I}}(x) \cup \{i\}$
  else
    $\tilde{\mathcal{A}}(x) \leftarrow \tilde{\mathcal{A}}(x) \cup \{i\}$
  end if
end for
return: $\tilde{\mathcal{I}}(x)$, $\tilde{\mathcal{A}}(x)$

Sect. V. Both matrices can be found by calculating the QR decomposition of $(G^{\tilde{\mathcal{A}}})'$. Furthermore note that the matrices $G^{\tilde{\mathcal{A}}}_{\mathcal{R}}$ and $G^{\tilde{\mathcal{A}}}_{\mathcal{N}}$ need not be calculated if $\tilde{\mathcal{A}}(x) = \emptyset$ in time step t.

V. EXAMPLE

Consider the discrete-time double integrator that results from zero-order hold discretization and sample time $T_s = 0.01$s

$$x(t+1) = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix} x(t) + \begin{pmatrix} \frac{1}{2}T_s^2 \\ T_s \end{pmatrix} u(t),$$

subject to the linear input and state constraints

$$\begin{aligned} -10 \leq & u(t) \leq 10, \\ [-25 \quad -25]' \leq & x(t) \leq [25 \quad 25]', \end{aligned}$$

with horizons $N_y = 40$, $N_c = N_y$ and $N_u = N_y - 1$. The matrices P and K are set to the solution of the Riccati equation (4). The weighting matrices are chosen as $Q = \mathrm{diag}([1 \quad 1])$ and $R = 10$. A parametric quadratic program of the form (3) with $q = 244$ constraints results for this system, which can be solved with the MPT toolbox [11]. The MPT toolbox yields a piecewise affine control law on a state space partition with $n_{\mathcal{P}} = 3012$ polytopes.

We conduct numerical experiments to compare a standard online MPC implementation to two variants of the accelerated online MPC proposed here. The following abbreviations are introduced for ease of reference:

full-QP: Solve the QP (5) in each time step t. This corresponds to an MPC without any of the reductions proposed here.

red.-QP-1: Use the weak relation (14) to detect and omit inactive constraints in each time step t, and solve the reduced QP (13c). Note that $\tilde{\mathcal{A}}(x) = \emptyset$ in this variant for all t, since (12) is not used.

red.-QP-2: Proceed as in red.-QP-1. In addition, use the strong relation (12) to convert inequality constraints to equality constraints for convex $\mathcal{G}_i$ in each time step.

Each of these three MPC variants, i.e. full-QP, red.-QP-1, and red.-QP-2, is combined with an active set QP solver, and an interior point QP solver, to result in a total of six MPC implementations. We use the QP solver quadprog of the Matlab optimization toolbox with the active-set and interior-point-convex options[2]. We choose these generic solvers for simplicity, for ease of reproducibility in future comparisons, and in order not to mix effects due to tailored QP solvers with the reductions to be assessed here. We apply each of the six MPC variants to 10^5 random feasible initial conditions and record the computation time needed to solve one time step of (3) for every variant and initial condition. The computation times reported for the variant red.-QP-1 include the times needed to determine $\tilde{\mathcal{I}}(x)$ and to create the respective reduced QP. The computation times reported for the variant red.-QP-2 include the times needed to determine $\tilde{\mathcal{I}}(x)$, $\tilde{\mathcal{A}}(x)$, and to create the respective reduced QP, including the QR decomposition discussed in Sect. IV-B.

[2]Intel Core2Quad Q9550 processor, Suse Linux, Matlab R2011a. No time was spent to optimize the code.

The results of the numerical experiments are summarized in Tab. I and Fig. 3. Consider the results for the red.-QP-1 variant first (first and third part of Tab. I and Fig. 3a, c). The average computation times for both solvers are smaller for red.-QP-1 than for full-QP (roughly 30% (53%) reduction for the active-set (interior-point) solver, respectively). Now consider the results for the red.-QP-2 variant (second and forth part of Tab. I and Fig. 3b, d). Combining the active-set solver with red.-QP-2 results in a reduction of the same order as with variant red.-QP-1 (roughly 31% on average). The interior-point solver leads to an reduction of the average computation time of roughly 47%, which is slightly smaller than the red.-QP-1 variant. The significant reduction of minimal time, in both variants, corresponds to the case when all constraints are known to be either active or inactive, and consequently no optimization problem has to be solved.

We note that it took 1.72s and 0.44s to construct the regions of activity $\mathcal{G}_i$ with a simple Matlab implementation of Alg. 2 and to check which of the $\mathcal{G}_i$ are convex, respectively. We found 80 regions of activity to be convex, 80 to be not convex and 84 constraints which are not active in any polytope. Since these calculations are carried out before runtime of the MPC controllers, the time required for them does not affect the times reported in Tab. I and Fig. 3.

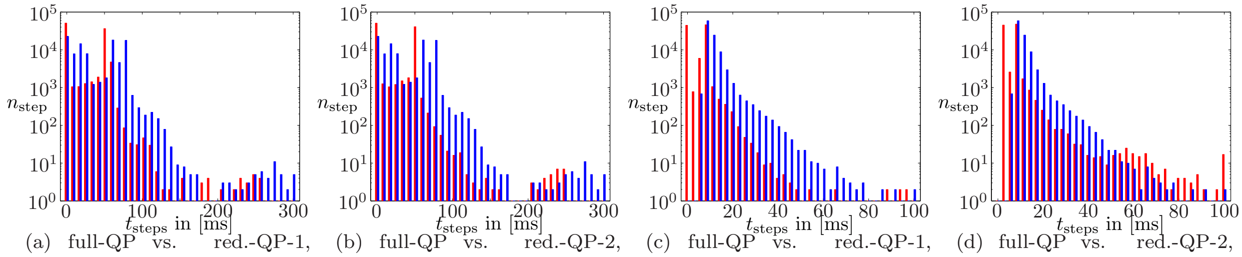
Finally, we note that the maximal time, $t_{\max}$, spent for solving a single MPC (3) is also reduced significantly in all cases. We report the times $t_{\max}$ in Tab. I, because maximal computation times of MPC algorithms are of interest from a practical point of view.

VI. CONCLUSIONS AND OUTLOOK

We introduced a new online MPC variant that combines aspects of online and explicit MPC. Essentially, information on the explicit structure of the control law is used to accelerate the online solution of QPs. The proposed method is fundamentally different from existing ones with the same goal [3]–[5].

The number of polytopes, $n_{\mathcal{P}}$, needed to be stored to implement an explicit MPC controller grows rapidly in the number q of constraints of the MPC problem (presumably $n_{\mathcal{P}}$ grows exponentially in q). The method proposed here requires to store only q polytopes, the regions of activity $\mathcal{G}_i$, instead of the $n_{\mathcal{P}} \gg q$ polytopes of the explicit MPC law.

The proposed method requires to carry out *offline* computations. Since the paper intends to assess the *online* speedup that can be achieved, the offline calculations are implemented in a very simple fashion. We obviously need to investigate how to calculate the q regions of activity without calculating the $n_{\mathcal{P}} \gg q$ polytopes of the explicit MPC law first. Future research is devoted to investigating an approach based on the vertices of the control law [12]. By using the vertices and their surrounding active sets we can avoid to calculate all hyperplanes that define the state space partition. Moreover, the vertices can be used to speed up the computation of convex hulls $\mathrm{conv}(\mathcal{G}_i)$ from $\mathcal{G}_i$.



(a) full-QP vs. red.-QP-1, active-set
(b) full-QP vs. red.-QP-2, active-set
(c) full-QP vs. red.-QP-1, interior-point-convex
(d) full-QP vs. red.-QP-2, interior-point-convex

Fig. 3. Histograms of computation times t_{steps} required to solve the MPC problem (3) once. A pair $(t_{\text{steps}}, n_{\text{steps}})$ gives the number n_{steps} of runs among the total of 10^5 randomized runs that required the computation time t_{steps}. The four diagrams correspond to the four reduced variants, cf. the label underneath the figures. Results for the respective full QP variant are shown in blue in every diagram for ease of comparison. Data for the reduced variants is shown in red. Note the intervals shown on for t_{steps} differ from diagram to diagram.

	active-set solver					interior point solver				
	full QP	red.-QP-1	Diff.	red.-QP-2	Diff.	full QP	red.-QP-1	Diff.	red.-QP-2	Diff.
t_{max}	308.432	256.127	-16.96%	252.512	-18.13%	252.534	96.856	-61.65%	147.589	-41.56%
t_{min}	2.685	1.126	-58.06%	0.706	-73.71%	6.073	1.143	-81.18%	1.714	-71.78%
t_{avg}	37.079	25.862	-30.25%	25.600	-30.96%	10.601	4.981	-53.02%	5.659	-46.62%

TABLE I

SUMMARY OF THE NUMERICAL EXPERIMENTS. SYMBOLS $t_{\max}$ AND $t_{\min}$ DENOTE THE MAXIMUM AND MINIMUM TIME REQUIRED AMONG THE 10^5 RUNS; t_{avg} DENOTES THE AVERAGE OVER THE 10^5 RUNS. TIMES ARE GIVEN IN MILLISECONDS.

We note that it is straightforward to construct new variants of the proposed algorithm, and we claim that online MPC without any of the proposed accelerations on the one hand, and explicit MPC on the other hand can be conceived as limiting cases of all variants. This can be seen as follows. Our numerical experiments show that it is beneficial to remove some constraints that are inactive even if not all inactive constraints can be found. Consequently, the regions of activity $\mathcal{G}_i$ can be replaced by more or less coarse outer approximations $\bar{\mathcal{G}}_i$ of $\mathcal{G}_i$ (cf. Sect. IV-A). In fact the convex hulls $\bar{\mathcal{G}}_i = \text{conv}(\mathcal{G}_i)$ are merely one convenient choice for $\bar{\mathcal{G}}$. Other outer approximations, e.g. based on hyperrectangles or -ellipsoids, will provide other tradeoffs between runtime storage requirement and reduction of the number of regions to a number $\hat{q} < q$. Conversely, it is reasonable to explore variants that cover $\mathcal{X}$ with $\hat{q}$ regions where $q \leq \hat{q} \leq n_{\mathcal{P}}$. To this end note that there always exist a partition for every $\mathcal{G}_i$ into several convex polytopes. This claim holds, since $\mathcal{G}_i$ is the union of convex polytopes by construction. Other partitions of $\mathcal{G}_i$ may exist, however. This simple insight suggests to attempt partitioning every $\mathcal{G}_i$ into as few as possible convex polytopes such that $\hat{q}$ regions cover the state space, where $q \leq \hat{q} \leq n_{\mathcal{P}}$. The explicit control law is the limiting case $\hat{q} = n_{\mathcal{P}}$. In this case no online QP has to be solved at all, since the exact active set $\mathcal{A}(x)$ can be determined for the current x, and since $\mathcal{A}(x)$ defines the control action.

Finally, we note that the proposed method can be applied to other parametric optimization problems.

ACKNOWLEDGMENT

Support by the Deutsche Forschungsgemeinschaft (DFG) under grant MO 1086/11-1 is gratefully acknowledged.

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. Seron, G. Goodwin, and J. De Dona, "Characterisation of receding horizon control for constrained systems," *Asian Journal of Control*, vol. 5, pp. 271 – 286, 2003.

[3] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, vol. 18, pp. 816 – 830, 2008.

[4] G. Pannocchia, J. B. Rawlings, and S. J. Wright, "Fast, large-scale model predictive control by partial enumeration," *Automatica*, vol. 43, pp. 852 – 860, 2007.

[5] G. Pannocchia, S. J. Wright, and J. B. Rawlings, "Partial enumeration MPC: Robust stability results and application to an unstable CSTR," *Journal of Process Control*, vol. 21, no. 10, pp. 1459 – 1466, 2011.

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

[7] P. Tøndel, T. A. Johansen, and A. Bemporad, "Further results on multiparametric quadratic programming," in *Proc. of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 3173 – 3178.

[8] P. Patrinos and H. Sarimveis, "A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings," *Automatica*, vol. 46, no. 9, pp. 1405 – 1418, 2010.

[9] 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, pp. 2209 – 2214, 2006.

[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Elsevier, 2007.

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

[12] M. Mönnigmann and M. Jost, "Vertex based calculation of explicit MPC laws." in *Proceedings of the 2012 American Control Conference*, Montreal, Canada, 2012, pp. 423 – 428.