Constraint removal in linear MPC: an improved criterion and complexity analysis

Michael Jost, Gabriele Pannocchia and Martin Mönnigmann

Abstract— **Constraint removal accelerates model predictive control by detecting inactive constraints at the yet unknown optimal solution and removing them from the online optimization problem. We show in this paper that the number of removed constraints can be increased further by generalizing previously used inactivity criteria. The proposed generalization does not depend on information available at previous time steps, and consequently can also be applied at the initial state. In addition, we provide a detailed analysis of the computational complexity of the proposed variant and of existing constraint removal methods, applied to both active-set (AS) and interior-point (IP) solvers. Finally, we compare the different constraint removal variants in numerical experiments to corroborate the complexity analysis carried out, showing the greatest benefits of the proposed variant, especially with IP solvers.**

I. Introduction

Model predictive control (MPC) is a well-known approach to control large, constrained, multivariable systems. However, since MPC requires solving optimal control problems on a receding horizon, it is computationally expensive.

For linear systems subject to state and input constraints and with quadratic weights on the states and inputs, the optimization problem to be solved during runtime is a quadratic program (QP). The solution to this QP is a piecewise affine function of the state [1], [2]. Several researchers exploited the knowledge of the structure of the solution to reduce the required online calculations. Ferreau et al. [3] used the fact that the set of constraints that are active at the optimal solution of successive QPs does not change too much. Pannocchia et al. [4] enumerate the active sets that occur most frequently during runtime. More recently, we showed that the time required to calculate the optimal control law can be reduced by removing constraints from the optimization problem that are found to be inactive at the optimum before actually solving it [5]. Several methods to find such a priori inactive constraints have been proposed [5]–[8]. Some of them detect inactive constraints using precalculated, structural information, others do not require any offline preparation. We stress that constraint removal (CR) does not only remove redundant constraints, i.e., constraints that cannot be active for any state, from the QP, but the set of a priori inactive constraints is a function of the initial state of the optimal control problem.

In this paper we propose an extension to the CR techniques presented in [5], [7] that is based on Hölder's inequality. With the proposed modification the number of a priori inactive constraints can be increased, which further reduces the required calculation time. We also analyze the complexity of CR-MPC in combination with both active-set and interior-point solvers. Finally, an example is used to demonstrate the effectiveness of our approach.

We state the problem class and summarize existing constraint removal variants in Sects. II and III, respectively. The main result, i.e., the improved inactivity criterion is given in Sect. IV. The computational complexity of existing variants and the new one are compared in Sect. V. We apply the proposed approach to an example in Sect. VI and give some conclusions in Sect. VII.

II. Problem Statement

We consider linear discrete-time state space systems:

$$x(t+1) = Ax(t) + Bu(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the input, and (A, B) is assumed to be stabilizable. We assume state and input constraints

$$x(t) \in \mathbb{X} \subset \mathbb{R}^n, \quad u(t) \in \mathbb{U} \subset \mathbb{R}^m, \tag{2}$$

must be respected for all t, where $\mathbb{U}$ and $\mathbb{X}$ are compact, full-dimensional polytopes that contain the origin in their interior. The model predictive control problem treated here (see e.g. [9] for details) reads

$$\begin{aligned}\min_{U,X} \quad & \ell_f(\hat{x}(N)) + \sum_{k=0}^{N-1} \ell(\hat{x}(k), u(k)) \\ \text{s.t.} \quad & \hat{x}(k+1) = A\hat{x}(k) + Bu(k),\ k = 0, \ldots, N-1, \\ & \hat{x}(0) = x, \\ & \hat{x}(k) \in \mathbb{X}, \qquad k = 0, \ldots, N, \\ & u(k) \in \mathbb{U}, \qquad k = 0, \ldots, N-1, \\ & \hat{x}(N) \in \mathbb{X}_f,\end{aligned} \tag{3}$$

where $\hat{x}(k)$, $k = 0, \ldots, N$ are the predicted system states, x is the current system state, $X = (\hat{x}'(0), \ldots, \hat{x}'(N))'$, $U = (u'(0), \ldots, u'(N-1))'$, and N is the horizon length. We consider a quadratic stage and terminal cost, i.e., $\ell(x, u) = \frac{1}{2}x'\,Q\,x + \frac{1}{2}u'\,R\,u$ and $\ell_f(x) = \frac{1}{2}x'\,P\,x$, where $P \in \mathbb{R}^{n\times n}$, $P \succeq 0$, $Q \in \mathbb{R}^{n\times n}$, $Q \succeq 0$ and $R \in \mathbb{R}^{m\times m}$, $R \succ 0$ are the terminal, state and input weighting matrices, respectively, and $(Q^{1/2}, A)$ is detectable. Moreover, we assume the terminal set $\mathbb{X}_f \subset \mathbb{R}^n$ to be a full-dimensional polytope containing the origin in its interior. Under these assumptions it can be shown that (3) is a strictly convex optimization problem [1]. The MPC problem (3) can be reformulated to read

$$\begin{aligned}\min_{U} \quad & V(x, U), \\ \text{s. t.} \quad & GU - Ex \leq w,\end{aligned} \tag{$\mathbb{P}(x)$}$$

where $V(x, U) = {}^{1}\!/\!{}_{2}x'Yx + x'FU + {}^{1}\!/\!{}_{2}U'HU$, and $Y \in \mathbb{R}^{n\times n}$, $F \in \mathbb{R}^{n\times mN}$, $H \in \mathbb{R}^{mN\times mN}$, $G \in \mathbb{R}^{q\times mN}$, $E \in$

M. Jost and M. Mönnigmann are with Automatic Control and Systems Theory, Department of Mechanical Engineering, Ruhr-Universität Bochum, Germany. G. Pannocchia is with Department of Civil and Industrial Engineering, University of Pisa, Italy. E-mail: michael.s.jost@rub.de, gabriele.pannocchia@unipi.it, martin.moennigmann@rub.de.

$\mathbb{R}^{q\times n}$, $w \in \mathbb{R}^q$; and q denotes the number of constraints. Since (3) is a strictly convex QP, i.e., the matrix H in $\mathbb{P}(x)$ is positive definite, the solution to $\mathbb{P}(x)$ is unique if it exists [1]. Finally, we note that the reformulation of the optimal control problem (3) given in $\mathbb{P}(x)$, in which states are eliminated from the QP decision variables, is only convenient but not necessary for the constraint removal approaches proposed in this paper.

A. Notation

For any real matrix M, M^i denotes the i-th row of M. For any $\xi \in \mathbb{R}^n$, the sum norm (l_1-norm), Euclidean norm (l_2-norm), and max norm (l_∞-norm) are defined by $\|\xi\|_1 = \sum_{i=1}^n |\xi_i|$, $\|\xi\|_2 = \sqrt{\sum_{i=1}^n \xi_i^2}$, and $\|\xi\|_\infty = \max(|\xi_i|)$, respectively. The set of states for which $\mathbb{P}(x)$ is feasible is denoted by $\mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{X}$. Furthermore, $U^\star : \mathcal{X} \to \mathbb{U}^N$ and $u^\star : \mathcal{X} \to \mathbb{U}$ denote the optimal solution to $\mathbb{P}(x)$ and its first m elements, respectively, and $V^\star : \mathcal{X} \to \mathbb{R}$, $V^\star(x) = V(x, U^\star(x))$ is the value of the cost function in $\mathbb{P}(x)$ at the optimum.

Let $\mathcal{Q} = \{1, \ldots, q\}$ denote the index set of all constraints. A constraint $i \in \mathcal{Q}$ is said to be *active* for any given $x \in \mathcal{X}$, if $G^iU^\star(x) - E^ix = w^i$ and *inactive*, if $G^iU^\star(x) - E^ix < w^i$. Let

$$\begin{aligned} &\mathcal{A}(x) = \left\{ i \in \mathcal{Q} \,\middle|\, G^iU^\star(x) - E^ix = w^i \right\}, \\ &\mathcal{I}(x) = \mathcal{Q} \backslash \mathcal{A}(x), \end{aligned} \tag{4}$$

be short for the sets of active and inactive constraints at $x \in \mathcal{X}$, respectively. Finally, note that the constraints of $\mathbb{P}(x)$ define a convex polytope in the combined space of inputs and states

$$\mathcal{P} = \left\{ (U, x) \in \mathbb{R}^{mN} \times \mathbb{R}^n \,|GU - Ex \le w \right\}, \tag{5}$$

which is full dimensional, bounded and contains the origin in its interior if the assumptions on $\mathbb{X}$ and $\mathbb{U}$ hold.

III. Brief summary of constraint removal

The quadratic program $\mathbb{P}(x)$ can be simplified by removing some (or all) of the constraints that are inactive at the optimal solution *before* solving it. More precisely, the following result holds.

Proposition 1 (Reduced optimization problem [8], [10]): Let $x \in \mathcal{X}$ be arbitrary and let $\tilde{\mathcal{I}} \subseteq \mathcal{I}(x)$ be any subset of the inactive constraints. Consider the reduced optimization problem

$$\begin{aligned} \tilde{V}^\star(x) = \min_{\tilde{U}} \quad & \frac{1}{2}x'Yx + x'F\tilde{U} + \frac{1}{2}\tilde{U}'H\tilde{U}, \\ \text{s. t.} \quad & G^{\mathcal{Q}\backslash\tilde{\mathcal{I}}}\tilde{U} - E^{\mathcal{Q}\backslash\tilde{\mathcal{I}}}x \le w^{\mathcal{Q}\backslash\tilde{\mathcal{I}}}. \end{aligned} \tag{$\tilde{\mathbb{P}}(x)$}$$

Then $\tilde{\mathbb{P}}(x)$ and $\mathbb{P}(x)$ have the same unique solution $\tilde{U}^\star(x) = U^\star(x)$.

In order for Prop. 1 to be useful, some or all of the inactive constraints must obviously be identified before actually solving $\tilde{\mathbb{P}}(x)$. Several approaches to identifying inactive constraints have been presented by the authors of the present paper, see e.g. [5]–[7], [10]. We briefly summarize two of these in Secs. III-A and III-B, because they are required in the subsequent sections.

A. Constraint removal with bounds on $V^\star(x)$ (prelyap-MPC)

We showed in [8] that the optimal value function $V^\star(x)$ can be used to detected inactive constraints. Essentially, a lower bound $\sigma_i^\star$ on the optimal value function $V^\star(x)$ can be calculate before runtime that satisfies

$$\sigma_i^\star < V^\star(x) \text{ for all } x \in \mathcal{G}_i, \tag{6}$$

where $\mathcal{G}_i = \{x \in \mathcal{X} | \, i \in \mathcal{A}(x)\}$ is the so-called region of activity [5], [6] of constraint i [8]. If, for any $x \in \mathcal{X}$, the optimal cost function $V^\star(x)$ is smaller than the lower bound $\sigma_i^\star$, the corresponding constraint must be inactive. Unfortunately, $V^\star(x)$ is only known *after* solving the optimization problem. However, if $V^\star(x)$ is a Lyapunov function of the closed-loop system, it strictly decreases along any trajectory. If we combine this property with the bound $\sigma_i^\star$, the following lemma results.

Lemma 1 ([8], [10]): Let $x(t_0) \in \mathcal{X}$ be arbitrary, let $i \in \mathcal{Q}$ be an arbitrary constraint and assume $\sigma_i^\star \in \mathbb{R}$ satisfies (6). Then

$$V^\star(x(t_0)) < \sigma_i^\star \Rightarrow i \in \mathcal{I}(x(t)), \quad t \ge t_0,$$

i.e., constraint i remains inactive along the trajectory of the closed-loop system.

B. Online constraint removal (lyap-MPC)

It is required to calculate the bounds $\sigma_i^\star$, $i = 1, \ldots, q$ before runtime in the method presented in Sect. III-A. Unfortunately, these bounds $\sigma_i^\star$ must be recalculated whenever parameters of the MPC problem (3) such as weighting matrices or horizons are changed.

In [7], we proposed a method that does not require any precalculated bounds. The method is based on the following two ideas. First, an upper bound (denoted $\kappa_2(x)$ below) on the Euclidean norm of the optimal $U^\star(x)$ can be used to detect inactive constraints. This is summarized in Lemma 2.

Lemma 2 ([7], [10]): Let $x \in \mathcal{X}$ be arbitrary and assume there exists $\kappa_2(x) \ge 0$ such that $\|U^\star(x)\|_2 \le \kappa_2(x)$. Then

$$\left\|G^i\right\|_2 \kappa_2(x) < E^ix + w^i \Rightarrow i \in \mathcal{I}(x), \tag{7}$$

i.e. constraint i is inactive at the optimal solution to $\mathbb{P}(x)$. Secondly, if the optimal value function is a Lyapunov function of the closed-loop system, an appropriate bound $\kappa_2(x)$ can be calculated by exploiting its geometric properties. This is summarized in the following lemma.

Lemma 3 ([7]): Let $x \in \mathcal{X}$ be arbitrary, and assume $U^\star(x^-)$, the optimal input sequence at the preceding time step, has been found. Let $\rho(x^-) = V^\star(x^-) - \frac{1}{2}x'(Y - FH^{-1}F')x$. Then the Euclidean norm of the optimal $U^\star(x)$ is bounded above by

$$\kappa_2(x) = \sqrt{2\rho(x^-)\lambda_{\min}^{-1}(H)} + \left\|H^{-1}F'x\right\|_2, \tag{8}$$

i.e., $\|U^\star(x)\|_2 \le \kappa_2(x)$.

The upper bound $\kappa_2(x)$ on $\|U^\star(x)\|_2$, which is given in Lem. 3, can be used together with (7) to detect inactive constraints [7].

IV. An extension to Constraint Removal using Hölder's inequality

The proposed approaches detect a priori inactive constraints based on the optimal value function. Since the value of $V^\star(x)$ must be known to identify inactive constraints, neither Lemma 1 and Lemma 2 can be applied at the initial state[1].

[1] We note for completeness that extensions to Lem. 1 and Lem. 2 are proposed in [8] and [7] that allow to remove some inactive constraints even at the initial state. These extensions are only able to remove *redundant* constraints, however.

The criterion presented here allows to remove inactive constraints during the first time step as well.

We first extend Lemma 2 by generalizing Cauchy and Schwarz's inequality by Hölder's inequality.

Lemma 4: Let $x \in \mathcal{X}$ be arbitrary and assume there exists $\kappa_\infty \geq 0$ such that $\|U^\star(x)\|_\infty \leq \kappa_\infty$. If, for any $i \in \mathcal{Q}$,

$$\left\|G^i\right\|_1 \kappa_\infty < E^i x + w^i, \tag{9}$$

then constraint i is inactive at the optimal solution to $\mathbb{P}(x)$ for x, i.e., $G^i U^\star(x) < E^i x + w^i$.

Proof: Consider the relations

$$G^i U^\star(x) \leq |G^i U^\star(x)| \leq \|G^i\|_1 \|U^\star(x)\|_\infty \leq \|G^i\|_1 \kappa_\infty,$$

which hold, because $\nu \leq |\nu|$ for all $\nu \in \mathbb{R}$, according to Hölder's inequality, and because $\|U^\star(x)\|_\infty \leq \kappa_\infty$ by assumption, respectively. Together with (9) this implies $G^i U^\star(x) < E^i x + w^i$, which proves the claim. ■

It remains to show how to calculate an upper bound κ_∞ for $\|U^\star(x)\|_\infty$. We stress x is not fixed in the following lemma but varies over $x \in \mathcal{X}$ in (10) in particular.

Lemma 5: Define $\nu = (\nu_1, \ldots, \nu_{mN})'$ by

$$\nu_i = \max_{(U,x) \in \mathcal{P}} |U_i|, \quad i = 1, \ldots, mN \tag{10}$$

Then, $\|U^\star(x)\|_\infty$ is bounded above by $\kappa_\infty = \|\nu\|_\infty$ for all $x \in \mathcal{X}$, i.e.,

$$\|U^\star(x)\|_\infty \leq \kappa_\infty \text{ for all } x \in \mathcal{X}.$$

Proof: Let $x \in \mathcal{X}$ be arbitrary and let $U^\star(x)$ be the corresponding optimal input sequence. Since $(U^\star(x), x) \in \mathcal{P}$, we have $|U_i^\star(x)| \leq \nu_i$ for all $i = 1, \ldots, mN$ by construction of ν_i in (10). Consequently, $\|U^\star(x)\|_\infty \leq \kappa_\infty$. Since $x \in \mathcal{X}$ was arbitrary, this proves the claim. ■

We collect several comments on the bound κ_∞:

(i) In contrast to $\kappa_2(x)$ from Sec. III-B, the bound κ_∞ is independent of the state and thus constant during runtime.

(ii) The calculation of the bound κ_∞ requires to solve mN linear programs, which are of similar size (in terms of their number of constraints) as the MPC problem itself.

(iii) If the system is subject to input box constraints of the form $u^- \leq u \leq u^+$ only, we have $\kappa_\infty = \max(\|u^+\|_\infty, \|u^-\|_\infty)$. In this case, no linear programs have to be solved.

Lemma 4 can be used to remove constraints before solving the optimization problem. However, the bound κ_∞ does not depend on the state and therefore may be too conservative to significantly reduce the number of constraints. This is different to the approaches summarized in Sec. III. Both approaches summarized in Sec. III detect more and more constraints to be inactive as the system converges to the origin until only the unconstrained solution remains. In order to retain this property, the inactivity criteria from Sec. III and Sec. IV can be combined. More precisely, we consider the following combinations of criteria:

prelyap-MPC: Inactive constraints are detected with precalculated bounds on $V^\star(x)$ as described in Lemma 1. (11-1)

höl-prelyap-MPC: Inactive constraints are detected with both Lemma 1 and the new criterion from Lemma 4. (11-2)

lyap-MPC: Online constraint removal as described in Lemma 2. (11-3)

höl-lyap-MPC: Online constraint removal with both Lemma 2 and the new criterion from Lemma 4. (11-4)

The implementation of höl-lyap-MPC is described in more detail in Alg. 1, which can be summarized as follows: First, the set $\tilde{\mathcal{I}}_2 = \left\{ i \in \mathcal{Q} \,\middle|\, \left\|G^i\right\|_2 \kappa_2(x) < E^i x + w^i \right\}$ of inactive constraints is calculated with Lemma 2 (line 2) [7]. Then, Hölder's inequality is used to construct the set $\tilde{\mathcal{I}}_\infty = \left\{ i \in \mathcal{Q} \,\middle|\, \left\|G^i\right\|_1 \kappa_\infty < E^i x + w^i \right\}$ (lines 3-5). The tighter bound is selected in lines 6-8. If the reduced QP is detected to be unconstrained, the unconstrained solution is applied. Otherwise, the reduced optimization problem $\tilde{\mathbb{P}}(x)$ is solved (lines 9-10). The other three variants can be implemented accordingly.

V. Some notes on the computational complexity

We analyze the computational complexity of MPC with constraint removal in this section. Let $\mathcal{O}(\cdot)$ denote the order of the number of elementary floating point arithmetic operations such as additions, multiplications and comparisons of real numbers necessary for a given calculation. Let $\vartheta_{\text{LP}}(s,t)$ and $\vartheta_{\text{QP}}(s,t)$ refer to the number of arithmetic operations required to solve a linear program and a quadratic program, respectively, with s decision variables and t constraints.

The section is split into three parts: We first analyze the computational effort required for preparatory calculations of the four methods (11) in Sect. V-A. Note these are the calculations that can be carried out before runtime of the actual MPC controller. They are therefore referred to as *offline* calculations. Subsequently, we discuss the number of operations required at runtime, i.e., *online*, for detecting inactive constraints and solving the resulting reduced QPs in Sect. V-B. Since the computational effort for solving a QP depends on the chosen method and solver, Sect. V-C gives a more detailed account of these numbers, distinguishing between active-set and interior-point solvers.

A. *Offline computational effort of the variants* (11)

prelyap-MPC, höl-prelyap-MPC: For this MPC variant, a lower bound $\sigma_i^\star$ on the objective function has to be calculated for each constraint (cf. Sect. III-A). As shown in [8], this requires solving a strictly convex quadratic program with $mN+n-1$ decision variables and $q-1$ inequality constraints. Thus, $q\vartheta_{\text{QP}}(mN+n-1, q-1)$ operations are required in total. For höl-prelyap-MPC the bound κ_∞ is required, which can be obtained by solving mN linear programs (cf. Lem. 5). Therefore, the total number of preparatory calculations is

$$q\vartheta_{\text{QP}}(mN+n-1, q-1) + mN\vartheta_{\text{LP}}(mN+n, q).$$

lyap-MPC, höl-lyap-MPC: The required preparatory operations are the calculation of the eigenvalue $\lambda_{\min}(H)$, the norms of G^i, $i = 1, \ldots, q$, and the matrices $H^{-1}F'$ and $Y - FH^{-1}F'$. Simple calculations show that the preparatory calculations in total are of order $\mathcal{O}\left(m^3N^3\right)$, where we assumed $n \leq mN$, the Cholesky factorization is used to calculate the inverse of the matrix H, and the positive definiteness of H is exploited. Again, for höl-lyap-MPC the bound κ_∞ is required, which can be obtained by solving

Algorithm 1 CR-MPC with Hölder's inequality.

```
**Input:** $x$, $\kappa_\infty$
Calculate set $\tilde{\mathcal{I}}_2$ using Lemma 2 and set $\tilde{\mathcal{I}}_\infty = \emptyset$.
**for all** $i = 1$ **to** $q$ **do**
    **if** $\left\|G^i\right\|_1 \kappa_\infty < E^i x + w^i$, **then** Constr. $i$ is inactive at state $x$: $\tilde{\mathcal{I}}_\infty \leftarrow \tilde{\mathcal{I}}_\infty \cup \{i\}$.
**end for**
Use best bound for constraint removal:
**if** $|\mathcal{I}_2| < |\mathcal{I}_\infty|$ **then** $\mathcal{Q} \leftarrow \mathcal{Q} \backslash \mathcal{I}_\infty$.
**else** $\mathcal{Q} \leftarrow \mathcal{Q} \backslash \mathcal{I}_2$.
**if** $\mathcal{Q} = \emptyset$ **then** QP is unconstrained: $U^\star(x) = -H^{-1}F'x$.
**else** Solve reduced QP $\tilde{\mathbb{P}}(x)$.
**Output:** $U^\star(x)$, $V^\star(x)$.
```

mN linear programs. In summary, we have

$$\mathcal{O}\left(m^3N^3\right) + mN\vartheta_{\text{LP}}(mN+n, q)$$

for this MPC variant.

B. Online computational effort of the variants (11)

The number of constraints in the reduced optimization problem $\tilde{\mathbb{P}}(x)$ is denoted by $\tilde{q} = q - |\tilde{\mathcal{I}}|$ for brevity.

prelyap-MPC, höl-prelyap-MPC: The prelyap-MPC variant requires testing whether $V^\star(x) < \sigma_i^\star$ for every i which involves only the comparison of two real numbers. Thus, q operations are required to construct the set $\tilde{\mathcal{I}}$. The optimal input can be found by solving the reduced optimization problem $\tilde{\mathbb{P}}(x)$, which requires $\vartheta_{\text{QP}}(mN, \tilde{q})$ operations. Thus, $q + \vartheta_{\text{QP}}(mN, \tilde{q})$ operations are required in total. The same computations need to be carried out in the höl-prelyap-MPC variant, which, however, additionally uses the bound κ_∞ and the criterion stated in Lem. 4. Since, evaluating the criterion (9) from Lemma 4 amounts to $q\,\mathcal{O}\left(n\right)$ operations,

$$q\,\mathcal{O}\left(n\right) + \vartheta_{\text{QP}}(mN, \tilde{q}) \tag{12}$$

operations are necessary in total. We assume κ_∞ is available from the offline calculations.

lyap-MPC, höl-lyap-MPC: The criterion (7) must be evaluated for every constraint, for which the bound $\kappa_2(x)$ is needed. Simple calculations show that this involves $q\,\mathcal{O}\left(n\right)$ and $\mathcal{O}\left(n^2 + nmN\right)$ operations, respectively. Thus, in summary the calculation of the control law with lyap-MPC requires $q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right) + \vartheta_{\text{QP}}(mN, \tilde{q})$ arithmetic operations. Again, höl-lyap-MPC additionally performs the inclusion test (9) for every constraint, which amounts to an additional $q\,\mathcal{O}\left(n\right)$ operations. Therefore, a total of

$$2q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right) + \vartheta_{\text{QP}}(mN, \tilde{q}) \tag{13}$$

are required for höl-lyap-MPC.

The results stated in Sects. V-A and V-B are summarized in Tab. I.

C. Achievable reduction with constraint removal

We show that the effort of identifying inactive constraints is outweighed by the savings even in the worst case for both active-set and interior-point solvers in this section. This implies that a reduction of the required number of arithmetic operations results for all proposed methods.

From comparing (12) to (13) we infer that the worst-case effort to set up and solve $\tilde{\mathbb{P}}(x)$ is required by höl-lyap-MPC. Thus, we compare (13) to the number of arithmetic operations required by full-MPC, i.e., to solve the QP $\mathbb{P}(x)$ without any reduction. We analyze the achievable acceleration for active-set and interior-point algorithms separately.

Acceleration achieved for active-set algorithms

The number of arithmetic operations required to solve a QP with mN decision variables and q constraints with an active-set algorithm is given by

$$\vartheta_{\text{QP}}^{\text{AS}} = \left((mN + n\mathcal{Y})^3 + 2m^2N^2 + 4mNq\right)\rho^{\text{AS}}(mN, q), \tag{14}$$

where $\rho^{\text{AS}}(mN, q)$ is the number of iterations and $n\mathcal{Y}$ is the average number of constraints inside the working set [11]. Let $\varphi = \frac{\tilde{q}}{q}$, $\varphi \in [0, 1]$. Combining (13) with (14) yields the number of operations required to identify the inactive constraints and to solve the reduced optimization problem $\tilde{\mathbb{P}}(x)$ with an active-set algorithm, which is $\tilde{\vartheta}_{\text{QP}}^{\text{AS}} = \left((mN + n\mathcal{Y})^3 + 2m^2N^2 + 4mN\tilde{q}\right)\rho^{\text{AS}}(mN, \tilde{q}) + 2q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right)$, or, equivalently,

$$\begin{aligned}\tilde{\vartheta}_{\text{QP}}^{\text{AS}} = &\left((mN + n\mathcal{Y})^3 + 2m^2N^2 + 4mN\varphi q\right)\rho^{\text{AS}}(mN, \varphi q) \\ &+ 2q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right).\end{aligned} \tag{15}$$

We examine the reduction achieved by CR using the operation count ratio τ^{AS}, which we define as the quotient of the number of arithmetic operations required to solve $\tilde{\mathbb{P}}(x)$ and the number of arithmetic operations required to solve $\mathbb{P}(x)$, i.e., the ratio between $\tilde{\vartheta}_{\text{QP}}^{\text{AS}}$ and $\vartheta_{\text{QP}}^{\text{AS}}$. Dividing (15) by (14), yields

$$\begin{aligned}\tau^{\text{AS}} = &\frac{(m^2N + 2m) + \varphi 8(m+n)}{m^2N + 10m + 8n} \\ &+ \frac{\mathcal{O}\left(n^2N + nmN\right)}{(m^3N^2 + 10m^2N + 8mnN)\rho^{\text{AS}}(mN, q)},\end{aligned} \tag{16}$$

where we assumed that (a) $q = 2mN + 2nN$, which holds if both the states and the inputs are subject to lower and upper bounds, (b) $n\mathcal{Y} \ll mN$, and (c) $\rho^{\text{AS}}(mN, q) = \rho^{\text{AS}}(mN, \varphi q)$, i.e., the same number of iterations is required to solve $\tilde{\mathbb{P}}(x)$ and $\mathbb{P}(x)$. The last term in (16) vanishes for sufficiently large horizons $N > n$, since $m^3N^2\rho^{\text{AS}}(mN, q) \gg n^2N$. This yields

$$\tau^{\text{AS}} = \frac{(m^2N + 2m) + \varphi 8(m+n)}{m^2N + 10m + 8n}. \tag{17}$$

We analyze τ^{AS} by considering the cases $m = n$ and $m \ll n$ separately. Consider the case $m = n$ first. Equation (17) simplifies to

$$\tau_{m=n}^{\text{AS}} = \frac{mN + 2}{mN + 18} + \varphi\frac{16}{mN + 18}.$$

Note that $\tau_{m=n}^{\text{AS}} \approx 1$ if $mN \gg 18$ even for $\varphi \to 0$. Hence, we can conclude that the number of arithmetic operations required to solve $\tilde{\mathbb{P}}(x)$ and $\mathbb{P}(x)$ is approximately equal. Consider $m \ll n$ next. The operation count ratio (17) simplifies to

$$\tau_{m\ll n}^{\text{AS}} = \frac{m^2N}{m^2N + 8n} + \varphi\frac{8n}{m^2N + 8n}.$$

Again, we have $\tau_{m\ll n}^{\text{AS}} \approx 1$ if $m^2N \gg 8n$ even if $\varphi \to 0$, which implies again that the number of arithmetic operations required to solve $\tilde{\mathbb{P}}(x)$ and $\mathbb{P}(x)$ is approximately equal.

Acceleration achieved for interior-point algorithms

We discuss the reduction achievable if interior-point algorithms are used to solve the QPs online. The number of

TABLE I: Computational effort of the variants (11). We use $\bar{n} = mN + n - 1$, $\bar{q} = q - 1$, and $w^i(x^+) = E^i x^+ + w^i$ for short.

Name	Description	Inclusion test	Offline Effort	Online Effort
full-MPC	classical MPC without preprocessing	–	–	$\vartheta_{\text{QP}}(mN, q)$
prelyap-MPC	lower bounds on the cost function	$V^\star(x) < \sigma_i \Rightarrow i \in \mathcal{I}(x)$	$q\vartheta_{\text{QP}}(\bar{n}, \bar{q})$	$q + \vartheta_{\text{QP}}(mN, \tilde{q})$
höl-prelyap-MPC	prelyap-MPC together with κ_∞ and Lem. 4	$V^\star(x) < \sigma_i \Rightarrow i \in \mathcal{I}(x)$ $\left\|G^i\right\|_1 \kappa_\infty < w^i(x^+) \Rightarrow i \in \mathcal{I}(x)$	$q\vartheta_{\text{QP}}(\bar{n}, \bar{q}) + mN\vartheta_{\text{LP}}(mN + n, q)$	$q\,\mathcal{O}\left(n\right) + \vartheta_{\text{QP}}(mN, \tilde{q})$
lyap-MPC	online constraint removal	$\left\|G^i\right\|_2 \kappa_2(x) < w^i(x^+) \Rightarrow i \in \mathcal{I}(x)$	$\mathcal{O}\left(m^3N^3\right)$	$q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right) + \vartheta_{\text{QP}}(mN, \tilde{q})$
höl-lyap-MPC	lyap-MPC together with κ_∞ and Lem. 4	$\left\|G^i\right\|_2 \kappa_2(x) < w^i(x^+) \Rightarrow i \in \mathcal{I}(x)$ $\left\|G^i\right\|_1 \kappa_\infty < w^i(x^+) \Rightarrow i \in \mathcal{I}(x)$	$\mathcal{O}\left(m^3N^3\right) + mN\vartheta_{\text{LP}}(mN + n, q)$	$2q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right) + \vartheta_{\text{QP}}(mN, \tilde{q})$

arithmetic operations required by a typical interior-point algorithm to solve a strictly convex QP with mN decision variables and q constraints is

$$\vartheta^{\text{IP}}_{\text{QP}} = \left((mN)^3 + (mN)^2(2q + 2.5)\right)\rho^{\text{IP}}(mN, q), \quad (18)$$

where the unknown function $\rho^{\text{IP}}(mN, q)$ describes the required number of iterations [11]. In (18), we neglect linear and bilinear terms in mN and q for simplicity. Let $\varphi = \frac{\tilde{q}}{q}$ again. The number of arithmetic operations required to detect the inactive constraints and to set up and solve the reduced optimization problem $\tilde{\mathbb{P}}(x)$ with an interior-point solver is $\tilde{\vartheta}^{\text{IP}}_{\text{QP}} = ((mN)^3 + (mN)^2(2\tilde{q} + 2.5))\rho^{\text{IP}}(mN, \tilde{q}) + 2q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right)$, or equivalently

$$\begin{aligned}\tilde{\vartheta}^{\text{IP}}_{\text{QP}} = &\left((mN)^3 + (mN)^2(2\varphi q + 2.5)\right)\rho^{\text{IP}}(mN, \varphi q)\\ &+ 2q\,\mathcal{O}\left(n\right) + \mathcal{O}\left(n^2 + nmN\right),\end{aligned} \quad (19)$$

which follows from combining (18) with (13). We analyze the achieved reduction using the operation count ratio τ^{IP} again, which we define as the ratio between $\tilde{\vartheta}^{\text{IP}}_{\text{QP}}$ and $\vartheta^{\text{IP}}_{\text{QP}}$. Thus, dividing (19) by (18) yields

$$\begin{aligned}\tau^{\text{IP}} = &\frac{m}{(5m + 4n)} + \varphi\frac{4(m + n)}{(5m + 4n)}\\ &+ \frac{\mathcal{O}\left(n^2N + nmN\right)}{\left(m^3N^2 + 4m^2N^2(m + n)\right)\rho^{\text{IP}}(mN, q)},\end{aligned} \quad (20)$$

where we used $q = 2mN + 2nN$ as before and assumed (a) $m^3N^3 \gg 2.5m^2N^2$ and (b) $\rho^{\text{IP}}(mN, q) = \rho^{\text{IP}}(mN, \varphi q)$. Using $m^3N^2\rho^{\text{AS}}(mN, q) \gg n^2N$ again the last term in (20) vanishes, and we have

$$\tau^{\text{IP}} = \frac{m}{(5m + 4n)} + \varphi\frac{4(m + n)}{(5m + 4n)}. \quad (21)$$

We analyze the operation count ration τ^{IP} by considering the cases $m = n$ and $m \ll n$ separately. First, assume $m = n$. In this case, (21) simplifies to

$$\tau^{\text{IP}}_{m=n} = \frac{1}{9} + \frac{8}{9}\varphi.$$

Note that we have $\tau^{\text{IP}}_{m=n} \to \frac{1}{9}$ for $\varphi \to 0$. Thus, if nearly all constraints are removed by constraint removal, the number of arithmetic operations is reduced by the factor 9. Now assume $m \ll n$. In this case, the first term in (21) vanishes and the operation count ratio simplifies to

$$\tau^{\text{IP}}_{m \ll n} = \varphi.$$

This operation count ratio implies that the number of arithmetic operations decreases as constraints are removed from consideration.

It can be summarized that the effort of detecting a priori inactive constraints is outweighed by the savings for both active-set and interior-point solvers. However, the benefit of solving a reduced optimization problem is larger for interior-point solvers. Essentially, this results because interior-point solvers always involve *all* constraints in their calculations. Removing constraints from consideration has a large effect here, since the inequality constraints are treated as equality constraints with slack variables. In contrast, active-set solvers work only with a *subset* of the constraints (i.e., the candidate active set), and therefore may not consider an a priori inactive constraint, regardless of whether it is removed in advance or not.

VI. Example

We consider the state space system that results from discretizing the continuous-time transfer function

$$G(s) = \begin{pmatrix} \frac{-5s+1}{36s^2+6s+1} & \frac{0.5s}{8s+1} & 0\\ 0 & \frac{0.1(-10s+1)}{s(8s+1)} & \frac{-0.1}{(64s^2+6s+1)s}\\ \frac{-2s+1}{12s^2+3s+1} & 0 & \frac{2(-5s+1)}{16s^2+2s+1} \end{pmatrix}, \quad (22)$$

with ZOH and $T_s = 1$s. The state and input constraints read $-10 \leq x_i(k) \leq 10$ for $i = 1, \ldots, 10$ and $-1 \leq u_j(k) \leq 1$ for $j = 1, \ldots, 3$, respectively. The matrix P is set to the solution of the discrete-time algebraic Riccati equation, and the weighting matrices on the states and inputs are $Q = I^{n \times n}$ and $R = 0.25I^{m \times m}$, respectively. The horizon is chosen to be $N = 30$. We used the approach proposed in [12] to calculate the terminal set $\mathbb{X}_f$ with the Riccati controller[2]. The resulting QP has 898 constraints and 90 decision variables.

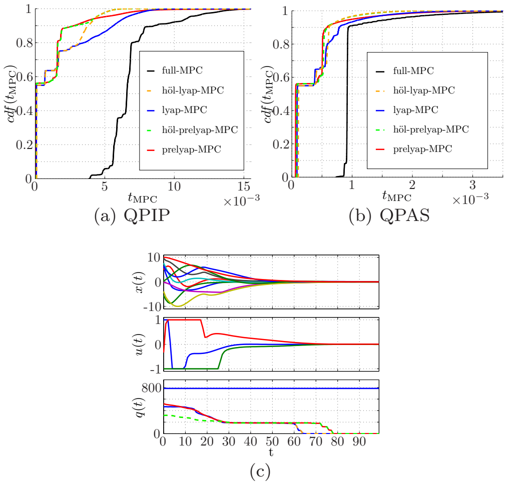
We chose 3405 random initial values and calculated trajectories for the closed-loop system until the state is sufficiently close to the origin. In total, 560513 QPs are solved. We measured the time required to calculate the optimal control law for classical MPC, i.e. MPC without preprocessing of its constraints (named full-MPC for convenience), and the constraint removal variants prelyap-MPC, höl-prelyap-MPC, lyap-MPC, and höl-lyap-MPC (see (11) for details). We solved the quadratic programs with the active-set solver QPAS and the interior-point solver QPIP from the QPC library [13][3].

The resulting computation times are given in Figure 1(a,b) as cumulative distribution functions $cdf(t_{\text{MPC}})$. The function $cdf(t_{\text{MPC}})$ is defined as the fraction of values that are equal or less than t_{MPC}.

Consider the results for the QPIP solver first, cf. Figure 1(a). About 55% of the QPs are found to be unconstrained by constraint removal. No optimization problem needs to be solved, but the unconstrained solution is applied directly.

[2] Note that with this choice of P and $\mathbb{X}_f$ the origin of the closed-loop system $x(k+1) = Ax + B\mu(x)$, where $\mu(x) := u_0^\star(x)$ is the MPC control law, is exponentially stable and the optimal value function $V^\star(x)$ is a Lyapunov function of the closed-loop system [9, Chp. 2.5.3.1, p. 142].

[3] All calculations were carried out in Matlab on an Intel i5-3570 CPU with 8Gb RAM running Suse Linux with Kernel 3.11.10-7.



(a) QPIP

(b) QPAS

(c)

Fig. 1: Results of the simulation study: cumulative distribution functions (cdf) of the step times achieved by prelyap-MPC (red, solid), höl-prelyap-MPC (green, dashed), lyap-MPC (blue, solid) and höl-lyap-MPC (orange, dashed) with the QPIP solver (a) and QPAS solver (b). (c) states $x(k)$ (upper), inputs $u(k)$ (middle) and number of constraints (bottom) of the reduced QPs for a trajectory. The black solid line shows the number of constraints of the full QP.

Since the control law can be calculated very quickly in these cases, the leftmost shoulders result for all constraint removal variants. Clearly, the cdfs resulting for all constraint removal variants lie to the left of that resulting for full-MPC (black curve). This implies a reduction of the average computation time.

Consider the cdf for lyap-MPC (blue, solid curve) and its extension höl-lyap-MPC (orange, dashed curve) next. Clearly, both cdfs coincide for $t_{\mathrm{MPC}} \leq 2.5 \cdot 10^{-3}$. This results because the same criterion is chosen, in other words the additional criterion from Lemma 4 together with the bound κ_∞ does not provide an improvement. However, for $t_{\mathrm{MPC}} > 2.5 \cdot 10^{-3}$ an improvement results with the additional criterion, and the cdf of höl-lyap-MPC lies to the left of that of lyap-MPC. Similar results are obtained for prelyap-MPC and its extension höl-prelyap-MPC.

The results are obtained for the QPAS solver are qualitatively similar to those obtained with the QPIP solver. We therefore omit a detailed discussion.

Consider Figure 1(c) next, which shows the states, the inputs and the number of constraints of the reduced quadratic programs. Clearly, the controller stabilizes the system and we have $\lim_{k\to\infty} x(k) = 0$ (upper plot). The number of constraints of the reduced QP are shown in the lowermost plot for all constraint removal variants. Consider lyap-MPC and its extension höl-lyap-MPC first (blue and orange curve, respectively). The number of constraints resulting for höl-lyap-MPC is smaller than for lyap-MPC for $k < 30$, and identical for $k \geq 30$. For $k < 30$ the additional criterion based on the bound κ_∞ reduces the number of constraints compared to lyap-MPC. Similar results are obtained for prelyap-MPC and höl-prelyap-MPC (red and green curve, respectively).

The average computation time that results for full-MPC in combination with the QPIP solver is 6.84ms. The reduction achieved with the proposed constraint removal approaches varies between 76% (achieved by lyap-MPC) and 84% (achieved by höl-prelyap-MPC). In contrast, the average computation time that results for full-MPC in combination with the QPAS solver is 1.01ms. Here, the achievable reduction varies between 63% (lyap-MPC) and 68% (höl-prelyap-MPC). The smaller reduction achieved with the QPAS solver corroborates the theoretical results obtained in Sec. V-C, i.e., constraint removal is more effective in combination with interior-point solvers.

VII. Conclusion

We proposed a new variant of constraint removal and discussed its properties. Furthermore, we compared the existing variants and the new variant with respect to their computational effort in detail for the first time.

Further research must address the extension of constraint removal to nonlinear MPC. Additionally, the robustness of constraint removal has only been addressed in the framework of tube-based MPC so far, whereas it is worth exploring how constraint removal for nominal MPC performs in closed loop with perturbed systems.

Acknowledgment

Support by the Deutsche Forschungsgemeinschaft (DFG) under grant MO 1086/11-1 is gratefully acknowledged. Furthermore, we would like to thank the reviewer of the paper [14] for his helpful comment on the generalization of Lemma 2 using Hölder's inequality.

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. M. Seron, J. A. De Dona, and G. C. Goodwin, "Global analytical model predictive control with input constraints," in *Proc. of the 39th IEEE CDC*, 2000, pp. 154 – 159.

[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] M. Jost and M. Mönnigmann, "Accelerating online MPC with partial explicit information and linear storage complexity in the number of constraints," in *Proc. of the ECC 2013*, Zurich, Switzerland, 2013, pp. 35–40.

[6] ——, "Accelerating model predictive control by online constraint removal," in *Proc. of the 52nd IEEE CDC*, Florence, Italy, 2013, pp. 5764–5769.

[7] M. Jost, G. Pannocchia, and M. Mönnigmann, "Online constraint removal: accelerating MPC with a Lyapunov function," *Automatica*, vol. 57, pp. 164–169, 2015.

[8] ——, "Accelerating linear model predictive control by constraint removal," *submitted*, 2014.

[9] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill Publishing, 2009.

[10] M. Jost, G. Pannocchia, and M. Mönnigmann, "Assessing the speed-up achievable by online constraint removal in MPC," in *Proc. of the ECC 2015*, Linz, Austria, 2015, pp. 3439–3444.

[11] M. S. K. Lau, S. P. Yue, K. V. Ling, and J. M. Maciejowski, "A Comparison of Interior-point and active-set methods for FPGA implementation of Model Predictive Control," in *Proc. of the ECC 2009*, Budapest, Hungary, 2009, pp. 156 – 161.

[12] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets,"*IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[13] D. A. Wills, *QPC - Quadratic Programming in C*, School of Electrical Engineering and Computer Science, University of Newcastle, 2009-08-11. Available: http://sigpromu.org/quadprog/index.html

[14] M. Jost, G. Pannocchia, and M. Mönnigmann, "Fast calculation of tube-based model predictive control laws for constrained linear systems with bounded disturbance." in *Proc. of the 54nd IEEE CDC*, Osaka, Japan, 2015, pp. 3651–3656.