

Efficient computation of Lipschitz constants for MPC with symmetries

Dieter Teichrib and Moritz Schulze Darup

Abstract—Lipschitz constants for linear MPC are useful for certifying inherent robustness against unmodeled disturbances or robustness for neural network-based approximations of the control law. In both cases, knowing the minimum Lipschitz constant leads to less conservative certifications. Computing this minimum Lipschitz constant is trivial given the explicit MPC. However, the computation of the explicit MPC may be intractable for complex systems. The paper discusses a method for efficiently computing the minimum Lipschitz constant without using the explicit control law. The proposed method simplifies a recently presented mixed-integer linear program (MILP) that computes the minimum Lipschitz constant. The simplification is obtained by exploiting saturation and symmetries of the control law and irrelevant constraints of the optimal control problem.

Index Terms—Model Predictive Control, Optimization, Mixed-Integer Programming, Computational methods

I. INTRODUCTION AND PROBLEM STATEMENT

Model predictive control (MPC) is an established method for the performance-oriented control of dynamical systems subject to state and input constraints (see [1] for an overview). For various applications or realizations of MPC, knowing a Lipschitz constant of the control law is beneficial. For instance, Lipschitz constants can be used to certify inherent robustness of classical MPC against unmodeled disturbances [2], [3]. Based on similar concepts, Lipschitz constants can be used to certify robustness for neural network-based approximations of MPC laws [4], [5].

As pointed out in [6], computing a Lipschitz constant for linear MPC is trivial, if the (piecewise affine) control law is explicitly known. However, the number of affine segments may grow exponentially with the state dimension and the number of constraints in the optimal control problem (OCP). Thus, explicitly computing the control law often becomes intractable for complex systems. This motivates the design of methods that allow the computation of a Lipschitz constant without explicitly computing the control law. Such methods have been previously addressed, e.g. in [4], [6]. In [6], a procedure enumerating potential active sets of the OCP is proposed. However, the procedure is inefficient and, in parts, based on an unproven conjecture (see [6, Conj. 4]). In [4], the problem of finding the smallest Lipschitz constant for MPC is formulated as a mixed-integer linear program (MILP). The procedure is elegant and it can be applied in case where the computation of the explicit control law is numerically intractable (cf. [4, Tab. I]). Still, solving the resulting MILP may be time-consuming since, typically, many binary variables are involved.

D. Teichrib and M. Schulze Darup are with the Control and Cyber-physical Systems Group, Department of Mechanical Engineering, TU Dortmund University, Germany. E-mails: dieter.teichrib@tu-dortmund.de, moritz.schulzedarup@tu-dortmund.de.

In this paper, we aim for a more efficient computation of Lipschitz constants via MILP. To this end, we first slightly improve the MILP formulation from [4] by reducing the initial number of binary variables. Afterwards, we present various preprocessing steps to further reduce the number of binary variables in the MILP. In this context, the most powerful reduction step builds on symmetries that often arise in MPC (see, e.g., [7]). Furthermore, the observation that the control law is constant in many regions for a large prediction horizon also allows a significant reduction in the number of binary variables.

The paper is organized as follows. The notation is given in the remainder of this section. Section II is devoted to the basics of MPC and presents a known representation of the OCP in terms of an MILP, which can be used to compute the minimum Lipschitz constant of the control law. In Section III, we describe several methods for reducing the number of binary variables needed to compute the minimum Lipschitz constant via MILP. We illustrate the effectiveness of the proposed method with some examples from the literature in Section IV and give a conclusion in Section V.

Notation. For matrices $\mathbf{K} \in \mathbb{R}^{m \times n}$, we denote its i -th row by $\mathbf{K}_{i,:}$, its j -th column by $\mathbf{K}_{:,j}$, and its elements by $K_{i,j}$, respectively. We further use the shorthand notation \mathbf{K}_i for $\mathbf{K}_{i,:}$ and we indicate different instances of a matrix \mathbf{K} by a superscript, e.g., $\mathbf{K}^{(i)}$. The product

$$\mathbf{K}\mathcal{X} := \{\mathbf{K}\mathbf{x} \in \mathbb{R}^m \mid \mathbf{x} \in \mathcal{X}\}$$

is defined for a compact and convex set \mathcal{X} . For a vector $\mathbf{v} \in \mathbb{R}^m$ we define $\text{diag}(\mathbf{v})$ as a diagonal matrix with the elements of \mathbf{v} being the diagonal elements. Moreover, \mathbf{I} refers to the identity matrix and we denote column vectors or matrices whose entries are all 0 respectively 1 by $\mathbf{0}$ respectively $\mathbf{1}$. Finally, all inequalities involving matrices or vectors are understood element-wise.

II. PRELIMINARIES

A. Lipschitz constants of piecewise affine functions

In general a Lipschitz constant of a function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ on a domain $\mathcal{F} \subseteq \mathbb{R}^n$ is a constant L satisfying

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\tilde{\mathbf{x}})\| \leq L\|\mathbf{x} - \tilde{\mathbf{x}}\|$$

for all $\mathbf{x}, \tilde{\mathbf{x}} \in \mathcal{F}$ and some vector norm $\|\cdot\|$. We here focus on p -norms and, in particular, the cases $p \in \{1, \infty\}$. We denote corresponding Lipschitz constants by L_p . Further, we aim for as small as possible Lipschitz constants and denote the smallest one by L_p^* . Now, under the assumption that \mathbf{f}

is a continuous piecewise affine (PWA) function on \mathcal{F} , i.e., of the form

$$\mathbf{f}(\mathbf{x}) = \begin{cases} \mathbf{K}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} & \text{if } \mathbf{x} \in \mathcal{R}^{(1)}, \\ \vdots \\ \mathbf{K}^{(s)}\mathbf{x} + \mathbf{b}^{(s)} & \text{if } \mathbf{x} \in \mathcal{R}^{(s)}, \end{cases} \quad (1)$$

with $\mathbf{K}^{(i)} \in \mathbb{R}^{m \times n}$ and $\mathbf{b}^{(i)} \in \mathbb{R}^m$, then it is well-known that

$$L_p^* = \max_{i \in \{1, \dots, s\}} \|\mathbf{K}^{(i)}\|_p, \quad (2)$$

where $\|\cdot\|_p$ here refers to the matrix norm induced by the vector p -norm (see, e.g., [8, Prop. 3.4]). As pointed out in [4, Lem. 5.1], the matrix norms $\|\mathbf{K}\|_1$ and $\|\mathbf{K}\|_\infty$ can be evaluated by solving a linear program (LP). In fact, one finds

$$\|\mathbf{K}\|_1 = \min_{l, \mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}} l \quad (3)$$

s.t. $\mathbf{1}^\top \mathbf{v}^{(j)} \leq l, \quad -\mathbf{v}^{(j)} \leq \mathbf{K}_{:,j} \leq \mathbf{v}^{(j)}, \quad \forall j \in \{1, \dots, n\}.$

Due to $\|\mathbf{K}\|_\infty = \|\mathbf{K}^\top\|_1$, the LP for computing the ∞ -norm can be formulated in a similar way.

B. Linear MPC via MILP

MPC for linear discrete-time systems builds on solving an OCP of the form

$$V(\mathbf{x}) := \min_{\substack{\mathbf{x}(0), \dots, \mathbf{x}(N) \\ \mathbf{u}(0), \dots, \mathbf{u}(N-1)}} \varphi(\mathbf{x}(N)) + \sum_{k=0}^{N-1} \ell(\mathbf{x}(k), \mathbf{u}(k)) \quad (4)$$

s.t. $\mathbf{x}(0) = \mathbf{x},$
 $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \quad \forall k \in \{0, \dots, N-1\},$
 $(\mathbf{x}(k), \mathbf{u}(k)) \in \mathcal{X} \times \mathcal{U}, \quad \forall k \in \{0, \dots, N-1\},$
 $\mathbf{x}(N) \in \mathcal{T}$

in every time step for the current state \mathbf{x} . Here, $N \in \mathbb{N}$ refers to the prediction horizon and

$$\varphi(\mathbf{x}) := \mathbf{x}^\top \mathbf{P}\mathbf{x} \quad \text{and} \quad \ell(\mathbf{x}, \mathbf{u}) := \mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u}$$

denote the terminal and stage cost, respectively, where we assume that the weighting matrices \mathbf{P} , \mathbf{Q} , and \mathbf{R} are positive definite. The dynamics of the linear prediction model are described by $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$. State and input constraints can be incorporated via the polyhedral sets \mathcal{X} and \mathcal{U} . Finally, a polyhedral terminal set \mathcal{T} allows to enforce closed-loop stability (see [9] for details). Condensing the OCP (4) leads to a parametric quadratic program (QP) of the form

$$U^*(\mathbf{x}) := \arg \min_U \frac{1}{2} \mathbf{U}^\top \mathbf{H}\mathbf{U} + \mathbf{x}^\top \mathbf{F}^\top \mathbf{U} \quad (5)$$

s.t. $\mathbf{G}\mathbf{U} \leq \mathbf{E}\mathbf{x} + \mathbf{d}$

with the decision variable $\mathbf{U} \in \mathbb{R}^{mN}$ reflecting the predicted input sequence (i.e., the stacked vectors $\mathbf{u}(0), \dots, \mathbf{u}(N-1)$) and with $\mathbf{H}, \mathbf{F}, \mathbf{G}, \mathbf{E}, \mathbf{d}$ denoting condensed matrices obtained from $\mathbf{A}, \mathbf{B}, \mathbf{P}, \mathbf{Q}, \mathbf{R}$ and the specifications of $\mathcal{X}, \mathcal{U}, \mathcal{T}$ (see [10, Chap. 3] for details). Now, MPC typically builds on applying the first element of the optimal input sequence, i.e., $\mathbf{u}^*(0)$, and repeating the procedure at the next sampling

instant. Hence, the resulting control law $\mathbf{f}_{\text{MPC}} : \mathcal{F}_{\text{MPC}} \rightarrow \mathcal{U}$ can be defined as

$$\mathbf{f}_{\text{MPC}}(\mathbf{x}) := \mathbf{S}\mathbf{U}^*(\mathbf{x}), \quad (6)$$

where $\mathbf{S} := (\mathbf{I} \ \mathbf{0} \ \dots \ \mathbf{0}) \in \mathbb{R}^{m \times mN}$ serves as a selection matrix and where the set \mathcal{F}_{MPC} contains all $\mathbf{x} \in \mathbb{R}^n$ for which (5) (or, equivalently, (4)) is feasible. Remarkably, \mathbf{f}_{MPC} is of the form (1) with regions $\mathcal{R}^{(i)}$ representing polyhedral sets with pairwise disjoint interiors [11, Thm. 4].

Clearly, the solution of (5) can be characterized by the Karush-Kuhn-Tucker (KKT) conditions

$$\mathbf{H}\mathbf{U}^*(\mathbf{x}) + \mathbf{F}\mathbf{x} + \mathbf{G}^\top \boldsymbol{\lambda}^*(\mathbf{x}) = \mathbf{0}, \quad (7a)$$

$$\mathbf{r}^*(\mathbf{x}) = \mathbf{E}\mathbf{x} + \mathbf{d} - \mathbf{G}\mathbf{U}^*(\mathbf{x}), \quad (7b)$$

$$\mathbf{r}^*(\mathbf{x}) \geq \mathbf{0}, \quad (7c)$$

$$\boldsymbol{\lambda}^*(\mathbf{x}) \geq \mathbf{0}, \quad (7d)$$

$$\text{diag}(\boldsymbol{\lambda}^*(\mathbf{x})) \mathbf{r}^*(\mathbf{x}) = \mathbf{0} \quad (7e)$$

where $q \in \mathbb{N}$ reflects the dimension of \mathbf{d} , i.e., the number of constraints. Now, let us assume upper bounds $\bar{\boldsymbol{\lambda}}$ and $\bar{\mathbf{r}}$ for the dual optimizers $\boldsymbol{\lambda}^*(\mathbf{x})$ respectively the residuals $\mathbf{r}^*(\mathbf{x})$ are known, i.e., $\boldsymbol{\lambda}^*(\mathbf{x}) \leq \bar{\boldsymbol{\lambda}}$ and $\mathbf{r}^*(\mathbf{x}) \leq \bar{\mathbf{r}}$ for all $\mathbf{x} \in \mathcal{F}_{\text{MPC}}$. Since (5) is feasible for all $\mathbf{x} \in \mathcal{F}_{\text{MPC}}$ by definition, such upper bounds exist and are finite. Then, the conditions (7c)–(7e) can be rewritten as

$$0 \leq \mathbf{r}^*(\mathbf{x}) \leq \text{diag}(\bar{\mathbf{r}})(\mathbf{1} - \boldsymbol{\delta}^*(\mathbf{x})), \quad (8a)$$

$$0 \leq \boldsymbol{\lambda}^*(\mathbf{x}) \leq \text{diag}(\bar{\boldsymbol{\lambda}}) \boldsymbol{\delta}^*(\mathbf{x}), \quad (8b)$$

$$\boldsymbol{\delta}^*(\mathbf{x}) \in \{0, 1\}^q \quad (8c)$$

[4, Eq. (13)]. Thus, the QP (5) can be solved by solving the MI feasibility problem

$$\text{find } \mathbf{U}^*, \boldsymbol{\lambda}^*, \mathbf{r}^*, \text{ and } \boldsymbol{\delta}^* \quad \text{s.t.} \quad (7a)–(7b) \text{ and } (8), \quad (9)$$

where we omit dependencies on \mathbf{x} for brevity.

C. Local MPC gain via MILP

Given a solution to (9), it is easy to see that the set

$$\mathcal{A}(\mathbf{x}) := \{i \in \{1, \dots, q\} \mid \delta_i^*(\mathbf{x}) = 1\}$$

reflects the active constraints for (5). In principle, $\mathcal{A}(\mathbf{x})$ allows to compute the local MPC segment, i.e., \mathbf{K}^* , \mathbf{b}^* , and \mathcal{R}^* such that $\mathbf{x} \in \mathcal{R}^*$ and

$$\mathbf{f}_{\text{MPC}}(\tilde{\mathbf{x}}) = \mathbf{K}^* \tilde{\mathbf{x}} + \mathbf{b}^* \quad \forall \tilde{\mathbf{x}} \in \mathcal{R}^*.$$

However, performing the computation analogously to [11] leads to nonlinear relations between \mathbf{K}^* and $\boldsymbol{\delta}^*(\mathbf{x})$. Hence, aiming for a combination of (2), (3) and (9) in one MILP, we need to derive \mathbf{K}^* differently. A suitable approach has been proposed in the proof of [4, Thm. 5.3]. The underlying idea is to sample the MPC segment (potentially even outside its domain \mathcal{R}^*) and to use the samples to characterize \mathbf{K}^* . The sampling points are chosen as

$$\mathbf{x}^{(j)} := \mathbf{x} + \mathbf{e}^{(j)} \quad \forall j \in \{1, \dots, n\},$$

where $\mathbf{e}^{(j)}$ refers to the j -th canonical unit vector. One then constructs corresponding $\mathbf{U}^{(j)}$ from the MPC segment

around \mathbf{x} by introducing the additional variables $\mathbf{U}^{(j)}$, $\boldsymbol{\lambda}^{(j)}$, and $\mathbf{r}^{(j)}$ and by augmenting the constraints (7a)–(7b) and (8) with

$$\mathbf{H}\mathbf{U}^{(j)} + \mathbf{F}(\mathbf{x} + \mathbf{e}^{(j)}) + \mathbf{G}^\top \boldsymbol{\lambda}^{(j)} = \mathbf{0} \quad (10a)$$

$$\mathbf{E}(\mathbf{x} + \mathbf{e}^{(j)}) + \mathbf{d} - \mathbf{G}\mathbf{U}^{(j)} = \mathbf{r}^{(j)} \quad (10b)$$

$$-M(\mathbf{1} - \boldsymbol{\delta}^*(\mathbf{x})) \leq \mathbf{r}^{(j)} \leq M(\mathbf{1} - \boldsymbol{\delta}^*(\mathbf{x})) \quad (10c)$$

$$-M\boldsymbol{\delta}^*(\mathbf{x}) \leq \boldsymbol{\lambda}^{(j)} \leq M\boldsymbol{\delta}^*(\mathbf{x}) \quad (10d)$$

for all $j \in \{1, \dots, n\}$, where M is a sufficiently large number such that feasible solutions satisfy $\mathbf{r}_i^{(j)}, \boldsymbol{\lambda}_i^{(j)} \in (-M, M)$. Then, we obviously have $\mathbf{S}(\mathbf{U}^{(j)} - \mathbf{U}^*(\mathbf{x})) = \mathbf{K}^* \mathbf{e}^{(j)}$ and consequently

$$\mathbf{K}^* = \mathbf{S}(\mathbf{U}^{(1)} - \mathbf{U}^*(\mathbf{x}) \quad \dots \quad \mathbf{U}^{(n)} - \mathbf{U}^*(\mathbf{x})). \quad (11)$$

We could now compute $\|\mathbf{K}^*\|_1$ by solving (3) for \mathbf{K}^* as in (11) subject to the additional constraints (7a)–(7b), (8), and (10), which results in an MILP. However, including a maximization analogously to (2) (here over $\mathbf{x} \in \mathcal{X}$) is non-trivial since (3) calls for a minimization. To circumvent this issue, one can consider the dual of (3) and observe that the corresponding optimizer is binary [4, Lem. 5.1]. This finally allows to compute the Lipschitz constant L_1^* for an MPC scheme based on an MILP with

$$q + (2m + 1)n \quad (12)$$

binary variables (or $q + (2n + 1)m$ for L_∞^*) [4].

III. REDUCING BINARY VARIABLES FOR EFFICIENCY

The numerical complexity for solving an MILP crucially depends on the number of binary variables. Hence, in order to enable more efficient Lipschitz constant computations for MPC, we aim for a reduction of the number of binary variables in the corresponding MILP. In this context, we first stress that the number in (12) consists of two terms resulting from the q constraints of the MPC-related QP in (5) and the $(2m + 1)n$ constraints in (3). Hence, we have two immediate options for reducing the number of binary variables: First, reducing the number of constraints in (5) relevant for the computation of Lipschitz constants and, second, implementing the norm evaluation more efficiently. We investigate these options in Sections III-A and III-B, respectively. Finally, we show in Section III-C that exploiting symmetries, which are often present in MPC, can be beneficial for the computation of Lipschitz constants.

A. More efficient norm computation

Roughly speaking, the computation of Lipschitz constants proposed in [4] builds on reformulations of the QP (5) and the LP (3) in terms of MI feasibility problems. In [4], the reformulation of (3) is realized based on the LP's dual. Next, we propose a direct reformulation based on the primal LP, which is inspired by MI modeling techniques from, e.g., [12] and which requires fewer binary variables than [4].

Lemma 1: Let $\mathbf{K} \in \mathbb{R}^{m \times n}$ and consider the conditions

$$\mathbf{K} = \mathbf{K}^+ - \mathbf{K}^-, \quad (13a)$$

$$\mathbf{0} \leq \mathbf{K}^+ \leq M(\boldsymbol{\delta}^{(1)} \quad \dots \quad \boldsymbol{\delta}^{(m)})^\top, \quad (13b)$$

$$\mathbf{0} \leq \mathbf{K}^- \leq M(\mathbf{1} - (\boldsymbol{\delta}^{(1)} \quad \dots \quad \boldsymbol{\delta}^{(m)})^\top), \quad (13c)$$

$$\mathbf{c}^\top = \mathbf{1}^\top(\mathbf{K}^+ + \mathbf{K}^-), \quad (13d)$$

$$\mathbf{c} \leq \mathbf{1}l \leq \mathbf{c} + M(\mathbf{1} - \boldsymbol{\delta}^{(m+1)}), \quad (13e)$$

$$\mathbf{1}^\top \boldsymbol{\delta}^{(m+1)} = 1, \quad (13f)$$

$$\boldsymbol{\delta}^{(1)}, \dots, \boldsymbol{\delta}^{(m+1)} \in \{0, 1\}^n \quad (13g)$$

for some M being larger than the largest absolute value of the entries in \mathbf{K} . Then, any solution to

$$\text{find } l, \mathbf{K}^+, \mathbf{K}^-, \mathbf{c}, \boldsymbol{\delta}^{(1)}, \dots, \boldsymbol{\delta}^{(m+1)} \quad \text{s.t.} \quad (13) \quad (14)$$

is such that $\|\mathbf{K}\|_1 = l$.

Proof: It is easy to see that the conditions (13a)–(13c) together with $\boldsymbol{\delta}^{(1)}, \dots, \boldsymbol{\delta}^{(m)} \in \{0, 1\}^n$ imply

$$(\mathbf{K}_{i,j}^+, \mathbf{K}_{i,j}^-) := \begin{cases} (\mathbf{K}_{i,j}, 0) & \text{if } \mathbf{K}_{i,j} \geq 0, \\ (0, -\mathbf{K}_{i,j}) & \text{otherwise.} \end{cases}$$

As a consequence, the entries of \mathbf{c} as in (13d) reflect the absolute column sums of \mathbf{K} . Due to (13e), (13f), and $\boldsymbol{\delta}^{(m+1)} \in \{0, 1\}^n$, l equals the largest entry of \mathbf{c} , which is, by definition, identical to $\|\mathbf{K}\|_1$. ■

Obviously, the MI feasibility problem (14) only requires $(m + 1)n$ binary variables and thus mn less than the counterpart in [4]. Again, a similar problem can easily be constructed to compute $\|\mathbf{K}\|_\infty$ using $(n + 1)m$ binary variables. Likewise, the reduction compared to [4] amounts to mn .

B. Excluding MPC constraints

We propose two approaches to reduce the number of constraints in (5) and consequently the number of binary variables in (9). The approaches are conceptually decoupled but it will turn out that their implementation can be efficiently coupled. The first approach builds on the straightforward observation that the i -th constraint in (5) (i.e., $\mathbf{G}_i \mathbf{U} \leq \mathbf{E}_i \mathbf{x} + \mathbf{d}_i$) is irrelevant for the MPC scheme, if there exists no $\mathbf{x} \in \mathcal{F}_{\text{MPC}}$ such that $\mathbf{G}_i \mathbf{U}^*(\mathbf{x}) = \mathbf{E}_i \mathbf{x} + \mathbf{d}_i$. According to the following lemma, such a situation can be identified based on an MI feasibility problem similar to (9).

Lemma 2: Let $i \in \{1, \dots, q\}$. If the MI feasibility problem (9) with the additional decision variable \mathbf{x} and with the additional constraint $\boldsymbol{\delta}_i^* = 1$ is infeasible, then

$$\mathbf{G}_i \mathbf{U}^*(\mathbf{x}) - \mathbf{E}_i \mathbf{x} < \mathbf{d}_i \quad \text{for all } \mathbf{x} \in \mathcal{F}_{\text{MPC}}.$$

Proof: We first note that the unmodified problem (9) has, by construction, the same feasible set \mathcal{F}_{MPC} as (5) with respect to the parameter \mathbf{x} . Now, the additional constraint $\boldsymbol{\delta}_i^* = 1$ implies $\mathbf{r}_i^* = 0$ in (8a) and consequently $\mathbf{G}_i \mathbf{U}^* = \mathbf{E}_i \mathbf{x} + \mathbf{d}_i$ in (7b). Hence, infeasibility of the MI feasibility problem in the claim with \mathbf{x} as a decision variable immediately implies that there exists no $\mathbf{x} \in \mathcal{F}_{\text{MPC}}$ such that $\mathbf{G}_i \mathbf{U}^*(\mathbf{x}) - \mathbf{E}_i \mathbf{x} = \mathbf{d}_i$. In other words, solving (5) for any feasible $\mathbf{x} \in \mathcal{F}_{\text{MPC}}$ results in $\mathbf{G}_i \mathbf{U}^*(\mathbf{x}) - \mathbf{E}_i \mathbf{x} < \mathbf{d}_i$. ■

Clearly, with the help of Lemma 2, we can eliminate irrelevant constraints by checking the corresponding MI feasibility problem for every (or some) $i \in \{1, \dots, q\}$. At this point, it might seem counterintuitive to consider multiple MI problems in order to simply the overlaying MILP of interest. However, our numerical benchmark in Section IV clearly shows that this approach is meaningful and that the overall runtime can be (significantly) shortened compared to a direct solution of the unmodified MILP for the computation of Lipschitz constants. Remarkably, Lemma 2 could also be used to simplify an MPC scheme offline in order to accelerate the QP solutions online.

The second approach for reducing binary variables associated to constraints differs from the first one in that it is tailored to the problem at hand. It is based on the observation that the MPC law of the form (1) often contains many segments with $\mathbf{K}^{(i)} = \mathbf{0}$ resulting in constant inputs determined by the bias term $\mathbf{b}^{(i)}$. In particular, this situation often arises if box-shaped input constraints are present. Clearly, due to $\|\mathbf{0}\|_p = 0$, the constant segments are irrelevant for the computation of Lipschitz constants (but they obviously matter for the MPC scheme). Now, according to the following theorem, our procedure to identify and exclude some of these segments is similar to the first approach.

Theorem 3: Let $i \in \{1, \dots, q\}$ and let \mathcal{F}_{MPC} be full-dimensional. Consider the MILP

$$\max_{\substack{U^*, U^{(1)}, \dots, U^{(m)}, \mathbf{x}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}, \\ \boldsymbol{\lambda}^*, \boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(m)}, \mathbf{r}^*, \mathbf{r}^{(1)}, \dots, \mathbf{r}^{(m)}, \boldsymbol{\delta}^*}} \sum_{j=1}^m U_j^{(j)} - U_j^* \quad (15)$$

subject to (7a)–(7b), (8), $\delta_i^* = 1$, and

$$\mathbf{H}U^{(j)} + \mathbf{F}\mathbf{x}^{(j)} + \mathbf{G}^\top \boldsymbol{\lambda}^{(j)} = \mathbf{0}, \quad (16a)$$

$$\mathbf{E}\mathbf{x}^{(j)} + \mathbf{d} - \mathbf{G}U^{(j)} = \mathbf{r}^{(j)}, \quad (16b)$$

$$\mathbf{0} \leq \mathbf{r}^{(j)} \leq \text{diag}(\bar{\mathbf{r}})(\mathbf{1} - \boldsymbol{\delta}^*), \quad (16c)$$

$$\mathbf{0} \leq \boldsymbol{\lambda}^{(j)} \leq \text{diag}(\bar{\boldsymbol{\lambda}}) \boldsymbol{\delta}^* \quad (16d)$$

for every $j \in \{1, \dots, m\}$. If the MILP is feasible and returns 0 as the optimal objective function value, then the i -th constraint can be omitted for the computation of the Lipschitz constant.

Proof: We initially neglect the objective function in (15) and investigate the corresponding MI feasibility problem. We further assume feasibility since the theorem is irrelevant otherwise. Now, we consider any feasible set of decision variables and note that, based on the corresponding U^* , \mathbf{x} , $\boldsymbol{\lambda}^*$, \mathbf{r}^* , and $\boldsymbol{\delta}^*$, we can construct another set of feasible variables by choosing $U^{(j)} := U^*$, $\mathbf{x}^{(j)} := \mathbf{x}$, $\boldsymbol{\lambda}^{(j)} := \boldsymbol{\lambda}^*$, and $\mathbf{r}^{(j)} := \mathbf{r}^*$ for every $j \in \{1, \dots, m\}$ (and keeping the other variables). Clearly, the associated objective function value is 0. In other words, given feasibility, the optimal value of the MILP is always non-negative. We next show that the optimal value is always positive if a feasible \mathbf{x} exists, for which the corresponding $\mathcal{A}(\mathbf{x})$ (containing i by construction) leads to $\mathbf{K}^* \neq \mathbf{0}$ and a full-dimensional \mathcal{R}^* . To this end, we recall that $\mathcal{A}(\mathbf{x})$ is determined by $\boldsymbol{\delta}^*$. We further note that the additional constraints (16) imply $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)} \in \mathcal{R}^*$

(analogously to $\mathbf{x} \in \mathcal{R}^*$). Hence, we find

$$U_j^{(j)} - U_j^* = \mathbf{K}_j^* \mathbf{x}^{(j)} + \mathbf{b}_j^* - \mathbf{K}_j^* \mathbf{x} - \mathbf{b}_j^* = \mathbf{K}_j^* (\mathbf{x}^{(j)} - \mathbf{x})$$

for every $j \in \{1, \dots, m\}$. Now, $\mathbf{K}^* \neq \mathbf{0}$ implies $\mathbf{K}_j^* \neq \mathbf{0}$ for at least one j . Due to \mathcal{R}^* being full-dimensional, there exist $\mathbf{x}^{(j)}, \mathbf{x} \in \mathcal{R}^*$ yielding a positive $U_j^{(j)} - U_j^*$. As a consequence, the optimal value in (15) will be positive since the other terms in the cost function have already been shown to be non-negative (due to the m independent $\mathbf{x}^{(j)}$). Conversely, if the MILP returns 0 as an optimal value, the active sets $\mathcal{A}(\mathbf{x})$ associated with feasible \mathbf{x} either correspond to $\mathbf{K}^* = \mathbf{0}$, lower dimensional \mathcal{R}^* , or both. Now, the former and the latter case are clearly irrelevant for computing the Lipschitz constant. However, also the remaining case $\mathbf{K}^* \neq \mathbf{0}$ on some lower dimensional domain \mathcal{R}^* is irrelevant if \mathcal{F}_{MPC} is full-dimensional (as assumed). In fact, due to continuity of \mathbf{f}_{MPC} [11, Thm. 4], the relevant gain \mathbf{K}^* will then be captured by some neighboring segment on a full-dimensional domain. ■

Theorem 3 provides another condition to potentially exclude the i -th constraint. Remarkably, the two conditions in the previous theorem and Lemma 2, while conceptually different, are methodically closely related. To see this, note that the constraints of the corresponding MI problems only differ in terms of (16). Now, it is easy to see that the constraints (16) are feasible whenever the corresponding constraints (7a)–(7b) and (8) are feasible for \mathbf{x} . Hence, we immediately find the following relation.

Corollary 4: The MI feasibility problem in Lemma 2 is feasible if and only if the MILP in Theorem 3 is feasible.

As a consequence, one can only investigate the MILP in Theorem 3 and exclude the i -th constraint if the MILP is either infeasible or returns 0.

C. Exploiting symmetries in MPC

Due to common symmetries in the constraints or the cost function, MPC often results in control laws, which likewise offer symmetries. Formalizing these symmetries can, e.g., be carried out analogously to [7, Def. 1]. There, a symmetry is expressed in terms of invertible matrices $(\boldsymbol{\Theta}, \boldsymbol{\Omega})$ satisfying

$$\boldsymbol{\Omega} \mathbf{f}_{\text{MPC}}(\mathbf{x}) = \mathbf{f}_{\text{MPC}}(\boldsymbol{\Theta} \mathbf{x}) \quad (17)$$

for every $\mathbf{x} \in \mathcal{F}_{\text{MPC}}$. Exploiting symmetries is, e.g., useful in the framework of explicit MPC [11] since it allows to reduce the domain for which the explicit control law has to be computed (and stored). To specify this, we first note that multiple symmetries in terms of tuples $(\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Omega}^{(1)}), \dots, (\boldsymbol{\Theta}^{(\sigma)}, \boldsymbol{\Omega}^{(\sigma)})$ can apply simultaneously with the canonical tuple $(\mathbf{I}_m, \mathbf{I}_n)$ being one of those. Then, we can substitute the constraint $\mathbf{x}(0) \in \mathcal{X}$ in (4) with $\mathbf{x}(0) \in \mathcal{X}_{\text{fun}}$ for any choice of (the so-called fundamental domain) $\mathcal{X}_{\text{fun}} \subseteq \mathcal{X}$ satisfying

$$\mathcal{F}_{\text{MPC}} \subseteq \bigcup_{i=1}^{\sigma} \boldsymbol{\Theta}^{(i)} \mathcal{X}_{\text{fun}}.$$

Clearly, in order to still enable the condensation to (5), it additionally makes sense to restrict ourselves to polyhedral

sets \mathcal{X}_{fun} . Now, assuming for a moment that \mathcal{X} and \mathcal{X}_{fun} are characterized by the same number of hyperplanes. Then, it is easy to see that the substitution above does not alter the number of constraints in (5). Hence, it is not immediately clear how exploiting symmetries can be beneficial for our purposes. In this context, we first note that substituting \mathcal{X} with a significantly smaller set \mathcal{X}_{fun} (for the constraint associated with $\mathbf{x}(0)$) often results in significantly more excluded constraints by the procedures related to Lemma 2 and Theorem 3 (see our benchmark in Sect. IV). Moreover, symmetries often yield relations like $\mathbf{K}^{(i)} = -\mathbf{K}^{(j)}$ implying $\|\mathbf{K}^{(i)}\|_p = \|\mathbf{K}^{(j)}\|_p$. However, symmetries do not always result in such trivial relations, in particular, in light of norms. To see this, note that (17) in combination with the structure (1) provides relations like

$$\mathbf{K}^{(i)}\mathbf{x} + \mathbf{b}^{(i)} = \Omega^{-1}\mathbf{K}^{(j)}\Theta\mathbf{x} + \Omega^{-1}\mathbf{b}^{(j)}.$$

Hence, instead of evaluating f_{MPC} for some $\mathbf{x} \in \mathcal{R}^{(i)}$, we could also make use of segment j containing $\Theta\mathbf{x}$. This would allow us to skip segment i in the context of the Lipschitz constant computation and to consider $\|\Omega^{-1}\mathbf{K}^{(j)}\Theta\|_p$ instead. This observation can be exploited in two ways. First, we could simply evaluate σ instances of the final MILP resulting for the tightened constraint $\mathbf{x}(0) \in \mathcal{X}_{\text{fun}}$ in order to capture all transformed segments via $\|\Omega^{-1}\mathbf{K}^{(j)}\Theta\|_p$ for every of the σ tuples $(\Theta^{(i)}, \Omega^{(i)})$. Second and more efficiently, we can evaluate the final MILP only once and consider only those transformations resulting in invariant norms, i.e.,

$$\|\Omega^{-1}\mathbf{K}\Theta\|_p = \|\mathbf{K}\|_p \quad \text{for every } \mathbf{K} \in \mathbb{R}^{m \times n}. \quad (18)$$

A sufficient condition for such transformations is as follows.

Lemma 5: Let $\|\Omega\|_p = 1$ and $\|\Theta\|_p = 1$, then (18) holds.

Proof: We first note that invertability of Ω and Θ implies $\|\Omega^{-1}\|_p = \|\Omega\|_p^{-1} = 1$ and $\|\Theta^{-1}\|_p = 1$. Hence,

$$\|\Omega^{-1}\mathbf{K}\Theta\|_p \leq \|\Omega^{-1}\|_p \|\mathbf{K}\|_p \|\Theta\|_p = \|\mathbf{K}\|_p$$

due to sub-multiplicativity. On the other hand,

$$\begin{aligned} \|\mathbf{K}\|_p &= \|\Omega\Omega^{-1}\mathbf{K}\Theta\Theta^{-1}\|_p \\ &\leq \|\Omega\|_p \|\Omega^{-1}\mathbf{K}\Theta\|_p \|\Theta^{-1}\|_p = \|\Omega^{-1}\mathbf{K}\Theta\|_p. \end{aligned}$$

In combination, an inclusion results, which proves (18). ■

While restrictive, common symmetries in MPC often satisfy the conditions in Lemma 5 (see, e.g., the examples in Sect. IV). It remains to comment on the identification of symmetries. In this context, we refer to the methods from [7], which allow to identify tuples (Θ, Ω) that reflect a symmetry purely based on \mathbf{A} , \mathbf{B} , \mathbf{P} , \mathbf{Q} , \mathbf{R} , \mathcal{X} , and \mathcal{U} , i.e., without computing the explicit control law.

D. Combined approaches

We are ready to combine our approaches for a more efficient computation of Lipschitz constants. As already indicated, various combinations of the proposed tools can be considered. We specify two variants that will be used for the numerical benchmark in Section IV. The variants differ in whether symmetries are exploited (according to the previous

section) or not. Hence, slightly neglecting the additional effort for the identification of symmetries and a fundamental domain \mathcal{X}_{fun} , the crucial difference is that either $\mathbf{x}(0) \in \mathcal{X}$ or $\mathbf{x}(0) \in \mathcal{X}_{\text{fun}}$ is considered as a constraint for the initial state in (4). Apart from this difference, all following steps are identical. In fact, we first condense the OCP to a QP of the form (5). We then use the MILP in Theorem 3 to reduce the number of constraints. More precisely, we investigate for each constraint i whether the MILP is infeasible or offers the optimal objective value 0. In any of these cases, the i -th constraint is deleted. For simplicity of notation, we do not introduce different instances of the QP parameters for the two variants or during the constraint reduction. In fact, we simply assume that the previous instances are overwritten. Once the reduced QP is obtained, we solve the following MILP in order to compute the Lipschitz constant $L_1^* = l^*$:

$$\max_{\substack{\mathbf{x}, l, \mathbf{K}^*, \mathbf{K}^+, \mathbf{K}^-, \mathbf{c}, \mathbf{U}^*, \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(n)}, \\ \delta^*, \delta^{(1)}, \dots, \delta^{(m+1)}, \lambda^*, \lambda^{(1)}, \dots, \lambda^{(n)}, \mathbf{r}^*, \mathbf{r}^{(1)}, \dots, \mathbf{r}^{(n)}}} l \quad (19)$$

subject to (7a)–(7b), (8), (10), (11), and (13). Despite the MILP formulation of the reduced QP, a central element is the novel norm computation according to Lemma 1. Again, L_∞^* can be computed analogously.

IV. NUMERICAL BENCHMARK

We demonstrate the effectiveness of the proposed procedures by applying them to examples from the literature summarized in Table I. All MILP in this section are solved using the mixed-integer solver from [16] with constants set to $\bar{\lambda} = \bar{r} = M = 10^4$. For every example, we first compute the minimum Lipschitz constants L_1^* and L_∞^* according to the method in [4, Thm. 5.3], which also served as the starting point for our investigations. The required computation time is listed in the sixth and eighth column of Table II, respectively.

Next, we follow the two variants in Section III-D in order to apply our novel procedures. Regarding the variant exploiting symmetries, we note that all examples in Table I offer rotational symmetries. More precisely, the matrices Θ and Ω are of the form

$$\Theta = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{pmatrix},$$

with $\varphi \in \{\pi, 2\pi\}$ and $\Omega \in \{-1, 1\}$ for Systems 1 and 2. For the two following systems, we have Θ as before and

$$\Omega = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} = \Theta^\top$$

with $\varphi \in \{\pi, 2\pi\}$ for System 3 and $\varphi \in \{\pi/2, \pi, 3\pi/2, 2\pi\}$ for System 4. For System 5, Θ likewise reflects rotation matrices (which we omit for brevity) and $\Omega \in \{-1, 1\}$. In all cases, we easily verify that the condition in Lemma 5 (i.e., $\|\Omega\|_p = \|\Theta\|_p = 1$) holds. Hence, we can restrict our analysis to the fundamental domains in the ninth column of Table I without the need to consider multiple instances of the MILP (19). Before analyzing the performance of our procedures, we note that we consider two different choices for the terminal set \mathcal{T} for each example. First, we simply choose $\mathcal{T} = \mathcal{X}$ (i.e., no

TABLE I: Example systems from the literature.

No.	A	B	\mathcal{X}	\mathcal{U}	Q	R	N	\mathcal{X}_{fun}	Reference
1.	$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$	$\begin{cases} \mathbf{x}_1 \leq 25 \\ \mathbf{x}_2 \leq 5 \end{cases}$	$ \mathbf{u} \leq 1$	I	0.1	10	$\begin{cases} 0 \leq \mathbf{x}_1 \leq 25 \\ \mathbf{x}_2 \leq 5 \end{cases}$	[13, Eqs. (2.8)–(2.9)]
2.	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 4 \end{pmatrix}$	$\begin{cases} \mathbf{x}_1 \leq 5 \\ \mathbf{x}_2 \leq 5 \end{cases}$	$ \mathbf{u} \leq 1$	I	4.5	8	$\begin{cases} 0 \leq \mathbf{x}_1 \leq 5 \\ \mathbf{x}_2 \leq 5 \end{cases}$	[14, Ex. 3]
3.	$\begin{pmatrix} 1.1 & 0.2 \\ -0.2 & 1.1 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0 \\ 0 & 0.4 \end{pmatrix}$	$\begin{cases} \mathbf{x}_1 \leq 5 \\ \mathbf{x}_2 \leq 5 \end{cases}$	$\begin{cases} \mathbf{u}_1 \leq 1 \\ \mathbf{u}_2 \leq 1 \end{cases}$	I	$0.1I$	3	$\begin{cases} 0 \leq \mathbf{x}_1 \leq 5 \\ \mathbf{x}_2 \leq 5 \end{cases}$	[15, Ex. 2.26]
4.	$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\begin{cases} \mathbf{x}_1 \leq 5 \\ \mathbf{x}_2 \leq 5 \end{cases}$	$\begin{cases} \mathbf{u}_1 \leq 1 \\ \mathbf{u}_2 \leq 1 \end{cases}$	I	I	10	$\begin{cases} 0 \leq \mathbf{x}_1 \leq 5 \\ 0 \leq \mathbf{x}_2 \leq 5 \end{cases}$	[7, Ex. 1]
5.	$\begin{pmatrix} 1 & 0.5 & 0.125 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.02 \\ 0.125 \\ 0.5 \end{pmatrix}$	$\begin{cases} \mathbf{x}_1 \leq 20 \\ \mathbf{x}_2 \leq 3 \\ \mathbf{x}_3 \leq 1 \end{cases}$	$ \mathbf{u} \leq 0.5$	I	1	3	$\begin{cases} \mathbf{x}_1 \leq 20 \\ \mathbf{x}_2 \leq 3 \\ 0 \leq \mathbf{x}_3 \leq 1 \end{cases}$	[13, Rem. 4.8]

TABLE II: Computation of Lipschitz constants for different systems.

No.	\mathcal{T}	MILP from [4, Thm. 5.3]					MILP (19) using Lemma 1, Theorem 3, and symmetries					
		$\#\mathcal{R}$	$\#\delta$	L_1^*	Time [s]	L_∞^*	Time [s]	$\#\delta^{(1)}$	$\#\delta^{(2)}$	Time for L_1^* [s]	Time for L_∞^* [s]	Preprocessing [s]
1.	\mathcal{X}	211	64	1.89	103.72	1.27	55.51	22	12	0.1735	0.2481	0.0115
1.	\mathcal{S}	195	70	1.89	9.98	1.27	43.90	22	12	0.1453	0.2289	0.0156
2.	\mathcal{X}	35	52	0.50	15.24	0.50	32.44	32	17	0.6331	0.4759	0.0143
2.	\mathcal{S}	35	58	0.50	19.05	0.50	43.24	32	17	0.3275	0.5374	0.0136
3.	\mathcal{X}	79	28	16.10	2.59	11.70	3.37	18	15	0.3079	0.5262	0.0587
3.	\mathcal{S}	93	32	20.13	1.75	14.63	1.40	16	16	1.2126	1.6429	0.0863
4.	\mathcal{X}	491	84	1.69	71.36	1.69	242.74	36	12	0.2590	0.2520	0.0145
4.	\mathcal{S}	441	88	1.69	66.14	1.69	85.64	36	12	1.6266	0.1910	0.0248
5.	\mathcal{X}	117	30	12.00	2.11	8.00	3.49	28	22	0.3966	0.9440	0.0434
5.	\mathcal{S}	107	46	12.00	9.69	8.00	22.87	28	22	1.1988	2.4368	0.0539

terminal constraints). Second, we consider $\mathcal{T} = \mathcal{S}$ with \mathcal{S} denoting the largest positively invariant set, where the linear quadratic regulator (LQR) can be applied without violating constraints. The computation has been carried out according to [17].

Now, in Table II, we list the number of binary variables of the MILP in [4, Thm. 5.3] in the column $\#\delta$. Further, we also list the number of regions $\mathcal{R}^{(i)}$ of the explicit control law, computed using the multi-parametric toolbox (MPT, [18]), under $\#\mathcal{R}$ as an orientation. Key performance indicators of our procedures are listed in columns nine to 13. First, $\#\delta^{(1)}$ is the number of binary variables for the simplified MILP without considering symmetries. Second, $\#\delta^{(2)}$ reflects the same figure with symmetries. Next, the total computation times for evaluating L_1^* and L_∞^* are listed, respectively. These include the times for all preprocessing steps (such as constraint elimination), which are exclusively listed in the last column for completeness.

As apparent from Table II, the time required to compute the Lipschitz constants can be significantly reduced for all examples. Moreover, one can observe that the computation time for the preprocessing is negligible compared to the time required to compute L_p^* . In fact, although MILP are solved during the preprocessing, the overall time can be reduced in all cases. Furthermore, for the proposed method, the variation of the computation times between the examples is lower. This may indicate, that the proposed method scales better with the model complexity.

Finally, Figure 1 highlights in purple the regions that are considered by the MILP (19) for computing the Lipschitz constant after reducing the number of binary variables according to Lemma 2 and Theorem 3 under consideration of symmetries. As apparent from Figure 2, Theorem 3 allows to identify all regions with a local gain of zero. Moreover, all regions that have the same local gain due to symmetries are also excluded from the computation. Thus only regions that are relevant for computing a Lipschitz constant are considered.

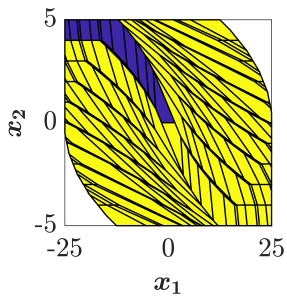


Fig. 1: State space partition of the control law.

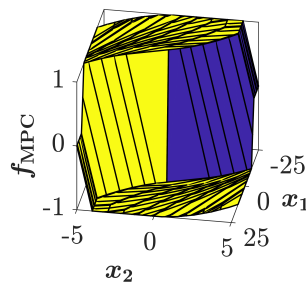


Fig. 2: Optimal control law for System 1 with $\mathcal{T} = \mathcal{X}$.

V. CONCLUSION

We presented an efficient set of methods to compute the minimum Lipschitz constant of an MPC law. The method adopts a known MILP for the computation of Lipschitz constants and uses various procedures (see Lemma 1, Lemma 2 and Theorem 3) to reduce the number of binary variables in the MILP and thus its complexity. The most powerful reduction builds on exploiting saturation and symmetries of the control law. However, the proposed reduction steps can also be applied to systems without symmetries. This allows an efficient computation of the minimum Lipschitz constant even for complex systems of moderate size. For future research it would be interesting to investigate if the MILP (19) can be adopted for the computation of Lipschitz constants of the piecewise quadratic optimal value function $V(\mathbf{x})$ of the OP (4).

REFERENCES

- [1] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition, 2017.
- [2] P.O.M. Scokaert, J.B. Rawlings, and E.S. Meadows. Discrete-time stability with perturbations: application to model predictive control. *Automatica*, 33(3):463–470, 1997.
- [3] D. Limon Marruedo, T. Alamo, and E.F. Camacho. Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of mpc. In *Proc. of the 2002 American Control Conference*, pages 364–369, 2002.
- [4] F. Fabiani and P. J. Goulart. Reliably-stabilizing piecewise-affine neural network controllers. *IEEE Transactions on Automatic Control*, 68(9):5201–5215, 2023.
- [5] D. Teichrib and M. Schulze Darup. Error bounds for maxout neural network approximations of model predictive control. In *Proc. of the 2023 IFAC World Congress*, pages 10853–10859, 2023.
- [6] M. Schulze Darup, M. Jost, G. Pannocchia, and M. Mönnigmann. On the maximal controller gain in linear MPC. *IFAC-PapersOnLine*, 50(1):9218–9223, 2017.
- [7] C. Danielson and F. Borrelli. Symmetric linear model predictive control. *IEEE Transactions on Automatic Control*, 60(5):1244–1259, 2015.
- [8] V. V. Gorokhovik, O. I. Zorko, and G. Birkhoff. Piecewise affine functions and polyhedral sets. *Optimization*, 31(3):209–221, 1994.
- [9] D. Q. Mayne, J. B. Rawlings, C.V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.
- [10] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.
- [11] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
- [12] M. Fischetti and J. Jo. Deep neural networks and mixed-integer linear optimization. *Constraints*, 23(3):296–309, 2018.
- [13] P.-O. Gutman and M. Cwikel. An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. *IEEE Transactions on Automatic Control*, 32(3):251–254, 1987.
- [14] M. Schulze Darup and M. Cannon. Some observations on the activity of terminal constraints in linear MPC. In *Proc. of the 2016 European Control Conference*, pages 4977–4983, 2016.
- [15] M. Schulze Darup. *Numerical methods for the investigation of stabilizability of constrained systems*. PhD thesis, Ruhr-Universität Bochum, Universitätsbibliothek, 2014.
- [16] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.*, 2022.
- [17] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Trans. Autom. Control*, 36(9):1008–1020, 1991.
- [18] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the 2013 European Control Conference*, pages 502–510, 2013.