

Nearly Optimal Simple Explicit MPC Controllers with Stability and Feasibility Guarantees

Holaza, J.; Takacs, B.; Kvasnica, M.; Di Cairano, S.

TR2014-087 July 2014

Abstract

We consider the problem of synthesizing simple explicit model predictive control feedback laws that provide closed-loop stability and recursive satisfaction of state and input constraints. The approach is based on replacing a complex optimal feedback law by a simpler controller whose parameters are tuned, off-line, to minimize the reduction of the performance. The tuning consists of two steps. In the first step, we devise a simpler polyhedral partition by solving a parametric optimization problem. In the second step, we then optimize parameters of local affine feedbacks by minimizing the integrated squared error between the original controller and its simpler counterpart. We show that such a problem can be formulated as a convex optimization problem. Moreover, we illustrate that conditions of closed-loop stability and recursive satisfaction of constraints can be included as a set of linear constraints. Efficiency of the method is demonstrated on two examples.

Optimal Control Applications and Methods

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Nearly optimal simple explicit MPC controllers with stability and feasibility guarantees

J. Holaza^{1,2}, B. Takács^{1,2}, M. Kvasnica^{1,2,*},[†] and S. Di Cairano^{1,2}

¹*Slovak University of Technology in Bratislava, Slovakia*

²*Mitsubishi Electric Research Laboratories, Boston, USA*

SUMMARY

We consider the problem of synthesizing simple explicit model predictive control feedback laws that provide closed-loop stability and recursive satisfaction of state and input constraints. The approach is based on replacing a complex optimal feedback law by a simpler controller whose parameters are tuned, off-line, to minimize the reduction of the performance. The tuning consists of two steps. In the first step, we devise a simpler polyhedral partition by solving a parametric optimization problem. In the second step, we then optimize parameters of local affine feedbacks by minimizing the integrated squared error between the original controller and its simpler counterpart. We show that such a problem can be formulated as a convex optimization problem. Moreover, we illustrate that conditions of closed-loop stability and recursive satisfaction of constraints can be included as a set of linear constraints. Efficiency of the method is demonstrated on two examples. Copyright © 2014 John Wiley & Sons, Ltd.

Received 21 October 2013; Revised 6 May 2014; Accepted 29 May 2014

KEY WORDS: model predictive control; parametric optimization; embedded control

1. INTRODUCTION

Model predictive control (MPC) has become a very popular control strategy especially in process control [1, 2]. MPC is endorsed mainly because of its natural capability of designing feedback controllers for large MIMO systems while considering all of the system's physical constraints and performance specifications, which are implicitly embedded in the optimization problem. Solution of such an optimization problem yields a sequence of predicted optimal control inputs, from which only the first one is applied to the system. Hence, to achieve feedback, the optimization is repeated at each sampling instant, which in turn requires adequate hardware resources. To mitigate the required computational effort, explicit MPC [3] was introduced. In this approach, the repetitive optimization is abolished and replaced by a mere function evaluation, which makes MPC feasible for applications with limited computational resources such as in automotive [4, 5] and aerospace [6] industries. The feedback function is constructed off-line for all admissible initial conditions by parametric programming [7–9]. As shown by numerous authors (see, e.g., [10–14]), for a rich class of MPC problems, the pre-computed solution takes a form of a piecewise affine (PWA) function that maps state measurements onto optimal control inputs. Such a function, however, is often very complex and its complexity can easily exceed limits of the selected implementation hardware.

Therefore, it is important to keep complexity of explicit MPC solutions under control and to reduce it to meet required limits. This task is commonly referred to as *complexity reduction*. Numerous procedures have been proposed to achieve such a goal. Two principal directions are followed in the literature. One option is to replace the complex optimal explicit MPC feedback law by another

*Correspondence to: M. Kvasnica, Slovak University of Technology in Bratislava, Slovakia.

[†]E-mail: michal.kvasnica@stuba.sk

controller, which retains optimality. This can be achieved, for example, by merging together the regions in which local affine expressions are identical [15], by devising a lattice representation of the PWA function [16], or by employing clipping filters [17]. Another possibility is to find a simpler explicit MPC feedback while allowing for a certain reduction of performance with respect to the complex optimal solution. Examples of these methods include, but are not limited to, relaxation of conditions of optimality [18], use of move-blocking [19], formulation of minimum-time setups [20], using multi-resolution techniques [21], approximation of the PWA feedback by a polynomial [22, 23] or by another PWA function defined over orthogonal [24] or simplicial [25] domains, to name just a few. Compared with the performance lossless approaches, the methods that sacrifice some amount of performance typically achieve higher reduction of complexity.

In this paper, we propose a novel method of reducing complexity of explicit MPC solutions, which belong to the class of methods which trade lower complexity for certain reduction of performance. In the presented method, however, the reduction of performance is mitigated as much as possible, hence achieving nearly-optimal performance with low complexity. The presented paper extends our previous results in [26] and [27] by providing detailed technical analysis of the presented results and, more importantly, by introducing synthesis of nearly-optimal explicit MPC controllers that achieve closed-loop stability. The method is based on the assumption that a complex explicit MPC feedback law $\mu(x)$ is given, encoded as a PWA function of the state measurements x . Our objective is to replace $\mu(\cdot)$ by a simpler PWA function $\tilde{\mu}(\cdot)$ such that (i) $\tilde{\mu}(x)$ generates a feasible sequence of control inputs for all admissible values of x ; (ii) $\tilde{\mu}(x)$ renders the closed-loop system asymptotically stable; and (iii) the integrated square error between $\mu(\cdot)$ and $\tilde{\mu}(\cdot)$ (i.e., the suboptimality of $\tilde{\mu}(\cdot)$ with respect to $\mu(\cdot)$) is minimized. By doing so, we obtain a simpler explicit feedback law $\tilde{\mu}(\cdot)$, which is safe (i.e., it provides constraint satisfaction and closed-loop stability) and is nearly optimal.

Designing an appropriate approximate controller, $\tilde{\mu}(\cdot)$ requires first the construction of the polytopic regions over which $\tilde{\mu}(\cdot)$ is defined and then the synthesis of local affine expressions in each of the regions. We propose to approach the first task by solving a simpler MPC optimization problem with a shorter prediction horizon. In this way, we obtain a simple feedback $\hat{\mu}(\cdot)$ as a PWA function. However, such a simpler feedback typically exhibits large deterioration of performance compared with $\mu(\cdot)$. To mitigate such a performance loss, we retain the regions of $\hat{\mu}(\cdot)$ but refine the associated local affine feedback laws to obtain the function $\tilde{\mu}(\cdot)$ such that the error between $\mu(\cdot)$ and $\tilde{\mu}(\cdot)$ is minimized. Here, instead of minimizing the point-wise error as in [26], we illustrate how to minimize the integral of the squared error directly, which is a better indicator of suboptimality. In Section 3, we show that if $\tilde{\mu}(\cdot)$ is required to possess the recursive feasibility property, then the problem of finding the appropriate local feedback laws is always feasible. In other words, we can always refine $\hat{\mu}(\cdot)$ as to obtain a better-performing explicit controller $\tilde{\mu}(\cdot)$. We subsequently extend the procedure and show how to formulate the search for the parameters of $\tilde{\mu}(\cdot)$ by solving a convex quadratic program such that asymptotic closed-loop stability is attained in Section 4. The procedure is summarized in Section 5 and two examples are presented in Section 6. Conclusions are drawn in Section 7.

2. PRELIMINARIES AND PROBLEM DEFINITION

2.1. Notation and definitions

We denote by \mathbb{R} , \mathbb{R}^n and $\mathbb{R}^{n \times m}$ the real numbers, n -dimensional real vectors and $n \times m$ dimensional real matrices, respectively. \mathbb{N} denotes the set of non-negative integers, and \mathbb{N}_i^j , $i \leq j$, the set of consecutive integers, that is, $\mathbb{N}_i^j = \{i, \dots, j\}$. For a vector-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\text{dom}(f)$ denotes its domain. For an arbitrary set S , $\text{int}(S)$ denotes its interior.

Definition 2.1 (Polytope)

A polytope $\mathcal{P} \subset \mathbb{R}^n$ is a convex, closed, and bounded set defined as the intersection of a finite number c of closed affine half-spaces $a_i^T x \leq b_i$, $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $\forall i \in \mathbb{N}_1^c$. Each polytope can be compactly represented as

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}, \quad (1)$$

with $A \in \mathbb{R}^{c \times n}$, $b \in \mathbb{R}^c$.

Definition 2.2 (Vertex representation of a polytope)

Every polytope $\mathcal{P} \subset \mathbb{R}^n$ in (1) can be equivalently written as

$$\mathcal{P} = \left\{ x \mid x = \sum_i \lambda_i v_i, 0 \leq \lambda_i \leq 1, \sum_i \lambda_i = 1 \right\}, \quad (2)$$

where $v_i \in \mathbb{R}^n$, $\forall i \in \mathbb{N}_1^M$ are the vertices of the polytope.

Definition 2.3 (Polytopic partition)

The collection of polytopes $\{\mathcal{R}_i\}_{i=1}^M$ is called a partition of polytope \mathcal{Q} if

1. $\mathcal{Q} = \bigcup_i \mathcal{R}_i$.
2. $\text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset, \forall i \neq j$.

We call each polytope of the collection a *region* of the partition.

Definition 2.4 (Polytopic PWA function)

A vector-valued function $f : \Omega \rightarrow \mathbb{R}^m$ is called PWA over polytopes if

1. $\Omega \subset \mathbb{R}^n$ is a polytope.
2. There exist polytopes $\mathcal{R}_i, i \in \mathbb{N}_1^M$ such that $\{\mathcal{R}_i\}_{i=1}^M$ is a partition of Ω .
3. For each $i \in \mathbb{N}_1^M$, we have $f(x) = F_i x + g_i$, with $F_i \in \mathbb{R}^{m \times n}$, $g_i \in \mathbb{R}^m$.

Definition 2.5 (Maximum control invariant set)

Let $x_{k+1} = Ax_k + Bu_k$ be a linear system that is subject to constraints $x \in \mathcal{X}, u \in \mathcal{U}, \mathcal{X} \subseteq \mathbb{R}^n, \mathcal{U} \subseteq \mathbb{R}^m$. Then the set

$$\mathcal{C}_\infty = \{x_0 \in \mathcal{X} \mid \forall k \in \mathbb{N} : \exists u_k \in \mathcal{U} \text{ s.t. } Ax_k + Bu_k \in \mathcal{X}\} \quad (3)$$

is called the maximum control invariant set.

Remark 2.6

Under mild assumptions, the set \mathcal{C}_∞ in (3) is a polytope, which can be computed, for instance by the MPT Toolbox [28]. The interested reader is referred to [29] and [30] for literature on computing (maximum) control invariant sets.

2.2. Explicit model predictive control

We consider the control of linear discrete-time systems in the state-space form

$$x(t+1) = Ax(t) + Bu(t), \quad (4)$$

with t denoting multiples of the sampling period, $x \in \mathbb{R}^n, u \in \mathbb{R}^m$, (A, B) controllable, and the origin being the equilibrium of (4). The system in (4) is subject to state and input constraints

$$x(t) \in \mathcal{X}, u(t) \in \mathcal{U}, \forall t \in \mathbb{N}, \quad (5)$$

where $\mathcal{X} \subset \mathbb{R}^n, \mathcal{U} \subset \mathbb{R}^m$ are polytopes that contain the origin in their respective interiors. We are interested in obtaining a feedback law $\mu : \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that $u(t) = \mu(x(t))$ drives all states of (4) to the origin while providing recursive satisfaction of state and input constraints, that is, $\forall t \in \mathbb{N} x(t) \in \mathcal{X}, u(t) \in \mathcal{U}$.

As shown for instance in [3], the feedback law $\mu(x)$ can be obtained by computing the explicit representation of the optimizer to the following optimization problem:

$$\mu = \arg \min \sum_{k=0}^{N-1} (x_{k+1}^T Q_x x_{k+1} + u_k^T Q_u u_k) \quad (6a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, \forall k \in \mathbb{N}_0^{N-1}, \quad (6b)$$

$$u_k \in \mathcal{U}, \forall k \in \mathbb{N}_0^{N-1}, \quad (6c)$$

$$x_1 \in \mathcal{C}_\infty, \quad (6d)$$

where x_k, u_k denote, respectively, predictions of the states and inputs at the time step $t + k$, initialized from $x_0 = x(t)$. Moreover, $N \in \mathbb{N}$ is the prediction horizon and $Q_x \succeq 0, Q_u \succ 0$ are the weighting matrices of appropriate dimensions. In the receding horizon implementation of MPC, we are only interested in the first element of the optimal sequence of inputs $U_N^* = [u_0^{*T}, \dots, u_{N-1}^{*T}]^T$. Hence, the receding horizon feedback law is given by

$$\mu(x) := [I_{m \times m} \ 0_{m \times m} \ \cdots \ 0_{m \times m}] U_N^*. \quad (7)$$

Remark 2.7

Note that constraint (6d) implies that if \mathcal{C}_∞ is a control invariant set satisfying (3), $x_k \in \mathcal{X}$ can be satisfied $\forall k \in \mathbb{N}_0^N$.

By solving (6) using *parametric programming* (see [9, 31]), one obtains the explicit representation of the so-called *explicit MPC* feedback law $\mu(\cdot)$ in (7) as a function of the initial condition $x_0 = x(t)$,

$$\mu(x_0) := \begin{cases} F_1 x_0 + g_1 & \text{if } x_0 \in \mathcal{R}_1, \\ \vdots & \\ F_M x_0 + g_M & \text{if } x_0 \in \mathcal{R}_M, \end{cases} \quad (8)$$

with $F_i \in \mathbb{R}^{m \times n}$ and $g_i \in \mathbb{R}^m$.

Theorem 2.8 ([32])

The function $\mu : \mathbb{R}^n \rightarrow \mathbb{R}^m$ in (8) is a polytopic PWA function (cf. Definition 2.4) where $\mathcal{R}_i \subset \mathbb{R}^n$ are the polytopes $\forall i \in \mathbb{N}_1^M$ and M denotes the total number of polytopes. Moreover, the domain of $\mu(\cdot)$ is $\Omega = \bigcup_i \mathcal{R}_i$ where Ω is a polytope such that

$$\Omega = \{x_0 \mid \exists u_0, \dots, u_{N-1} \text{ s.t. (6c) - (6d) holds}\} \quad (9)$$

is the set of all initial conditions for which problem (6) is feasible. Furthermore, $\{\mathcal{R}_i\}$ is the partition of Ω , compare with Definition 2.3.

2.3. Problem statement

The main issue of explicit MPC is that the complexity of the feedback law $\mu(\cdot)$ in (8), expressed by the number of polytopes M , grows exponentially with the prediction horizon N . The more polytopes constitute $\mu(\cdot)$, the more memory is required to store the function in the control hardware and the longer it takes to obtain the value of the optimizer for a particular value of the state measurements. Therefore, we want to replace $\mu(\cdot)$ by a similar, yet less complex, PWA feedback law $\tilde{\mu}(\cdot)$ while preserving recursive satisfaction of constraints in (5). The price we are willing to pay for obtaining a simpler representation is suboptimality of $\tilde{\mu}(\cdot)$ with respect to the optimal representation $\mu(\cdot)$.

Problem 2.9

Given an explicit representation of the MPC feedback function $\mu : \mathbb{R}^n \rightarrow \mathbb{R}^m$ as in (8), we want to synthesize a PWA function $\tilde{\mu} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with

$$\tilde{\mu}(x) = \tilde{F}_i x + \tilde{g}_i \text{ if } x \in \tilde{\mathcal{R}}_i, \forall i \in \mathbb{N}_1^{\tilde{M}}, \quad (10)$$

that is, to find the integer $\tilde{M} < M$, polytopes $\tilde{\mathcal{R}}_i \subset \mathbb{R}^n$, $i \in \mathbb{N}_1^{\tilde{M}}$, and gains $\tilde{F}_i \in \mathbb{R}^{m \times n}$, $\tilde{g}_i \in \mathbb{R}^m$ such that

- R1: For each $x \in \text{dom}(\mu)$, the simpler feedback $\tilde{\mu}(\cdot)$ provides recursive satisfaction of state and input constraints in (5), that is, $\forall t \in \mathbb{N}$, we have that $\tilde{\mu}(x(t)) \in \mathcal{U}$ and $Ax(t) + B\tilde{\mu}(x(t)) \in \mathcal{X}$.
- R2: Feedback $\tilde{\mu}(\cdot)$ renders the origin an asymptotically stable equilibrium of the closed-loop system $x(t+1) = Ax(t) + B\tilde{\mu}(x(t))$.
- R3: $\tilde{\mu}(\cdot)$ is chosen such that the squared error between the PWA functions $\mu(\cdot)$ and $\tilde{\mu}(\cdot)$, when integrated over the domain of $\mu(\cdot)$, Ω , is minimized.

$$\min \int_{\Omega} \|\mu(x) - \tilde{\mu}(x)\|_2^2 dx. \quad (11)$$

In (11), dx is the Lebesgue measure of Ω , see [33]. The task of Problem 2.9 is illustrated graphically in Figure 1.

Remark 2.10

Replacing the integrated squared error criterion in (11) by point-wise squared errors of the form

$$\min \sum_i \|\mu(w_i) - \tilde{\mu}(w_i)\|_2^2 \quad (12)$$

for a finite set of points w_i that can be counterproductive. Take the case of Figure 1, consider only region $\tilde{\mathcal{R}}_1$, and let w_1, w_2 be its vertices. Then the point-wise error is small (because it is evaluated only in the vertices), whereas the integrated error between $\mu(\cdot)$ and $\tilde{\mu}(\cdot)$ is significantly larger. This issue can be mitigated, to some extent, by devising many evaluation points w_i . But one still only obtains an approximation of the integrated error criterion. Therefore, in this paper, we show how to minimize (11) directly, without resorting to point-wise approximations of the error objective.

3. SIMPLE CONTROLLERS WITH GUARANTEES OF RECURSIVE FEASIBILITY

In this section, we propose a two-step procedure for synthesis of a simple feedback $\tilde{\mu}(\cdot)$ that fulfills requirements R1 and R3 of Problem 2.9. The closed-loop stability criterion R2 will be addressed in

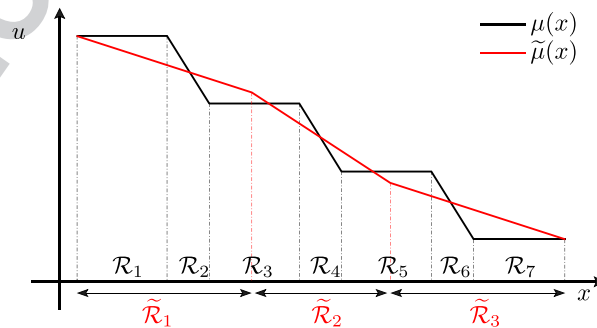


Figure 1. The function $\mu(\cdot)$, shown in black, is given. The task in Problem 2.9 is to synthesize the function $\tilde{\mu}(\cdot)$, shown in red, which is less complex (here it is defined just over three regions instead of seven for $\mu(\cdot)$) and minimizes the integrated square error (11).

Section 4. In the first step, we construct polytopes $\tilde{\mathcal{R}}_i, i \in \mathbb{N}_1^{\tilde{M}}$ with $\tilde{M} \ll M$ (recall that M is the number of polytopes that define the optimal feedback $\mu(\cdot)$) such that

$$\bigcup_i \tilde{\mathcal{R}}_i = \bigcup_j \mathcal{R}_j, \quad (13)$$

that is, that the domain of $\tilde{\mu}(\cdot)$ is identical to the domain of $\mu(\cdot)$. In the second step for each $i \in \mathbb{N}_1^{\tilde{M}}$, we choose the gains \tilde{F}_i and offsets \tilde{g}_i of $\tilde{\mu}(\cdot)$ in (10) such that the simpler feedback $\tilde{\mu}(\cdot)$ provides recursive satisfaction of constraints in (5) and the approximation error in (11) is minimized.

3.1. Selection of the polytopic partition

The objective here is to find polytopic regions $\tilde{\mathcal{R}}_i, i \in \mathbb{N}_1^{\tilde{M}}$ such that (13) holds with $\tilde{M} < M$. First, recall that from Theorem 2.8, $\bigcup_j \mathcal{R}_j = \Omega$ by (9). Hence, we require $\bigcup_i \tilde{\mathcal{R}}_i = \Omega$. We propose to obtain polytopes $\tilde{\mathcal{R}}_i$ by solving (6) again but with a lower value of the prediction horizon, say with $\hat{N} < N$, where N is the prediction horizon for which the original (complex) controller μ was obtained. Then, by Theorem 2.8, we obtain the feedback law $\hat{\mu}(\cdot)$ as a PWA function of x

$$\hat{\mu}(x) = \hat{F}_i x + \hat{g}_i \text{ if } x \in \tilde{\mathcal{R}}_i, \forall i \in \mathbb{N}_1^{\tilde{M}}, \quad (14)$$

which is defined over \tilde{M} polytopes $\tilde{\mathcal{R}}_i$.

Lemma 3.1

Let $\mu(\cdot)$ as in (8) be obtained by solving (6) according to Theorem 2.8 for some prediction horizon N . Let $\hat{\mu}(\cdot)$ be the explicit MPC feedback function in (14), obtained by solving (6) for some $\hat{N} < N$. Then (13) holds.

Proof

The feasible set Ω in (9) is the projection of constraints in (6) onto the x -space, see, for example, [34, 35]. Because (6d) are the only state constraints of the problem, Ω is independent of the choice of the prediction horizon. Therefore, $\Omega_N = \Omega_{\hat{N}}$. Finally, because $\bigcup_j \mathcal{R}_j = \Omega_N = \Omega_{\hat{N}} = \bigcup_i \tilde{\mathcal{R}}_i$ by Theorem 2.8, the result follows. \square

Thus, we can obtain polytopic regions $\tilde{\mathcal{R}}_i$ of the simpler function (10) by solving (6) explicitly for a shorter value of the prediction horizon. To achieve the least complex representation of $\tilde{\mu}(\cdot)$, it is recommended to choose low values of \hat{N} . The smallest number of polytopes, that is, \tilde{M} , will be achieved for $\hat{N} = 1$.

Remark 3.2

The advantage of the procedure presented here is that the domain of $\mu(\cdot)$ is partitioned into $\{\tilde{\mathcal{R}}_i\}$ in such a way that the approximation problem is always feasible, that is, there always exists parameters \tilde{F}_i, \tilde{g}_i in (10) such that $\tilde{\mu}$ guarantees recursive satisfaction of input and state constraints. This is not always the case if an arbitrary partition is selected.

Remark 3.3

By solving (6) for $\hat{N} < N$, we obtain the explicit representation of a simple controller $\hat{\mu}(\cdot)$ as a PWA function in (14). Such a function already provides recursive satisfaction of constraints in (5) due to (6d) and therefore solves R1 in Problem 2.9. However, there is no guarantee that $\hat{\mu}(\cdot)$ minimizes the approximation error (11). Hence, (14) is expected to exhibit significant suboptimality when compared with the (complex) optimal feedback $\mu(\cdot)$. In the following section, we aim at refining local affine feedback laws of $\hat{\mu}(\cdot)$ such that the amount of suboptimality is significantly reduced.

3.2. Function fitting

In the previous section, we have shown how to compute the polytopic partition $\tilde{\mathcal{R}}_i$ by solving (6) using parametric programming for $\hat{N} < N$. Next, we aim at finding parameters \tilde{F}_i, \tilde{g}_i

First, recall that, from Theorem 2.8, the polytopes $\tilde{\mathcal{R}}_i$ form the partition of the domain of $\hat{\mu}(\cdot)$, that is, their respective interiors do not overlap. Therefore, we can split the search for \tilde{F}_i, \tilde{g}_i for $i \in \mathbb{N}_1^{\tilde{M}}$ in Problem 2.9 into a series of \tilde{M} problems of the following form:

$$\min_{\tilde{F}_i, \tilde{g}_i} \int_{\tilde{\mathcal{R}}_i} \|\mu(x) - \tilde{\mu}(x)\|_2^2 dx, \quad (15a)$$

$$\text{s.t. } \tilde{F}_i x + \tilde{g}_i \in \mathcal{U}, \quad \forall x \in \tilde{\mathcal{R}}_i, \quad (15b)$$

$$Ax + B(\tilde{F}_i x + \tilde{g}_i) \in \mathcal{C}_\infty, \quad \forall x \in \tilde{\mathcal{R}}_i. \quad (15c)$$

Here, recall that $\tilde{\mu}(x) = \tilde{F}_i x + \tilde{g}_i$ is the affine representation of the approximate control law valid in a particular region $\tilde{\mathcal{R}}_i$ via (10). The constraint (15b) ensures satisfaction of input constraints, whereas (15c) provides recursive satisfaction of state constraints because \mathcal{C}_∞ is assumed to satisfy Definition 2.5.

However, there are three technical issues, which complicate the search for \tilde{F}_i, \tilde{g}_i from (15).

1. Even when x is restricted to a particular polytope $\tilde{\mathcal{R}}_i$, $\mu(\cdot)$ over $\tilde{\mathcal{R}}_i$ is still a PWA function.
2. The integration in (15a) has to be performed over polytopes in dimension $n \geq 1$.
3. The constraints in (15b) and (15c) have to hold for all points $x \in \tilde{\mathcal{R}}_i$, that is, for an infinite number of points.

The first issue can be tackled as follows. Consider a fixed index i , that is, take $\tilde{\mathcal{R}}_i$ and recall that the (complex) optimal feedback $\mu(\cdot)$ is defined over M polytopes \mathcal{R}_j . For each $j \in \mathbb{N}_1^M$, compute first the intersection between $\tilde{\mathcal{R}}_i$ and \mathcal{R}_j , that is,

$$\mathcal{Q}_{i,j} = \tilde{\mathcal{R}}_i \cap \mathcal{R}_j, \quad \forall j \in \mathbb{N}_1^M. \quad (16)$$

Because $\tilde{\mathcal{R}}_i$ and \mathcal{R}_j are assumed to be polytopes, each $\mathcal{Q}_{i,j}$ is a polytope as well. In each intersection $\mathcal{Q}_{i,j}$, the expressions for both $\mu(\cdot)$ and $\tilde{\mu}(\cdot)$ are affine, which follows from (8) and (10), respectively. Hence, we can equivalently represent the approximation objective (15a) as

$$\min_{\tilde{F}_i, \tilde{g}_i} \sum_{j \in \mathcal{J}_i} \int_{\mathcal{Q}_{i,j}} \|(F_j x - g_j) - (\tilde{F}_i x + \tilde{g}_i)\|_2^2 dx, \quad (17)$$

where F_j and g_j are the gains and offsets of the optimal feedback in region \mathcal{R}_j . The outer summation only needs to consider indices of polytopes of $\mu(\cdot)$ for which the intersection in (16) is non-empty, that is, $\mathcal{J}_i = \{j \in \mathbb{N}_1^M \mid \tilde{\mathcal{R}}_i \cap \mathcal{R}_j \neq \emptyset\}$ for a fixed i .

To evaluate the integral in (17), recall that for each i - j combination, F_j and g_j are known matrices/vectors, but \tilde{F}_i and \tilde{g}_i are optimization variables. Furthermore, $\mathcal{Q}_{i,j}$ are polytopes in \mathbb{R}^n with $n \geq 1$. To obtain an analytic expression for the integral, we use the result of [36], extended by [33]

Lemma 3.4 ([33])

Let f be a homogeneous polynomial of degree d in n variables, and let s_1, \dots, s_{n+1} be the vertices of an n -dimensional simplex Δ . Then

$$\int_{\Delta} f(y) dy = \beta \sum_{1 \leq i_1 \leq \dots \leq i_d \leq n+1} \sum_{\epsilon \in \{\pm 1\}^d} \left(\left(\prod_{j=1}^d \epsilon_j \right) \cdot f \left(\sum_{k=1}^d \epsilon_k s_{i_k} \right) \right), \quad (18)$$

where

$$\beta = \frac{\text{vol}(\Delta)}{2^d d! \binom{d+n}{d}} \quad (19)$$

and $\text{vol}(\Delta)$ is the volume of the simplex.

However, Lemma 3.4 is not directly applicable to evaluate the integral in (17) because the polytopes $\mathcal{Q}_{i,j}$ are not simplices in general. To proceed, we therefore first have to tessellate each polytope $\mathcal{Q}_{i,j}$ into simplices $\Delta_{i,j,1}, \dots, \Delta_{i,j,K}$ with $\text{int}(\Delta_{i,j,k_1}) \cap \text{int}(\Delta_{i,j,k_2}) = \emptyset$ for all $k_1 \neq k_2$ and $\cup_k \Delta_{i,j,k} = \mathcal{Q}_{i,j}$. Then we can rewrite (17) as a sum of the integrals evaluated over each simplex

$$\min_{\tilde{F}_i, \tilde{g}_i} \sum_{j \in \mathcal{J}_i} \sum_{k=1}^{K_{i,j}} \int_{\Delta_{i,j,k}} \|(F_j x - g_j) - (\tilde{F}_i x + \tilde{g}_i)\|_2^2 dx, \quad (20)$$

where $K_{i,j}$ is the number of simplices tessellating $\mathcal{Q}_{i,j}$. Furthermore, note that Lemma 3.4 only applies to homogeneous polynomials. The integral error in (20), however, is not homogeneous. To see this, expand $f(x) := \|(F_j x + g_j) - (\tilde{F}_i x + \tilde{g}_i)\|_2^2$ to $f(x) := x^T Q x + r^T x + q$ with

$$Q = F_j^T F_j - 2F_j \tilde{F}_i + \tilde{F}_i^T \tilde{F}_i, \quad (21a)$$

$$r = 2(F_j^T \tilde{g}_i + \tilde{F}_i^T \tilde{g}_i - \tilde{F}_i^T g_j - F_j^T \tilde{g}_i), \quad (21b)$$

$$q = g_j^T g_j - 2g_j^T \tilde{g}_i + \tilde{g}_i^T \tilde{g}_i. \quad (21c)$$

Then we can see that $f(x)$ is a quadratic function in the optimization variables \tilde{F}_i and \tilde{g}_i , but is not homogeneous, because not all of its monomials have the same degree (in particular, we have monomials of degrees 2, 1, and 0 in f). However, because an integral is closed under linear combinations, we have that

$$\int_{\Delta} f(x) = \int_{\Delta} f_{\text{quad}}(x) + \int_{\Delta} f_{\text{lin}}(x) + \int_{\Delta} f_{\text{const}}, \quad (22)$$

with $f_{\text{quad}}(x) := x^T Q x$, $f_{\text{lin}} := r^T x$ and $f_{\text{const}} := q$ and the integrand dx is omitted for brevity. Because each of these newly defined functions is a homogeneous polynomial of degrees 2, 1, and 0, respectively, the integral $\int_{\Delta} f(x) dx$ can now be evaluated by applying (18) of Lemma 3.4 to each integral in the right-hand side of (22). We hence obtain an analytic expression for the integral error as a quadratic function of the unknowns \tilde{F}_i and \tilde{g}_i .

Remark 3.5

The integral of a constant q over a compact set Δ is equal to a scaled volume of Δ , that is, $\int_{\Delta} q = q \text{vol}(\Delta)$.

Remark 3.6

To see that the integral in (17) is a quadratic function of decision variables \tilde{F}_i and \tilde{g}_i , note that in the integration rule (18), β is a constant, ϵ_i are ± 1 , hence, (18) is a scaled sum of values of $f(\cdot)$, evaluated at vertices of the simplex. Because $f(\cdot)$ is a quadratic function as in (21), the conclusion follows.

Finally, when optimizing for \tilde{F}_i and \tilde{g}_i , we need to ensure that the constraints in (15) hold for all points $x \in \tilde{\mathcal{R}}_i$. By our assumptions, the sets \mathcal{U} and \mathcal{C}_{∞} are polytopes and hence can be represented by $\mathcal{U} = \{u \mid H_u u \leq h_u\}$ and $\mathcal{C}_{\infty} = \{x \mid H_c x \leq h_c\}$. By using $u = \tilde{F}_i x + \tilde{g}_i$, constraints (15b) and (15c) can be compactly written as

$$\forall x \in \tilde{\mathcal{R}}_i : f(x) \leq 0, \quad (23)$$

with

$$f(x) := \begin{bmatrix} H_u \tilde{F}_i \\ H_c (A + B \tilde{F}_i) \end{bmatrix} x + \begin{bmatrix} H_u \tilde{g}_i - h_u \\ H_c \tilde{g}_i - h_c \end{bmatrix}. \quad (24)$$

Then we can state our next result.

Lemma 3.7

Let $\mathcal{V}_i = \{v_{i,1}, \dots, v_{i,n_{v,i}}\}$, and $v_{i,j} \in \mathbb{R}^n$ be the vertices of polytope $\tilde{\mathcal{R}}_i$ (see Definition 2.2). Then (23) is satisfied $\forall x \in \tilde{\mathcal{R}}_i$ if and only if $f(v_{i,j}) \leq 0$ holds for all vertices.

Proof

To simplify the exposition, we replace $\tilde{\mathcal{R}}_i$ by \mathcal{P} to avoid double indexing, and we let the vertices of \mathcal{P} be v_1, \dots, v_{n_v} . As seen from (24), $f(\cdot)$ is a linear function of x . Necessity is obvious because $v_j \in \mathcal{P}$ trivially holds for all vertices, compare with Definition 2.2. To show sufficiency, represent each point of \mathcal{P} as a convex combination of its vertices v_j , that is, $z = \sum_j \lambda_j v_j$. Then $f(z) \leq 0 \forall z \in \mathcal{P}$ is equivalent to $f(\sum_j \lambda_j v_j) \leq 0, \forall \lambda \in \Lambda$, where $\Lambda = \{\lambda \mid \sum_j \lambda_j = 1, \lambda_j \geq 0\}$ is the unit simplex. Because $f(\cdot)$ is assumed linear, we have $f(\sum_j \lambda_j v_j) = \sum_j \lambda_j f(v_j)$. Therefore, $\sum_j \lambda_j f(v_j) \leq 0$ holds for an arbitrary $\lambda \in \Lambda$ because $f(v_j) \leq 0$ is assumed to hold and because each λ_j is non-negative. Therefore, $f(v_j) \leq 0 \Rightarrow f(z) \leq 0 \forall z \in \mathcal{P}$. \square

By combining Lemma 3.7 with the integration result in (22), we can formulate the search for \tilde{F}_i, \tilde{g}_i from (15) as

$$\min_{\tilde{F}_i, \tilde{g}_i} \sum_{j \in \mathcal{J}_i} \sum_{k=1}^{K_{i,j}} \int_{\Delta_{i,j,k}} \|(F_j x - g_j) - (\tilde{F}_i x + \tilde{g}_i)\|_2^2 dx, \quad (25a)$$

$$\text{s.t. } \tilde{F}_i v_{i,\ell} + \tilde{g}_i \in \mathcal{U}, \forall v_{i,\ell} \in \text{vert}(\tilde{\mathcal{R}}_i), \quad (25b)$$

$$A v_{i,\ell} + B(\tilde{F}_i v_{i,\ell} + \tilde{g}_i) \in \mathcal{C}_\infty, \forall v_{i,\ell} \in \text{vert}(\tilde{\mathcal{R}}_i), \quad (25c)$$

where $\text{vert}(\tilde{\mathcal{R}}_i)$ enumerates all vertices of the corresponding polytope. Because each polytope $\tilde{\mathcal{R}}_i$ has only finitely many vertices [37], problem (25) has a finite number of constraints. Moreover, the objective in (25a) is a quadratic function in the unknowns \tilde{F}_i, \tilde{g}_i and its analytic form can be obtained via (18). Finally, because the sets \mathcal{U} and \mathcal{C}_∞ are assumed to be polytopic, all constraints in (25) are linear. Thus, problem (25) is a quadratic optimization problem for each $i \in \mathbb{N}_1^{\tilde{M}}$, where \tilde{M} is the number of polytopes that constitute the domain of $\tilde{\mu}(\cdot)$ in (10).

As our next result, we show that if polytopes $\tilde{\mathcal{R}}_i$ are chosen as suggested by Lemma 3.1, then (25) is always feasible for each $i \in \mathbb{N}_1^{\tilde{M}}$.

Corollary 3.8

Let $\tilde{\mathcal{R}}_i, i \in \mathbb{N}_1^{\tilde{M}}$ be obtained by Lemma 3.1 for $\hat{N} < N$. Then the optimization problem (25) is always feasible, that is, for each $i \in \mathbb{N}_1^{\tilde{M}}$, there exists matrices \tilde{F}_i and vectors \tilde{g}_i such that the simplified feedback $\tilde{\mu}(x)$ from (10) provides recursive satisfaction of constraints in (5) for an arbitrary $x \in \Omega$.

Proof

It follows directly from the fact that the polytopes $\tilde{\mathcal{R}}_i$ in (25) are the same as in (14); therefore, the choice $\tilde{F}_i = \hat{F}_i$ and $\tilde{g}_i = \hat{g}_i$ obviously satisfies all constraints in (25). \square

Remark 3.9

The improved feedback $\tilde{\mu}(\cdot)$ in (10), whose parameters \tilde{F}_i, \tilde{g}_i are obtained from (25), is not necessarily continuous. If desired, continuity can be enforced by adding the constraints $\tilde{F}_i w_k + \tilde{g}_i = \tilde{F}_j w_k + \tilde{g}_j$ to constraints in (25), where w_k are all vertices of the $n - 1$ dimensional intersection $\tilde{\mathcal{R}}_i \cap \tilde{\mathcal{R}}_j, \forall i, j \in \mathbb{N}_1^{\tilde{M}}$. Note that, because the simple feedback $\hat{\mu}$ is continuous, the choice $\tilde{F}_i = \hat{F}_i, \tilde{g}_i = \hat{g}_i$ is a feasible continuous solution in (25). Hence, the conclusions of Lemma 3.7 hold even if continuity of (10) is enforced. Needless to say, sacrificing continuity allows for a greater reduction of the approximation error in (25a).

Remark 3.10

Optimization problem (25) naturally covers the multi-input scenario where $\tilde{F}_i \in \mathbb{R}^{m \times n}$, $\tilde{g}_i \in \mathbb{R}^m$ with $m \geq 1$.

4. CLOSED-LOOP STABILITY

Although the procedure in Section 3 yields a feedback $\tilde{\mu}(\cdot)$ simpler than $\mu(\cdot)$ that guarantees recursive satisfaction of state and input constraints in (5) and minimizes the loss of optimality measured by (11), it does not, however, provide a priori guarantees of closed-loop stability. In this section, we therefore show how to adjust the search for parameters of $\tilde{\mu}(\cdot)$ in (10) such that the closed-loop system

$$x(t+1) = Ax(t) + B\tilde{\mu}(x(t)) \quad (26)$$

is asymptotically stable with respect to the origin as an equilibrium point.

To achieve such a property, we will assume that for system (4), we have knowledge of a convex piecewise linear (PWL) Lyapunov function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ with $\text{dom}(V) \supseteq \mathcal{C}_\infty$. Such a PWL Lyapunov function can be straightforwardly obtained by considering the Minkowski function (also called the Gauge function) of \mathcal{C}_∞ in (3). Let the minimal half-space representation of \mathcal{C}_∞ be normalized to

$$\mathcal{C}_\infty = \{x \in \mathbb{R}^n \mid Wx \leq \mathbf{1}\}, \quad (27)$$

where $\mathbf{1}$ is a column vector of the ones in the appropriate dimension. Then $V(\cdot)$ is given [29] as

$$V(x) := \max_{k \in \mathbb{N}_1^d} w_k^T x, \quad (28)$$

where w_j^T denotes the j -th row of $W \in \mathbb{R}^{d \times n}$ in (27). It follows from [29, 38] that $V(\cdot)$ of (28) is a Lyapunov function for system (4), with domain \mathcal{C}_∞ . Importantly, note that K of affine functions defining (28) is equal to d , the number of facets of \mathcal{C}_∞ .

Then it is well known (see, e.g., [38]) that $\tilde{\mu}(\cdot)$ will render the closed-loop system (26) asymptotically stable if

$$V(Ax + B\tilde{\mu}(x)) \leq \gamma V(x) \quad (29)$$

holds for all $x \in \mathcal{C}_\infty$ and for some $\gamma \in [0, 1)$. By adding (29) to the constraints of (15), we can formulate the search for parameters \tilde{F}_i, \tilde{g}_i of a stabilizing feedback $\tilde{\mu}(\cdot)$ in (10) as

$$\min_{\tilde{F}_i, \tilde{g}_i} \int_{\tilde{\mathcal{R}}_i} \|\mu(x) - \tilde{\mu}(x)\|_2^2 dx \quad (30a)$$

$$\text{s.t. } \tilde{F}_i x + \tilde{g}_i \in \mathcal{U}, \forall x \in \tilde{\mathcal{R}}_i, \quad (30b)$$

$$V(Ax + B(\tilde{F}_i x + \tilde{g}_i)) \leq \gamma V(x), \forall x \in \tilde{\mathcal{R}}_i, \quad (30c)$$

which needs to be solved for all regions $\tilde{\mathcal{R}}_i$ of $\tilde{\mu}(\cdot)$.

Remark 4.1

Because any γ level set with $\gamma \in [0, 1)$ of a Lyapunov function is an invariant set, constraint (30c) entails the invariance constraint Ax

$V(\cdot)$ as in (28) implicitly guarantees that $u = \tilde{F}_i x + \tilde{g}_i = 0$ for $x = 0$.

Lemma 4.2

With $\mathcal{U} = \{u \mid H_u u \leq h_u\}$ and $\mathcal{C}_\infty = \{u \mid Wx \leq \mathbf{1}\}$ as in (27), problem (30) is equivalent to

$$\min_{\tilde{F}_i, \tilde{g}_i} \sum_{j \in \mathcal{J}_i} \sum_{k=1}^{K_{i,j}} \int_{\Delta_{i,j,k}} \|(F_j x - g_j) - (\tilde{F}_i x + \tilde{g}_i)\|_2^2 dx, \quad (31a)$$

$$\text{s.t. } H_u (\tilde{F}_i v_{i,\ell} + \tilde{g}_i) \leq h_u, \quad \forall v_{i,\ell} \in \text{vert}(\tilde{\mathcal{R}}_i), \quad (31b)$$

$$w_k^T (A v_{i,\ell} + B (\tilde{F}_i v_{i,\ell} + \tilde{g}_i)) \leq \gamma \bar{m}_i, \quad \forall k \in \mathbb{N}_1^d, \quad \forall v_{i,\ell} \in \text{vert}(\tilde{\mathcal{R}}_i), \quad (31c)$$

with

$$\bar{m}_i = \max_{\ell \in \mathbb{N}_1^{n_{v,i}}} \max_{k \in \mathbb{N}_1^d} w_k^T v_{i,\ell} \quad (32)$$

and $v_{i,1}, \dots, v_{i,n_{v,i}}$ being the vertices of polytope $\tilde{\mathcal{R}}_i$.

Proof

First, note that (31b) is identical to (25b) and that (25c) is entailed in (31c), compare with Remark 4.1. Therefore, it suffices to show that (31c) is equivalent to (30c). With $V(\cdot)$ as in (28), the constraint (30c) yields

$$\max_{k \in \mathbb{N}_1^d} w_k^T (Ax + B (\tilde{F}_i x + \tilde{g}_i)) \leq \gamma \max_{k \in \mathbb{N}_1^d} w_k^T x, \quad \forall x \in \tilde{\mathcal{R}}_i. \quad (33)$$

Because $\tilde{\mathcal{R}}_i$ is assumed to be a polytope and because the arguments of the maximum in the right-hand side of (33) are linear functions of x , the maximum is attained at one of the vertices $\{v_{i,1}, \dots, v_{i,n_{v,i}}\}$ of $\tilde{\mathcal{R}}_i$ and is denoted by \bar{m}_i as in (32). Then, we can equivalently write (33) as

$$\max_{k \in \mathbb{N}_1^d} w_k^T (Ax + B (\tilde{F}_i x + \tilde{g}_i)) \leq \gamma \bar{m}_i, \quad \forall x \in \tilde{\mathcal{R}}_i. \quad (34)$$

Next, denote $f(x) := \max_{k \in \mathbb{N}_1^d} w_k^T (Ax + B (\tilde{F}_i x + \tilde{g}_i))$ and recall that the maximum of affine functions is a convex function [39, Section 3.2.3]. With $f(\cdot)$ convex, it is trivial to show that $f(x) \leq \bar{m}_i$ for all $x \in \tilde{\mathcal{R}}_i$ if and only if $f(v_{i,\ell}) \leq \bar{m}_i$ for all vertices $v_{i,1}, \dots, v_{i,n_{v,i}}$ of polytope $\tilde{\mathcal{R}}_i$. Hence, (34) is equivalent to

$$\max_{k \in \mathbb{N}_1^d} w_k^T (A v_{i,\ell} + B (\tilde{F}_i v_{i,\ell} + \tilde{g}_i)) \leq \gamma \bar{m}_i, \quad \forall v_{i,\ell} \in \text{vert}(\tilde{\mathcal{R}}_i). \quad (35)$$

Finally, because $\max_k w_k^T z \leq \bar{m}_i$ holds if and only if $w_k^T z \leq \bar{m}_i$ is satisfied for all k , we obtain

$$w_k^T (A v_{i,\ell} + B (\tilde{F}_i v_{i,\ell} + \tilde{g}_i)) \leq \gamma \bar{m}_i, \quad \forall k \in \mathbb{N}_1^d, \quad \forall v_{i,\ell} \in \text{vert}(\tilde{\mathcal{R}}_i), \quad (36)$$

which is precisely the same as in (31c). \square

Remark 4.3

Note that \bar{m}_i in (32) can be computed analytically once the vertices of $\tilde{\mathcal{R}}_i$ are known.

For each region $\tilde{\mathcal{R}}_i$, (31) is a quadratic program for the objective function (31a), (cf. Remark 3.6) and all constraints in (31) are linear functions of \tilde{F}_i and \tilde{g}_i . Hence, the search for parameters \tilde{F}_i, \tilde{g}_i of a stabilizing simpler feedback $\tilde{\mu}(\cdot)$ of (10) can be formulated as a series of \tilde{M} quadratic programs, as captured by the following theorem.

Theorem 4.4

Suppose that the quadratic programs in (31) are feasible for all regions $\tilde{\mathcal{R}}_i$, $i = 1, \dots, \tilde{M}$ and for a selected $\gamma \in [0, 1)$. Then the refined simpler feedback $\tilde{\mu}(\cdot)$ of (10) provides recursive satisfaction of state and input constraints in (5), attains asymptotic stability of the closed-loop system in (26), and minimizes the integrated squared error in (11).

Proof

The first two constraints in (31) are the same as in (25) and enforce recursive feasibility according to Corollary 3.8. Similarly, minimization of the integrated squared error is the same as in (31a). Finally, feasibility of (31) implies that there exist parameters \tilde{F}_i , \tilde{g}_i of $\tilde{\mu}(\cdot)$, which enforces a given decay of the Lyapunov function by (29) and by Lemma 4.2. \square

Remark 4.5

Unlike Corollary 3.8, which provides necessary and sufficient conditions, feasibility of QPs (31) is merely sufficient for the existence of $\tilde{\mu}(\cdot)$ that renders the closed-loop system asymptotically stable. If the QPs are infeasible, one can enlarge the value of γ , provided that it fulfills $\gamma \in [0, 1)$ or alternatively employs a new partition $\{\tilde{\mathcal{R}}_i\}_i^{\tilde{M}}$ obtained for a different value of \hat{N} in Lemma 3.1.

5. COMPLETE PROCEDURE

Here, we summarize the procedure developed in Sections 3 and 4. In order to devise a simpler explicit feedback law $\tilde{\mu}(\cdot)$ in (10) that solves Problem 2.9, that is, that approximates a given complex solution $\mu(\cdot)$, provides recursive satisfaction of input and state constraints, and maintains closed-loop stability, we propose to proceed as follows:

1. Select $\hat{N} < N$ and obtain $\tilde{\mathcal{R}}_i$ by solving (6). Denote by \tilde{M} the number of regions $\tilde{\mathcal{R}}_i$.
2. If closed-loop stability is to be enforced, select $\gamma \in [0, 1)$.
3. For each $i \in \mathbb{N}_1^{\tilde{M}}$ do.
4. Compute $\mathcal{Q}_{i,j}$ from (16) for each $j \in \mathbb{N}_1^{\tilde{M}}$.
5. Triangulate each intersection $\mathcal{Q}_{i,j}$ into simplices $\Delta_{i,j,1}, \dots, \Delta_{i,j,K}$ and enumerate their respective vertices.
6. Obtain the analytic expression of the integrals in (25a), (if only recursive feasibility is desired) or in (31a), (for synthesis of a closed-loop stabilizing feedback) by (18).
7. Enumerate vertices of $\tilde{\mathcal{R}}_i$ and obtain \tilde{F}_i , \tilde{g}_i by solving (25) or (31) as a quadratic optimization problem.

We remark that Steps 5–7 need to be performed for each combination of indices i and j for which $\mathcal{Q}_{i,j}$ in (16) is a non-empty set. Obtaining the polytopes $\tilde{\mathcal{R}}_i$ in Step 1 by solving (6) explicitly can be performed, for example, by the MPT Toolbox [28] or by the Hybrid Toolbox [40]. Computation of intersections, tessellation (via Delaunay triangulation), and enumeration of vertices in Steps 4 and 5 can also be done by MPT. Finally, the optimization problem (25) can be formulated by YALMIP [41] and solved using off-the-shelf software, for example, by GUROBI [42] or quadprog of MATLAB.

6. EXAMPLES

In this section, we demonstrate the effectiveness of the presented explicit MPC complexity reduction method on two examples with different number of states.

6.1. Two-dimensional example

Consider the second-order, discrete-time, linear time-invariant system

$$x(t+1) = \begin{bmatrix} 0.9539 & -0.3440 \\ -0.4833 & -0.5325 \end{bmatrix} x(t) + \begin{bmatrix} -0.4817 \\ -0.5918 \end{bmatrix} u(t), \quad (37)$$

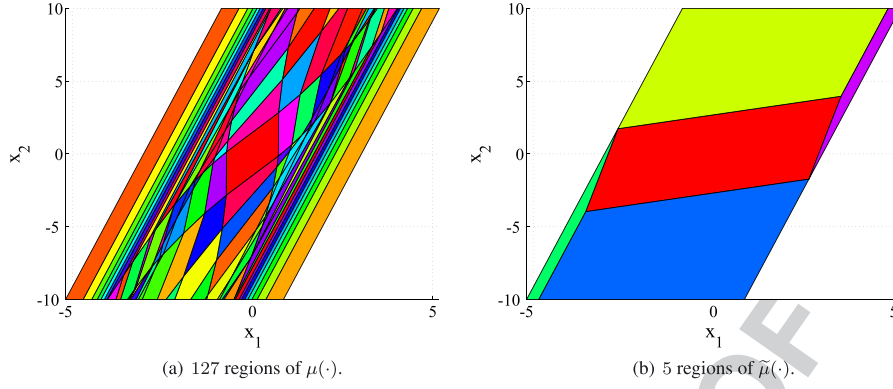


Figure 2. Regions of the complex controller $\mu(\cdot)$ and of the approximate feedback $\tilde{\mu}(\cdot)$.

which is subject to state constraints $-10 \leq x_i(t) \leq 10$, $i \in \mathbb{N}_1^2$ and input bounds $-0.5 \leq u(t) \leq 0.5$. We remark that the system is open-loop unstable with eigenvalues $\lambda_1 = 1.0584$ and $\lambda_2 = -0.6370$. The complex explicit MPC controller $\mu(\cdot)$ in (8) was obtained by solving (6) for $Q_x = I_{2 \times 2}$, $Q_u = 2$, and $N = 20$. Its explicit representation was defined over $M = 127$ polytopic regions $\mathcal{R}_i \subset \mathbb{R}^2$, shown in Figure 2(a). All computations were carried out on a 2.7-GHz CPU using MATLAB and the MPT Toolbox.

To derive a simple representation of the MPC feedback as in (10), we have proceeded as outlined in Section 5. First, we have solved (6) with shorter prediction horizons $\tilde{N} \in \{1, 2, 3, 4\}$. This provided us with simple feedbacks $\hat{\mu}(\cdot)$ as in (14) with lower performances. The domains of these feedbacks were defined, respectively, by $\tilde{M} = \{3, 5, 11, 17\}$ regions $\tilde{\mathcal{R}}_i$. These regions were then employed in (31) to optimize the parameters \tilde{F}_i , \tilde{g}_i of the improved simple feedbacks $\tilde{\mu}(\cdot)$ in (10) while guaranteeing closed-loop stability. The fitting problems (31) were formulated by YALMIP and solved by quadprog.

Remark 6.1

In practice, to get the least complex approximate controller $\tilde{\mu}(\cdot)$, one would only consider the case with the smallest number of regions. We only consider various values of \tilde{M} to assess the suboptimality of $\tilde{\mu}(\cdot)$ with respect to $\mu(\cdot)$ as a function of the number of regions, \tilde{M} .

Next, we have assessed the degradation of performance induced by employing simpler feedbacks $\hat{\mu}(\cdot)$ and $\tilde{\mu}(\cdot)$ instead of the optimal controller $\mu(\cdot)$. To do so, for each suboptimal controller, we have performed closed-loop simulations for 10 000 equidistantly spaced initial conditions from the domain of $\mu(\cdot)$. In each simulation, we have evaluated the performance criterion $J_{\text{sim}} = \sum_{i=1}^{N_{\text{sim}}} x_i^T Q_x x_i + u_i^T Q_u u_i$ for $N_{\text{sim}} = 100$. For each investigated controller, we have subsequently computed mean values of this criterion over all investigated starting points. This ‘average’ performance indicators are denoted in the sequel as J_{opt} for the optimal feedback $\mu(\cdot)$, J_{simple} for the simple, but suboptimal controller $\hat{\mu}(\cdot)$, and J_{improved} for $\tilde{\mu}(\cdot)$, whose parameters were optimized in (31). Then we can express the average suboptimality of $\tilde{\mu}(\cdot)$ by $J_{\text{simple}}/J_{\text{opt}}$ and the suboptimality of $\tilde{\mu}(\cdot)$ by $J_{\text{improved}}/J_{\text{opt}}$, both converted to percentage. The higher the value, the larger the suboptimality of the corresponding controller is with respect to the optimal feedback $\mu(\cdot)$.

Concrete numbers are reported in Table I. As can be observed, lowering the prediction horizon significantly reduces complexity. However, suboptimality is inverse-proportional to complexity. For instance, solving (6) with $N = 1$ gives $\hat{\mu}(\cdot)$ that performs by 60% worse compared with the optimal feedback $\mu(\cdot)$ obtained for $N = 20$. Improving parameters of the feedback function via (31) resulted in an improved controller $\tilde{\mu}(\cdot)$ whose average suboptimality is only 25%. The amount of suboptimality can be further reduced by considering more complex partition of the feedback function. In all cases reported in Table I, the simpler feedback μ

closed-loop stability because the corresponding fitting problems (31) were feasible $\gamma < 1$.

Table I. Complexity and suboptimality comparison for the example in Section 6.1. The (complex) optimal controller consisted of 127 regions.

Prediction horizon	No. of regions	Suboptimality w.r.t. $\mu(\cdot)$ in (8)	
		$\hat{\mu}(\cdot)$ from Lemma 3.1 (%)	$\tilde{\mu}(\cdot)$ from (31) (%)
1	3	60.8	25.1
2	5	32.9	18.0
3	11	11.4	8.3
4	17	6.9	1.7

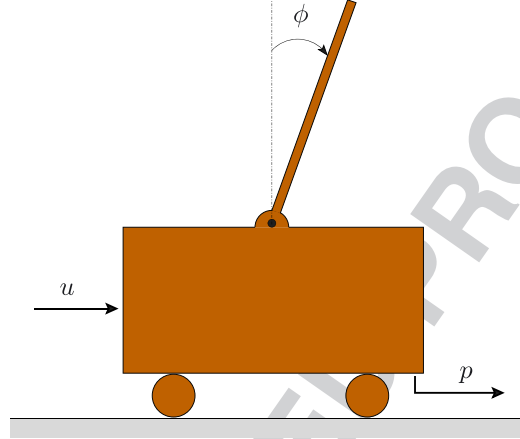


Figure 3. Inverted pendulum on a cart.

6.2. Inverted pendulum on a cart

Next, we consider an inverted pendulum mounted on a moving cart, shown in Figure 3. Linearizing the nonlinear dynamics around the upright, unstable equilibrium leads to the following linear model:

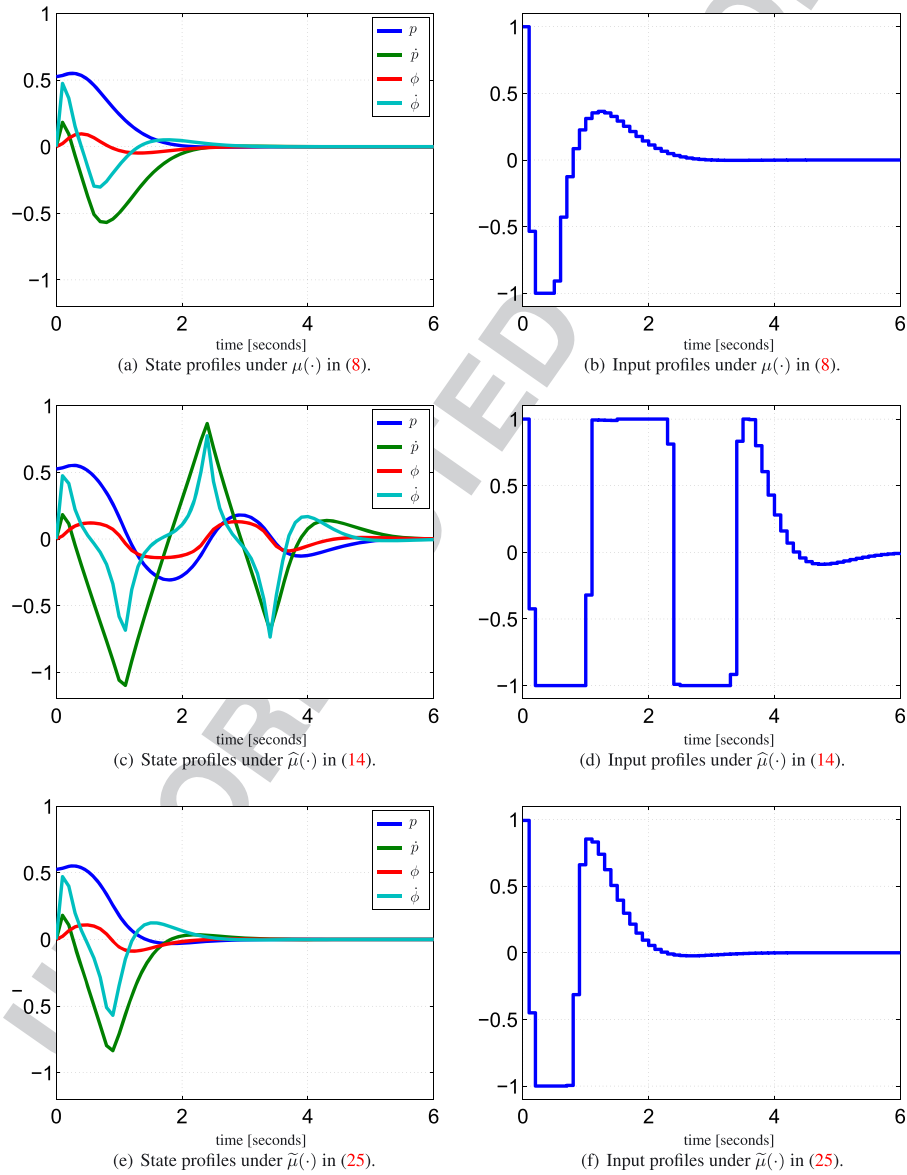
$$\begin{bmatrix} \dot{p} \\ \ddot{p} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.182 & 2.673 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.455 & 31.182 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.818 \\ 0 \\ 4.546 \end{bmatrix} u, \quad (38)$$

where p is the position of the cart, \dot{p} is the cart's velocity, ϕ is the pendulum's angle from the upright position, and $\dot{\phi}$ denotes the angular velocity. The control input u is proportional to the force applied to the cart. System (38) is then converted to (4) by assuming sampling time 0.1 s.

The optimal (complex) controller $\mu(\cdot)$ in (8) was then constructed by solving (6) with prediction horizon $N = 8$, penalties $Q_x = \text{diag}(10, 1, 10, 1)$, $Q_u = 0.1$, and constraints $|p| \leq 1$, $|\dot{p}| \leq 1.5$, $|\phi| \leq 0.35$, $|\dot{\phi}| \leq 1$, $|u| \leq 1$. Using the MPT toolbox, we have obtained $\mu(\cdot)$ defined over 943 polytopes of the four-dimensional state space. Subsequently, we have constructed simple feedbacks $\hat{\mu}(\cdot)$ according to Lemma 3.1 for prediction horizons $\hat{N} \in \{1, 2, 3\}$. This provided us with polytopic partitions $\{\tilde{\mathcal{R}}_i\}$ defined, respectively, by 35 polytopes for $\hat{N} = 1$, 117 regions for $\hat{N} = 2$, and 273 polytopes in case of $\hat{N} = 3$. For each partition, we have then optimized the gains \tilde{F}_i, \tilde{g}_i of $\tilde{\mu}(\cdot)$ in (10) by solving (25). The total runtime of the approximation procedure was 21 s for $\hat{N} = 1$, 58 s for $\hat{N} = 2$, and 116 s for $\hat{N} = 3$. In all cases, the effort for enumeration of vertices and triangulation of polytopes attributed to 40% of the overall runtime, the rest was spent in formulating and solving the QP problems (25).

Table II. Complexity and suboptimality comparison for the example in Section 6.2. The (complex) optimal controller consisted of 943 regions.

Prediction horizon	Number of regions	$\hat{\mu}(\cdot)$ from Lemma 3.1		$\tilde{\mu}(\cdot)$ from (31)	
		AST (s)	Suboptimality (%)	AST (s)	Suboptimality (%)
1	35	8.3	159.4	5.1	59.4
2	117	4.6	43.8	3.7	15.6
3	273	3.5	9.4	3.4	6.3

Figure 4. Simulated closed-loop profiles of pendulum's states and inputs under the (complex) optimal feedback $\mu(\cdot)$ in (8), under the simple controller $\hat{\mu}(\cdot)$ in (14) and under its optimized version $\tilde{\mu}(\cdot)$ obtained from (25).

To assess the degradation of performance induced by employing the simpler controllers instead of the optimal feedback, we have performed 100 closed-loop simulations for various values of the initial cart's position p . Then we have measured the number of simulation steps in which a particular controller drives all states into the ± 0.01 neighborhood of the origin. In other words, our performance evaluation criterion measures liveness properties of a particular controller. The average settling time for the optimal (complex) feedback $\mu(\cdot)$ was 32 sampling times (which corresponds to 3.2 s). The aggregated results showing performance of the two simple feedbacks $\hat{\mu}(\cdot)$ and $\tilde{\mu}(\cdot)$ are reported in Table II. The columns of the table represent, respectively, the prediction horizon \hat{N} and number of polytopes over which both simple controllers are defined, as well as the performance of the simple feedback $\hat{\mu}(\cdot)$ in (14). Here, AST stands for average settling time, and the suboptimality percentage represents the relative increase of the settling time compared with $\text{AST} = 3.2$ s for the optimal (complex) feedback. The final two columns show the performance of $\tilde{\mu}(\cdot)$, whose gains were optimized by (25). As it can be seen, refining the gains \tilde{F}_i, \tilde{g}_i via (25) significantly mitigates the degradation of performance.

To illustrate the differences in the performance of the three controllers, Figure 4 shows the closed-loop profiles of states and inputs under $\mu(\cdot)$, $\hat{\mu}(\cdot)$, and $\tilde{\mu}(\cdot)$ for the initial conditions $p(0) = 0.525$, $\dot{p}(0) = 0$, $\phi(0) = 0$, and $\dot{\phi}(0) = 0$. Here, we have employed the second case of Table II where $\hat{\mu}(\cdot)$ and $\tilde{\mu}(\cdot)$ were both defined over 117 polytopes. Comparing the state profiles in Figure 4(a, c, and e), we can clearly see the benefit of refining the gains of $\tilde{\mu}(\cdot)$ via (25). In particular, the performance of $\tilde{\mu}(\cdot)$ derived according to Section 3.2 is nearly identical to the performance of the optimal (complex) feedback $\mu(\cdot)$. The simple feedback $\hat{\mu}(\cdot)$, on the other hand, performs significantly worse. We remind that in all cases shown in Table II, the complexity of $\tilde{\mu}(\cdot)$ is significantly smaller than the number of regions of the optimal feedback (which was defined over 943 polytopes).

7. CONCLUSIONS

In this paper, we have introduced a novel method for reducing complexity of explicit MPC controllers. The procedure was based on replacing regions of the complex feedback $\mu(\cdot)$ by a simpler partition $\{\tilde{\mathcal{R}}_i\}$, followed by assigning to each region $\tilde{\mathcal{R}}_i$ a local affine expression $\tilde{F}_i x + \tilde{g}_i$ of the simpler feedback $\tilde{\mu}(\cdot)$ such that the reduction of performance with respect to μ is mitigated. The simpler partition was obtained by solving a simpler version of (6) with a lower value of the prediction horizon. Even though by doing so, we already obtain a simpler feedback law $\hat{\mu}(\cdot)$; by using the procedure of Section 3.2, we can significantly reduce the amount of suboptimality (cf. Remark 3.3). We have shown that the search for parameters \tilde{F}_i, \tilde{g}_i in (10) can be formulated as a quadratic optimization problem that entails conditions of recursive feasibility and closed-loop stability. Moreover, we have shown that if only recursive feasibility is required, such a fitting problem is always feasible if a control invariant constraint is employed in (6d). By means of two examples, we have shown that the induced loss of optimality is indeed mitigated. The computationally most challenging part of the approximation procedure is the enumeration of vertices and triangulation of polytopes, both of which are challenging in high dimensions. However, in dimensions below 5 (which are typically considered in explicit MPC), these tasks do not represent a significant obstacle. It is worth noting that the procedures of this paper can be applied to find optimal approximations of arbitrary PWA functions, not necessarily just of control laws. As an example, one can aim at approximating the optimal PWA value function (6a), followed by the reconstruction of the suboptimal control law by interpolation techniques [43].

ACKNOWLEDGEMENTS

J. Holaza, B. Takács and M. Kvasnica gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11 and the financial support of the Slovak Research and Development Agency under the project APVV 0551-11. This research was supported by Mitsubishi Electric Research Laboratories under a Collaborative Research Agreement.

REFERENCES

1. Maciejowski JM. *Predictive Control with Constraints*. Prentice Hall, 2002.
2. Qin SJ, Badgwell TA. A survey of industrial model predictive control technology. 2003; **11**:733–764.
3. Bemporad A, Morari M, Dua V, Pistikopoulos E. The explicit linear quadratic regulator for constrained systems. *Automatica* 2002; **38**(1):3–20.
4. Stewart G, Borrelli F. A model predictive control framework for industrial turbodiesel engine control. Cancun, Mexico, December 2008; 5704–5711.
5. Di Cairano S, Yanakiev D, Bemporad A, Kolmanovsky IV, Hrovat D. Model predictive idle speed control: design, analysis, and experimental evaluation. *IEEE Transactions on Control Systems Technology* 2012; **20**(1):84–97.
6. Di Cairano S, Park H, Kolmanovsky I. Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering. *International Journal of Robust and Nonlinear Control* 2012; **22**(12):1398–1427.
7. Willner L. On parametric linear programming. *SIAM Journal on Applied Mathematics* 1967; **15**(5):1253–1257.
8. Gal T, Nedoma J. Multiparametric linear programming. *Management Science* 1972; **18**:406–442.
9. Borrelli F. *Constrained Optimal Control of Linear and Hybrid Systems*, Vol. 290. Springer-Verlag, 2003.
10. Dua V, Pistikopoulos EN. An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research* 2000; **99**:123–139.
11. Bemporad A, Borrelli F, Morari M. Model predictive control based on linear programming—the explicit solution. *IEEE Transactions on Automatic Control* 2002; **47**(12):1974–1985.
12. Bemporad A, Borrelli F, Morari M. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control* 2003; **48**(9):1600–1606.
13. Spjøtvold J, Tøndel P, Johansen TA. A method for obtaining continuous solutions to multiparametric linear programs. *IFAC World Congress*, Prague, Czech Republic; 2005.
14. Baotić M, Christophersen FJ, Morari M. Constrained optimal control of hybrid systems with a linear performance index. *IEEE Transactions on Automatic Control* 2006; **51**(12):1903–1919.
15. Geyer T, Torrisi F, Morari M. Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica* 2008; **44**(7):1728–1740.
16. Wen C, Ma X, Ydstie BE. Analytical expression of explicit MPC solution via lattice piecewise-affine function. *Automatica* 2009; **45**(4):910–917.
17. Kvasnica M, Fikar M. Clipping-based complexity reduction in explicit mpc. *IEEE Transactions on Automatic Control* 2012; **57**(7):1878–1883.
18. Bemporad A, Filippi C. Suboptimal explicit RHC via approximate multiparametric quadratic programming. 2003; **117**(1):9–38.
19. Cagienard R, Grieder P, Kerrigan E, Morari M. Move blocking strategies in receding horizon control. *Journal of Process Control* 2007; **17**(6):563–570.
20. Grieder P, Kvasnica M, Baotić M, Morari M. Stabilizing low complexity feedback control of constrained piecewise affine systems. *Automatica* 2005; **41**(10):1683–1694.
21. Bayat F, Johansen T. Multi-resolution explicit model predictive control: delta-model formulation and approximation. *IEEE Transactions on Automatic Control* 2013; **58**(11):2979–2984.
22. Valencia-Palomo G, Rossiter J. Using Laguerre functions to improve efficiency of multi-parametric predictive control. *Proceedings of the American Control Conference*, Baltimore, USA, 2010; 4731–4736.
23. Kvasnica M, Löfberg J, Fikar M. Stabilizing polynomial approximation of explicit MPC. *Automatica* 2011; **47**(10):2292–2297.
24. Lu L, Heemels W, Bemporad A. Synthesis of low-complexity stabilizing piecewise affine controllers: a control-Lyapunov function approach. *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, IEEE, 2011; 1227–1232.
25. Bemporad A, Oliveri A, Poggi T, Stora M. Ultra-fast stabilizing model predictive control via canonical piecewise affine approximations. *IEEE Transactions on Automatic Control* 2011; **56**(12):2883–2897.
26. Holaza J, Takács B, Kvasnica M. Synthesis of simple explicit MPC optimizers by function approximation. *Proceedings of the 19th International Conference on Process Control*, Štrbské Pleso, Slovakia, June 18–21 2013; 377–382.
27. Takács B, Holaza J, Kvasnica M, di Cairano S. Nearly-optimal simple explicit MPC regulators with recursive feasibility guarantees. *Conference on Decision and Control*, Florence, Italy, December 2013. p. (accepted).
28. Kvasnica M, Grieder P, Baotić M. Multi-Parametric Toolbox (MPT), 2004. (Available from: <http://control.ee.ethz.ch/mpt/>).
29. Blanchini F, Miani S. *Set-Theoretic Methods in Control*. Birkhauser: Boston, 2008.
30. Dórea C, Hennes J. (A, B) -Invariant polyhedral sets of linear discrete-time systems. *Journal of Optimization Theory and Applications* 1999; **103**(3):521–542.
31. Baotić M. Optimal control of piecewise affine systems – a multi-parametric approach. *Dr. Sc. Thesis*, ETH Zurich, Zurich, Switzerland, March 2005.
32. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. 2002; **38**(1):3–20.
33. Baldoni V, Berline N, De Loera JA, Köppe M, Vergne M. How to integrate a polynomial over a simplex. *Mathematics of Computation* 2010; **80**(273):297.

34. Jones CN, Kerrigan EC, Maciejowski JM. Equality set projection: a new algorithm for the projection of polytopes in halfspace representation. *Technical Report CUED/F-INFENG/TR.463*, Department of Engineering, Cambridge University, UK, 2004. (Available from: <http://www-control.eng.cam.ac.uk/cnj22/>).
35. Jones CN. Polyhedral tools for control. *Ph.D. Dissertation*, University of Cambridge, Cambridge, U.K., July 2005.
36. Lasserre J, Avrachenkov K. The multi-dimensional version of $\int_a^b x^p dx$. *The American Mathematical Monthly* 2001; **108**(2):151–154.
37. Ziegler GM. *Lectures on Polytopes*. Springer, 1994.
38. Lazar M, de la Pena DM, Heemels W, Alamo T. On input-to-state stability of min-max nonlinear model predictive control. *Systems & Control Letters* 2008; **57**:39–48.
39. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press, 2004.
40. Bemporad A. Hybrid Toolbox - User's Guide, 2003. (Available from: <http://www.dii.unisi.it/hybrid/toolbox>).
41. Löfberg J. YALMIP: a toolbox for modeling and optimization in MATLAB. *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. (Available from: <http://users.isy.liu.se/johanl/yalmip/>).
42. Gurobi I. Optimization, "Gurobi optimizer reference manual," 2012. (Available from: <http://www.gurobi.com>).
43. Jones C, Morari M. Polytopic approximation of explicit model predictive controllers. *IEEE Transactions on Automatic Control* 2010; **55**(11):2542–2553.

UNCORRECTED PROOF