

# Constrained Time-Optimal Control of Linear Parameter-Varying Systems

Thomas Besselmann, Johan Löfberg and Manfred Morari

**Abstract**—For linear and hybrid systems, constrained time-optimal control was shown to be a low complexity alternative to the explicit solution of the constrained finite-time-optimal control problem. In this paper we show how Pólya’s relaxation can be employed to compute minimum-time controllers for discrete-time LPV systems. Contrary to previous publications, our approach allows the use of parameter-varying input matrices. In a comparison over 20 random system, it is shown that compared to explicit LPV-MPC the proposed approach achieves similar or even better control performance, while reducing the complexity of the resulting controller up to an order of magnitude.

## I. INTRODUCTION

Linear Parameter-Varying (LPV) systems constitute a powerful system class, incorporating a whole family of linear systems. They are used to model linear systems, whose dynamics depend on external scheduling signals, or to embed nonlinear systems. LPV systems mainly appear in gain-scheduling to expand the applicability of linear control techniques to some extent to nonlinear systems, [1]. Instead of interpolating control laws, the interpolation is brought to the modelling level, allowing for a-priori statements related to stability and control performance. Control of LPV systems stems from robust control and relies on semidefinite programming.

Also the Model Predictive Control (MPC) community has considered LPV systems in the past, enabling finite-horizon optimal control under state and input constraints, [2]. Since MPC requires the solution of an optimization problem in each sampling interval (a semidefinite program in the case of LPV systems, [2]), its application is limited to systems with sufficiently slow dynamics. A remedy for small systems is explicit MPC, the a-priori computation of the optimal control law. Recently, an explicit MPC scheme was proposed for LPV systems, [3]. This scheme is based on multi-parametric programming and thus shares also its shortcomings as high complexity of the resulting control law.

In [4] was shown for linear systems that minimum-time control can be a low complexity alternative for explicit MPC. The complexity of the control laws could be reduced by an order of magnitude, while yielding close to optimal control performance. In minimum-time control, instead of

minimizing an objective function, the number of time steps to converge to a terminal region is minimized. The main benefit is that the required computations are computationally less demanding. Therefore time-optimal control can be applied to systems of higher dimensions and/or for a longer prediction horizon. Moreover, if the control applied inside the terminal region is stabilizing and renders the terminal region invariant, the controller obtained using minimum-time control is stabilizing.

Contrary to previous schemes (e.g. in [5]), we consider the general class of discrete-time LPV systems, i.e. we do not restrict our attention to constant input matrices. The consequences of a varying input matrix are more extensive than one might expect, because it results in the closed-loop system leaving the class of polytopic systems.

## Notation

The following notation is used throughout this paper. Bracketed superscripts  $(\cdot)^{[j]}$  denote indices and not exponents. Calligraphic letters are used for uncountable sets, while fat capital letters are used for sets with a finite number of elements  $\mathbf{X} = \{x_1, \dots, x_n\}$ .  $\mathbf{1}$  denotes a column vector with ones,  $\mathbf{1} = [1, \dots, 1]^T$ ,  $O$  a zero matrix of appropriate dimensions. Element-wise inequalities are denoted by  $\leq$ , while  $\preceq$  denotes matrix inequalities.

## II. PRELIMINARIES

### A. Set-theoretic Basics

This section contains some useful set-theoretic basics for discrete-time systems under the influence of a scheduling parameter. A profound treatment of set-theoretic methods in control can be found in [5], and the references herein. First we consider the discrete-time system

$$x_{k+1} = f(x_k, \theta_k) \quad (1)$$

with the state  $x_k$  under state constraints

$$x_k \in \mathcal{X} \subset \mathbb{R}^{n_x}, \quad (2)$$

and with the scheduling parameter  $\theta_k$  living in a bounded set,

$$\theta_k \in \Theta \subset \mathbb{R}^{n_\theta}. \quad (3)$$

**Definition 1:** (Robust Invariant Set). A set  $\mathcal{O} \subseteq \mathcal{X}$  is said to be a robust (positive) invariant set for system (1) subject to the constraints (2) – (3), if  $\forall x_k \in \mathcal{O}$  also  $x_{k+1} \in \mathcal{O} \forall \theta_k \in \Theta$ .

**Definition 2:** (Maximal Robust Invariant Set  $\mathcal{O}_\infty$ ). The set  $\mathcal{O}_\infty$  is the maximal robust invariant set of system (1) subject to the constraints (2) – (3), if  $\mathcal{O}_\infty$  is robust invariant, contains

J. Löfberg is with the Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, WWW: <http://www.control.isy.liu.se>, Email: [johanl@isy.liu.se](mailto:johanl@isy.liu.se)

Th. Besselmann and M. Morari are with the Automatic Control Laboratory, ETH Zurich, Switzerland, WWW: <http://control.ee.ethz.ch>, Email: [besselmann@control.ee.ethz.ch](mailto:besselmann@control.ee.ethz.ch) | [morari@control.ee.ethz.ch](mailto:morari@control.ee.ethz.ch)

the origin and contains all robust invariant sets in  $\mathcal{X}$  that contain the origin.

The maximal robust invariant (MRI) set  $\mathcal{O}_\infty$  can be computed by making repeated use of presets, see Algorithm 1.

*Definition 3:* (Preset  $\Omega(\mathcal{Z})$ ). The preset  $\Omega(\mathcal{Z})$  contains all states which are mapped to  $\mathcal{Z}$  by system equation (1) subject to the constraints (2) - (3),

$$\Omega(\mathcal{Z}) = \{x \in \mathcal{X} | f(x, \theta) \in \mathcal{Z} \forall \theta \in \Theta\}. \quad (4)$$

The computation of presets for LPV systems will be shown in Section III.

---

**Algorithm 1** Computation of MRI set  $\mathcal{O}_\infty$

---

```

 $\mathcal{Z}_0 = \mathcal{X}$ 
 $\mathcal{Z}_1 = \Omega(\mathcal{Z}_0)$ 
 $i = 1$ 
while  $\mathcal{Z}_i \neq \mathcal{Z}_{i-1}$  do
   $\mathcal{Z}_{i+1} = \Omega(\mathcal{Z}_i)$ 
   $i = i + 1$ 
end while
 $\mathcal{O}_\infty = \mathcal{Z}_i$ 

```

---

Now consider the non-autonomous discrete-time system

$$x_{k+1} = f(x_k, u_k, \theta_k), \quad (5)$$

where the state  $x_k$ , the inputs  $u_k$  and the scheduling parameter  $\theta_k$  are constrained,

$$x_k \in \mathcal{X} \subset \mathbb{R}^{n_x}, \quad u_k \in \mathcal{U} \subset \mathbb{R}^{n_u}, \quad \theta_k \in \Theta \subset \mathbb{R}^{n_\theta}. \quad (6)$$

Analogue to robust invariant sets for system (1), control invariant sets are defined for system (5) subject to the constraints (6). Depending on the availability of the scheduling parameter  $\theta_k$  for feedback, two classes of control invariant sets are distinguished. If the scheduling parameter is unknown or not used for feedback purposes,  $u_k = \mu(x_k)$ , we consider the robust control invariant set.

*Definition 4:* (Robust Control Invariant Set  $\mathcal{C}_R$ ). A set  $\mathcal{C}_R \subseteq \mathcal{X}$  is said to be a robust control invariant set for the system (5) subject to the constraints (6), if  $\forall x_k \in \mathcal{C}_R \exists u_k \in \mathcal{U} : x_{k+1} = f(x_k, u_k, \theta_k) \in \mathcal{C}_R \forall \theta_k \in \Theta$ .

On the other hand, if the scheduling parameter is used for feedback,  $u_k = \mu(x_k, \theta_k)$ , the corresponding set is the gain-scheduling control invariant set.

*Definition 5:* (Gain-Scheduling Control Invariant Set  $\mathcal{C}_{GS}$ ). A set  $\mathcal{C}_{GS} \subseteq \mathcal{X}$  is said to be a gain-scheduling control invariant set for the system (5) subject to the constraints (6), if  $\forall x_k \in \mathcal{C}_{GS} \forall \theta_k \in \Theta : \exists u_k \in \mathcal{U} : x_{k+1} = f(x_k, u_k, \theta_k) \in \mathcal{C}_{GS}$ .

Analogue to Def. 2 the Maximal Robust / Gain-Scheduling Control Invariant Sets are defined.

*Definition 6:* (Maximal Robust Control Invariant Set  $\mathcal{C}_{R,\infty}$ ). The set  $\mathcal{C}_{R,\infty}$  is said to be the maximal robust control invariant set for the system (5) subject to the constraints (6), if  $\mathcal{C}_{R,\infty}$  is robust control invariant and contains all robust control invariant sets contained in  $\mathcal{X}$ .

*Definition 7:* (Maximal Gain-Scheduling Control Invariant Set  $\mathcal{C}_{GS,\infty}$ ). The set  $\mathcal{C}_{GS,\infty}$  is said to be the maximal

gain-scheduling control invariant set for the system (5) subject to the constraints (6), if  $\mathcal{C}_{GS,\infty}$  is gain-scheduling control invariant and contains all gain-scheduling control invariant sets contained in  $\mathcal{X}$ .

Since we are interested in control of LPV systems, where the scheduling parameter  $\theta_k$  is known and used for feedback, we restrict our attention from now on to the gain-scheduling case, and drop the subscript *GS*. Analogue to Def. 3, the definition of presets can be stated for the non-autonomous system (5) under gain-scheduling control.

*Definition 8:* (Preset  $\Omega(\mathcal{Z})$ ). For non-autonomous systems (5) subject to the constraints (6), the preset  $\Omega(\mathcal{Z})$  of a set  $\mathcal{Z}$  is defined as

$$\Omega(\mathcal{Z}) = \{x \in \mathcal{X} | \forall \theta \in \Theta \exists u \in \mathcal{U} : f(x, u, \theta) \in \mathcal{Z}\}. \quad (7)$$

The maximal gain-scheduling control invariant (MGSCI) set  $\mathcal{C}_\infty$  can be computed using Algorithm 2.

---

**Algorithm 2** Computation of MGSCI set  $\mathcal{C}_\infty$

---

```

 $\mathcal{Z}_0 = \mathcal{X}$ 
 $\mathcal{Z}_1 = \Omega(\mathcal{Z}_0)$ 
 $i = 1$ 
while  $\mathcal{Z}_i \neq \mathcal{Z}_{i-1}$  do
   $\mathcal{Z}_{i+1} = \Omega(\mathcal{Z}_i)$ 
   $i = i + 1$ 
end while
 $\mathcal{C}_\infty = \mathcal{Z}_i$ 

```

---

## B. Minimum Time Control

This section restates the computation of time-optimal controllers for linear discrete-time systems under the influence of a scheduling parameter. The task to reach a target set  $\mathcal{T}$  in the minimal number of time steps can be stated as

$$\min_{[\mathbf{U}, N]} N \quad (8a)$$

$$\text{s.t.} \quad \forall \theta_k \in \Theta, k = 0, \dots, N-1 : \quad (8b)$$

$$x_{k+1} = f(x_k, u_k, \theta_k), k = 0, \dots, N-1 \quad (8c)$$

$$x_k \in \mathcal{X}, u_k \in \mathcal{U}, k = 0, \dots, N-1 \quad (8d)$$

$$x_N \in \mathcal{T}, \quad (8e)$$

where the control inputs are merged to  $\mathbf{U} = \{u_0, \dots, u_{N-1}\}$ . The optimization problem (8) can be solved by applying Algorithm 2, initialized with  $\mathcal{Z}_0 = \mathcal{T}$ . In each iteration the set  $\mathcal{Z}_i$  is stored together with the control law  $u = \mu_i(x, \theta)$ , which steers the state into  $\mathcal{Z}_{i-1}$ . Online, we measure or estimate state  $x_0$  and scheduling parameter  $\theta_0$ , solve

$$i^* = \min_i i \quad (9a)$$

$$\text{s.t.} \quad x_0 \in \mathcal{Z}_i, \quad (9b)$$

and apply the control law  $u = \mu_{i^*}(x_0, \theta_0)$  associated to  $\mathcal{Z}_{i^*}$ . This procedure guarantees the convergence of (5) to the target region  $\mathcal{T}$  in the minimal number of time steps. Moreover, if a stabilizing control law is applied for states inside  $\mathcal{T}$ , and if  $\mathcal{T}$  is robust invariant under this control, then

feasibility of the minimum-time controller implies recursive constraint satisfaction and stability.

### C. Pólya's Relaxation

Pólya's relaxation allows the reformulation of constraints, which depend polynomially on the scheduling parameter into sufficient parameter-independent constraints. The relaxation is based on Pólya's theorem.

**Theorem 1: Pólya's theorem.** If a homogeneous polynomial  $p(\theta)$  is positive on the standard simplex  $\Theta$ , all coefficients of  $p_{N_p}(\theta) = p(\theta) \cdot (\sum_{j=1}^{n_\theta} \theta^{[j]})^{N_p}$  are positive for a sufficiently large Pólya degree  $N_p$ .

*Proof:* See [6], [7].  $\blacksquare$

In the remaining part of the paper, we will repeatedly make use of the more obvious reverse of Pólya's theorem<sup>1</sup>, i.e., positive coefficients of the extended polynomial mean positivity over the whole simplex. By applying Pólya's relaxation, we mean the processing of the following steps:

- 1) Reformulate constraints which are polynomial in the scheduling parameter into a positivity constraint of a polynomial  $p(\theta)$ .
- 2) Homogenize the polynomial  $p(\theta)$  by multiplying single monomials with  $\sum_{j=1}^{n_\theta} \theta^{[j]} (= 1)$  until all monomials have the same degree.
- 3) Set the Pólya degree  $N_p$ , and compute the coefficients  $c_{N_p}$  of the extended polynomial  $p_{N_p}(\theta) = p(\theta) \cdot (\sum_{j=1}^{n_\theta} \theta^{[j]})^{N_p}$ . In this step some conservatism may be introduced depending on the selection of  $N_p$ . By increasing the polynomial degree  $N_p$ , the relaxations become tighter until the exact problem is considered.
- 4) Replace the polynomial constraint by  $c_{N_p} > 0$ .

## III. MAIN RESULTS

From now on, we consider the LPV system

$$x_{k+1} = A(\theta_k)x_k + B(\theta_k)u_k. \quad (10)$$

subject to polytopic state and input constraints

$$x_k \in \mathcal{X} = \{x | H_x x \leq \mathbf{1}\}, \quad u_k \in \mathcal{U} = \{u | H_u u \leq \mathbf{1}\}, \quad (11)$$

containing the origin in their interiors. The scheduling parameter vector  $\theta_k = [\theta_k^{[1]}, \dots, \theta_k^{[n_\theta]}]^T$  is measured online. Future values are however only known to lie in a standard  $n_\theta$ -simplex  $\Theta$  in the parameter space,

$$\theta_k \in \Theta := \{\theta_k \in \mathbb{R}_+^{n_\theta} | \sum_{j=1}^{n_\theta} \theta_k^{[j]} = 1\}. \quad (12)$$

The parameter-varying matrices  $A(\theta_k)$  and  $B(\theta_k)$  are known to lie in polytopes,

$$A(\theta_k) = \sum_{j=1}^{n_\theta} A_j \theta_k^{[j]}, \quad B(\theta_k) = \sum_{j=1}^{n_\theta} B_j \theta_k^{[j]}, \quad (13)$$

where  $A_j$  and  $B_j$  denote the  $j$ -th vertices of the polytopes. This polytopic description is a common assumption in the

LPV framework, see e.g. [1]. For the control problem to make sense, it is assumed that the system (10) is controllable (and observable) for all  $\theta_k \in \Theta$ , see [9], [10].

The objective is to compute a stabilizing minimum-time controller as in Section II-B for the LPV system (10) which takes knowledge of the scheduling parameter into account,

$$u_k = \mu(x_k, \theta_k). \quad (14)$$

The computation of a minimum-time controller for the LPV system (10) consists of three steps, in order to determine (i) a terminal controller for the terminal region  $\mathcal{T}$ , (ii) the terminal region  $\mathcal{T}$  itself, which should be robust invariant under the terminal control, and (iii) the control laws  $\mu_i(x, \theta)$  which steer the state in the minimal number of steps to  $\mathcal{T}$ .

Unfortunately, taking the scheduling parameter into account (14) renders the closed-loop system polynomially<sup>2</sup> dependant on the scheduling parameter, such that the set of states which can be mapped into a polytope under (14) is an intersection of infinitely many polytopes, thus convex, but not necessarily a polytope itself. In the following we propose to employ Pólya's relaxation (Section II-C) to approximate the MRI and MGSCI set of the LPV system (10) by polytopic subsets, and to use these polytopic sets within the minimum-time framework.

### A. Terminal Region Control

The first step of the computations is concerned with the control of the unconstrained system, which is to be applied in an environment around the origin where no constraints are active. In principle, every control technique for unconstrained LPV systems can be employed here, assumed it is asymptotically stable and renders a set of states robust invariant. A straight-forward choice is the computation of LQR controllers for each vertex of the parameter simplex (12), and to interpolate the obtained feedback matrices,

$$u_k = K(\theta)x_k = \sum_{j=1}^{n_\theta} K_j \theta_k^{[j]} x_k. \quad (15)$$

In many cases this will result in a controller with satisfying stability and performance properties. On the other hand, guarantees can only be given for the vertices of the parameter simplex, while stability and performance might decrease for scheduling parameter values in the relative interior of the simplex. Therefore we recommend to compute the state feedback matrices  $K_j$  in (15) by an approach similar to the computation of discrete-time LQR controllers, which takes the whole parameter simplex into account. This approach is facilitated by a matrix-valued version of Pólya's relaxation, [11]. We start with the assumption of a quadratic cost function  $J = x_k^T P x_k$  with  $P$  symmetric and positive definit, and require that the cost decreases in each step at least by a quadratic stage cost,

$$x_{k+1}^T P x_{k+1} \leq x_k^T P x_k - x_k^T Q x_k - u_k^T R u_k. \quad (16)$$

<sup>1</sup>The presented usage of Pólya's theorem is implemented in YALMIP as one of the so called filters in the robust optimization framework, [8].

<sup>2</sup>Assumed (14) is an affine or polynomial function of the scheduling parameter.

Inserting the state update equation (10) and the state feedback equation (15) yields

$$x_k^T [(A(\theta_k) + B(\theta_k)K(\theta_k))^T P (A(\theta_k) + B(\theta_k)K(\theta_k)) - P + Q + K(\theta_k)^T R K(\theta_k)] x_k \leq 0 \quad \forall \theta_k \in \Theta. \quad (17)$$

For Eq. (17) to be fulfilled, the matrix inequality

$$[(A(\theta_k) + B(\theta_k)K(\theta_k))^T P (A(\theta_k) + B(\theta_k)K(\theta_k)) - P + Q + K(\theta_k)^T R K(\theta_k)] \preceq 0 \quad \forall \theta_k \in \Theta \quad (18)$$

must hold. This matrix inequality is not linear in the matrix variables  $K(\theta_k)$  and  $P$ . Substituting  $Y := P^{-1}$  and  $F(\theta_k) := K(\theta_k)Y$ , and multiplying from left and right with  $Y$  yields

$$[(A(\theta_k)Y + B(\theta_k)F(\theta_k))^T Y^{-1} (A(\theta_k)Y + B(\theta_k)F(\theta_k)) - Y P Y + Y Q Y + F(\theta_k)^T R F(\theta_k)] \preceq 0 \quad \forall \theta_k \in \Theta. \quad (19)$$

Applying Schur complement we obtain the linear matrix inequality (LMI) (20) which inherits the requirements  $Y \succeq 0, Q \succeq 0, R \succeq 0$ . The LMI (20) is linear in the matrix variables  $Y$  and  $F(\theta_k)$ , but still depends polynomially on the scheduling parameter  $\theta_k$ . Hence we apply Pólya's relaxation to obtain an LMI independent of the scheduling parameter  $\theta_k$ . In the semidefinite program we can minimize the trace or the largest eigenvalue of  $P$ , to obtain the cost function with the minimum expected/worst-case cost. The Pólya degree  $N_p$  can be chosen high since there is a strong influence on the performance of  $K(\theta)$ , but only indirect influence on complexity through the size of the terminal region  $\mathcal{T}$ , which depends on (15).

*Remark 1:* It is also possible to use not a constant  $Y$ , but a polynomial function of the parameter  $Y(\theta_k)$ . This would reduce the conservatism of the approach, but at the same time lead to a controller  $K(\theta_k)$ , which is a rational function of the scheduling parameter.

## B. Terminal Region

In the second step the terminal region  $\mathcal{T}$  is determined. The terminal region is required to be robust invariant under the unconstrained control law (15), and represents a set of states where neither state nor input constraints (11) are active. The terminal region  $\mathcal{T}$  can be computed by Algorithm 1 combined with Pólya's relaxation. If the polytopic set  $\mathcal{Z}_i = \{x | H_i x \leq 1\}$  is given, its preset  $\Omega(\mathcal{Z}_i)$  is

$$\Omega(\mathcal{Z}_i) = \left\{ x \left| \begin{array}{l} H_x x \leq 1, \\ H_u K(\theta) x \leq 1, \\ H_i (A(\theta) + B(\theta)K(\theta)) x \leq 1, \\ \forall \theta \in \Theta \end{array} \right. \right\}. \quad (21)$$

Using Pólya's relaxation, the constraints of (21) can be transformed into parameter-independent constraints, and we obtain a polytopic approximation of this preset, namely  $\Omega_{N_p}(\mathcal{Z}_i) = \{H_{i+1} x \leq 1\}$ .

*Definition 9:* (Polytopic Preset  $\Omega_{N_p}(\mathcal{Z}_i)$ ). A set  $\Omega_{N_p}(\mathcal{Z}_i)$  is called polytopic preset of the polytope  $\mathcal{Z}_i$  for the system (10) subject to the constraints (11), if it contains all states

which fulfil the constraints arising when applying Pólya's relaxation with degree  $N_p$  to the preset  $\Omega(\mathcal{Z}_i)$ .

For  $\Omega_{N_p}(\mathcal{Z}_i)$  again the polytopic preset  $\Omega_{N_p}(\Omega_{N_p}(\mathcal{Z}_i))$  can be computed. Repeated computation of polytopic presets within Algorithm 1 yields the polytopic approximate maximum robust invariant (AMRI) set  $\mathcal{O}_{\infty}^{N_p}$ .

The AMRI sets  $\mathcal{O}_{\infty}^{N_p}$  are robust invariant and can be taken as terminal region  $\mathcal{T}$ . The choice of the Pólya degree  $N_p$  reflects a trade-off between the number of facets of  $\mathcal{O}_{\infty}^{N_p}$  and its size. It influences not only the shape of the terminal region  $\mathcal{T}$ , but also the complexity of the succeeding control laws  $\mu_i(x, \theta)$ .

Note that the robust invariance property ensures that the state remains inside  $\mathcal{T}$  once it has entered the terminal region, and that this property holds independent of the actual scheduling parameter values and no matter, how the state feedback matrices (15) were obtained.

## C. Minimum Time Iterations

After determining an invariant terminal region and terminal region control, the control laws  $\mu_i(x, \theta)$ , which steer the state in the minimal number of time steps to the terminal region  $\mathcal{T}$ , are computed. For this purpose Algorithm 2 is applied, while Pólya's relaxation is employed for the computation of polytopic presets.

In the proposed procedure, the dependence of the input on the scheduling parameter has to be set beforehand. This dependence can be polynomial, but due to complexity reasons, we recommend to choose an affine dependence,

$$u_k = \sum_{j=1}^{n_{\theta}} \theta_k^{[j]} u_k^{[j]}(x_k). \quad (22)$$

For notational simplicity we unite the vertex control laws to  $\mathbf{U}_k = [u_k^{[1]}, \dots, u_k^{[n_{\theta}]}]$ .

The control law  $\mu_i(x, \theta)$ , which steers the state from  $\mathcal{Z}_{i+1}$  to  $\mathcal{Z}_i$ , is not unique, but can be selected from a set of control laws. For this selection the following multi-parametric program can be employed,

$$\min_{\mathbf{U}_k} J(x_k, u_k) \quad (23a)$$

$$\text{s.t.} \quad H_x x_k \leq 1, \quad (23b)$$

$$H_u u_k \leq 1, \quad (23c)$$

$$H_i (A(\theta_k) x_k + B(\theta_k) u_k) \leq 1, \quad (23d)$$

$$\forall \theta_k \in \Theta, \quad (23e)$$

where the objective function  $J(x_k, u_k)$  determines the preferred selection. A possible objective function is

$$J(x_k, u_k) = \max_{\theta_k} \|Q x_{k+1}\|_p + \|R u_k\|_p, \quad (24)$$

where  $p$  denotes a polytopic norm. This objective function represents a minimization of the worst-case cost during a one-step-ahead prediction. Analogue to [3], an epigraph reformulation can be used to handle the maximization over the scheduling parameter.

It is possible, but not necessary, to minimize the worst case, since convergence to  $\mathcal{T}$  is guaranteed, and the choice

$$\begin{bmatrix} -Y & A(\theta_k)Y + B(\theta_k)F(\theta_k) & O & O \\ (A(\theta_k)Y + B(\theta_k)F(\theta_k))^T & -Y & Y & F(\theta_k)^T \\ O & Y & -Q^{-1} & O \\ O & F(\theta_k) & O & -R^{-1} \end{bmatrix} \preceq 0 \quad \forall \theta_k \in \Theta, \quad (20)$$

of the objective function  $J(x_k, u_k)$  can focus entirely on control performance. When common operating points of the scheduling parameter  $[\bar{\theta}_1, \dots, \bar{\theta}_{n_{\bar{\theta}}}]$  (e.g. the vertices) are known, it is also possible to minimize the cost predictions for these operating points,

$$\begin{aligned} & J(x_k, u_k) \\ = & \sum_{r=1}^{n_{\bar{\theta}}} (A(\bar{\theta}_r)x_k + B(\bar{\theta}_r)u_k)^T P (A(\bar{\theta}_r)x_k + B(\bar{\theta}_r)u_k) \\ & + u_k^T R u_k. \end{aligned} \quad (25)$$

Since an epigraph formulation is not necessary, this objective function can be quadratic. The matrix  $P$  can be taken from the terminal region control, s.t. (25) reflects the infinite horizon cost under the assumption of unconstrained state evolution.

Compared to explicit LPV-MPC, which was presented in [3], (23) comprises significantly fewer constraints, mainly because the objective function takes only the current and the succeeding step directly into account, and thus has a simpler structure. This is the main reason for the low complexity of minimum-time control compared to explicit LPV-MPC. Additionally, the future scheduling parameter values are unknown, what can lead to a broad variety of possible future trajectories, such that the exact future costs are simply not available. Hence it is questionable if much effort should be spent on a trade-off between current and future costs, or if a simple approximation of the future costs as in (25) is not sufficient.

The Pólya degree  $N_p$ , which is applied to (23), is not only determining the size and number of constraints of the polytopic preset  $\Omega_{N_p}(\mathcal{Z}_i)$ , but also has a direct impact on the number of constraints of the multi-parametric program (23), and thus on the complexity of the resulting control laws  $\mu_i(x_k, \theta_k)$ . Hence it should be selected rather low.

*Remark 2:* It should be noted that the resulting control law is not a minimum-time control law in a strict sense, since there might be states in the preset, which do not belong to the polytopic preset, and thus are not mapped to the target region in the minimum number of steps.

#### IV. NUMERICAL EXAMPLES

In order to justify claims about complexity and control performance, we compared the proposed minimum-time scheme to explicit LPV-MPC by means of 20 random examples. This comparison is by no means a proof, but can serve as an indicator of to be expected properties of the minimum time scheme. For the computations we employed the NAG solver under MPT 2.6, [12], interfaced via YALMIP, [8].

The example systems consist of two states and one input. The scheduling parameters live in standard 2-simplices, and

the elements of the system matrices are uniformly distributed random numbers in the interval  $[-1,1]$ , i.e. also unstable systems are considered. The state and input constraints were chosen to be

$$\mathcal{X} = \{x \mid \|x\|_{\infty} \leq 10\}, \quad \mathcal{U} = \{u \mid |u| \leq 1\}. \quad (26)$$

Three methods were compared. The first is the minimum-time scheme proposed in this paper, denoted by mt. The terminal region control was computed by solving the LMI (20), while the cost function (25) penalizing the vertex predictions was minimized during the minimum-time iterations. The Pólya degree of the relaxations was  $N_p = 10$  for the terminal region controller, and  $N_p = 2$  for the terminal region and minimum-time iterations.

The second method is the explicit LPV-MPC scheme from [3], minimizing the  $\infty$ -norm of finite-horizon predictions and denoted by eMPCz. We were also interested in the role of the uncontrolled successor state (uss)  $z$  in eMPCz, thus we included as a third method an analogue explicit LPV-MPC scheme denoted by eMPCx, which does not utilize the uncontrolled successor state. The Pólya degree in both methods was set to  $N_p = 2$ , and the matrix  $P$  from (20) was used in the terminal cost.

The setup of the three methods is as follows. The stage cost matrices are

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1. \quad (27)$$

The prediction horizon (and the number of minimum-time iterations, respectively) was set to  $N = 3$ . In all three methods the control inputs depend affinely on the scheduling parameter. In order to ensure a fair comparison between the number of regions, the terminal region of the minimum-time controller was included as terminal region constraints in the other two methods. Thus the feasible region in principle is the same for all three methods, and only the use of the uss  $z$  changes the feasible region for eMPCz.

For all 20 systems, control laws were computed employing the three mentioned methods. The computation of eMPCz and eMPCx for systems 13 and 19 was aborted after several hours. The complexity of the remaining control laws is depicted in Fig. 1. The complexity of mt varies from about half the complexity of eMPCz, up to an order of magnitude difference (system 2). The complexity of eMPCx is usually higher than of eMPCz, even significantly, but there are also systems, where eMPCx is less complex than eMPCz (systems 9, 20). For all systems, the offline computation time for mt was way shorter than for eMPCz and eMPCx.

In order to compare the control performance of the three methods, the state space was gridded into 900 points, and each point served as initial state for the systems under one

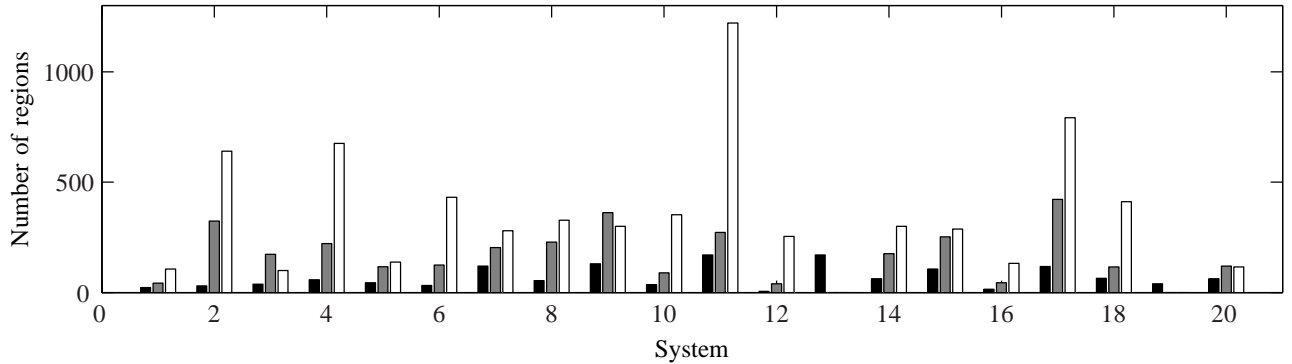


Fig. 1: Number of regions of the control laws mt (black), eMPCz (gray) and eMPCx (white) for 20 example systems.

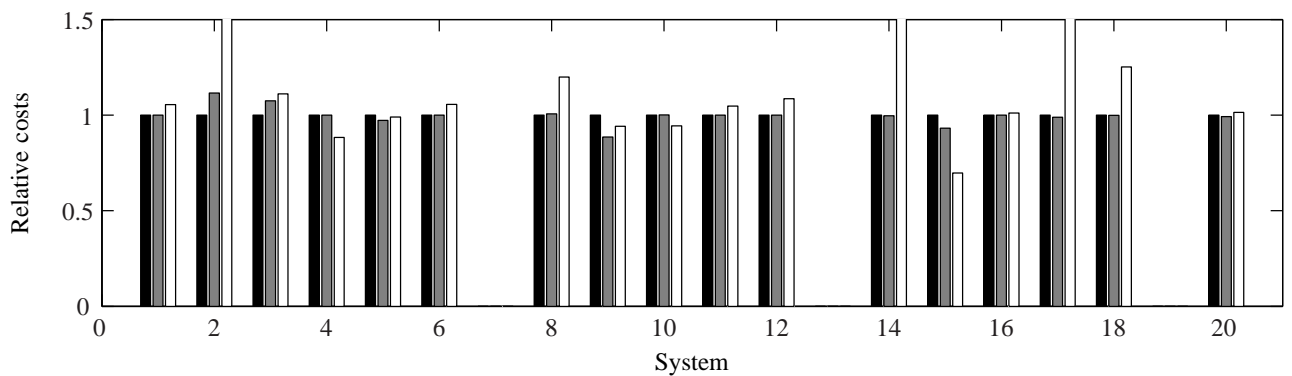


Fig. 2: Relative average costs under the control laws mt (black), eMPCz (gray) and eMPCx (white) for 20 example systems.

of the three control methods. The closed-loop system was simulated for 40 time steps, and the  $\infty$ -norm of the actual stage costs were added up. The scheduling parameter values at each time step were selected randomly in the parameter simplex. These simulations were repeated ten times with different scheduling parameter values (all control methods were confronted with the same scheduling parameter values, of course). Finally the average cost values over all grid points and all simulations were taken. The average costs relative to the costs of mt are depicted in Fig. 2. For system 7, eMPCx is not stabilizing, therefore system 7 is not included in Fig. 2. Otherwise it can be seen that eMPCz usually outperforms eMPCx, but sometimes yields worse performance (systems 4,10,15). The control performance of mt is often comparable to eMPCz, though there also exist systems where one method outperforms the other.

What can be drawn as conclusions? First that minimum-time MPC represents also for LPV systems a low complexity alternative to finite-horizon optimal control with intriguing control performance. Occasionally it even outperforms explicit LPV-MPC, likely because the terminal cost matrices  $P$  were adapted to quadratic costs and because explicit LPV-MPC is optimizing worst-case costs instead of average costs. Additionally the benefit of using the  $uss\ z$  within explicit LPV-MPC was demonstrated, in average improving control performance with lower complexity. Hence the  $uss\ z$  should only be disregarded in exceptions, or to avoid a possible reduction of the set of feasible solutions.

## REFERENCES

- [1] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled  $H_\infty$  control of linear parameter-varying systems: A design example," *Automatica*, vol. 31, no. 9, pp. 1251 – 1261, 1995.
- [2] Y. Lu and Y. Arkun, "Quasi-min-max MPC algorithms for LPV systems," *Automatica*, vol. 36, pp. 527 – 540, 2000.
- [3] T. Besselmann, J. Löfberg, and M. Morari, "Explicit model predictive control for linear parameter-varying systems," in *Proc. of the Conf. on Decision & Control*, Cancun, Mexico, 2008.
- [4] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," in *Proc. of the Conf. on Decision & Control*, Maui, Hawaii, USA, dec 2003, pp. 3179–3184.
- [5] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, T. Basar, Ed. Boston: Birkhäuser, 2008.
- [6] G. Pólya, "Über positive Darstellung von Polynomen," *Vierteljahrsschrift d. naturforschenden Gesellschaft in Zürich*, vol. 73, pp. 141 – 145, 1928, reprinted in: *Collected Papers, Volume 2*, 309 – 313, Cambridge: MIT Press, 1974.
- [7] V. Powers and B. Reznick, "A new bound for Pólya's theorem with applications to polynomial positive on polyhedra," *Journal of Pure and Applied Algebra*, vol. 164, pp. 221 – 229, 2001.
- [8] J. Löfberg, "Modeling and solving uncertain optimization problems in YALMIP," *IFAC World Congress*, 2008.
- [9] L. Silverman and H. Meadows, "Controllability and observability in time-variable linear systems," *SIAM Journal on Control*, vol. 5, no. 1, pp. 64 – 73, 1967.
- [10] G. Balas, J. Bokor, and Z. Szabó, "Invariant subspaces for LPV systems and their applications," *IEEE Trans. on Automatic Control*, vol. 48, no. 11, pp. 2065 – 2069, Nov. 2003.
- [11] C. Scherer and C. Hol, "Asymptotically exact relaxations for robust LMI problems based on matrix-valued sum-of-squares," in *Proc. of the Mathematical Theory of Networks and Systems*, Leuven, Belgium, July 2004.
- [12] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari, "Multi-Parametric Toolbox (MPT)," in *Hybrid Systems: Computation and Control*, Mar. 2004, pp. 448–462.