

Explicit MPC for LPV Systems: Stability and Optimality

Thomas Besselmann, Johan Löfberg and Manfred Morari *Fellow, IEEE*

Abstract—This paper considers high-speed control of constrained linear parameter-varying systems using model predictive control. Existing model predictive control schemes for control of constrained linear parameter-varying systems typically require the solution of a semi-definite program at each sampling instance. Recently, variants of explicit model predictive control were proposed for linear parameter-varying systems with polytopic representation, decreasing the online computational effort by orders of magnitude. Depending on the mathematical structure of the underlying system, the constrained finite-time optimal control problem can be solved optimally, or close-to-optimal solutions can be computed. Constraint satisfaction, recursive feasibility and asymptotic stability can be guaranteed a-priori by an appropriate selection of the terminal state constraints and terminal cost. The paper at hand gathers previous developments and provides new material such as a proof for the optimality of the solution, or, in the case of close-to-optimal solutions, a procedure to determine a bound on the suboptimality of the solution.

Index Terms—Explicit model predictive control, linear parameter-varying systems, constrained control, optimal control, dynamic programming.

I. INTRODUCTION

LINEAR Parameter-Varying (LPV) systems are an important system class, whose dynamics depend linearly on the state and input of the system, but also on some scheduling parameter. Hence an LPV system describes a *family* of linear systems. The LPV paradigm states that no *a-priori* information about the scheduling parameter values is available, but that the parameter can be measured or estimated online. The interest in LPV systems is motivated by their use in gain-scheduling control techniques, and by the possibility to embed nonlinear systems into the LPV framework by covering nonlinearities within the scheduling parameter, [1]. Therefore the LPV framework enables, to some extent, the application of linear control methods to nonlinear systems, while providing rigorous statements on stability and performance of the closed-loop system, [2], [3].

Model Predictive Control (MPC) is a popular control method both in theory and in practice, [4], [5]. In MPC the *control* inputs are computed by repeatedly solving optimization problems which incorporate finite-horizon *predictions* based on a discrete-time *model* of the system. MPC is among the most widely applied control methods, especially in the

field of process control. This is mainly due to its flexibility, and to the ability to take constraints on the input and/or states directly into account.

The main drawback of MPC is, in general, the computational effort required to solve the constrained finite-time optimal control (CFTOC) problem at each sampling instance. This effort can prevent the application of MPC to systems with a high sampling rate, or at least makes such an application expensive, since the necessary computational equipment has to be provided.

A remedy for this problem was proposed in form of *explicit MPC*, an alternative way to solve the underlying CFTOC problem. Explicit MPC is based on parametric programming, which allows to solve linear and quadratic programs for an entire set of parameters, [6]. Instead of solving the CFTOC problem repeatedly online, the optimal inputs to the system are precomputed as a piece-wise affine functions of the state and stored in a look-up table. Online only the look-up table has to be evaluated, which allows the application of MPC to systems with sampling rates in the range of microseconds, [7], [8]. Efficient algorithms for the computation of explicit solutions to the CFTOC problem were developed for linear systems, [9], for hybrid systems, [10] and for uncertain linear systems, [11]. Recently, three variants of explicit MPC were proposed also for LPV systems with polytopic representation, [12]–[14], all handling different assumptions on the LPV system at hand. The contribution of this paper is to deliver a consolidated treatment of the topic and to complement the proposed methods by a-priori stability guarantees, a proof for the optimality of an affine input parametrization for the case of constant input matrix, and a procedure to determine bounds on the suboptimality in the case of a varying input matrix.

This publication is structured as follows. Firstly the considered problem is stated. In Section III, the special case of constant input matrices is elaborated. Section IV tackles the more general case of a varying input matrix. Guarantees for closed-loop stability under the proposed methods are discussed in Section V. Afterwards numerical examples are presented, to demonstrate the application of the proposed methods.

II. PROBLEM STATEMENT

We consider discrete-time LPV systems of polytopic representation, which are defined by the state-update equation

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

T. Besselmann is with the Control and Optimization Group, ABB Corporate Research, 5405 Baden-Dättwil, Switzerland (email: besselmann@control.ee.ethz.ch)

J. Löfberg is with the Division of Automatic Control, Linköping University, Sweden (email: johanl@isy.liu.se)

M. Morari is with the Automatic Control Laboratory, ETH Zurich, 8092 Zurich, Switzerland (email: morari@control.ee.ethz.ch)

with

$$A(\theta_k) = \sum_{j=1}^{n_\theta} A_j \theta_{k,j}, \quad B(\theta_k) = \sum_{j=1}^{n_\theta} B_j \theta_{k,j}, \quad (1b)$$

and

$$\theta_k \in \Theta := \left\{ \theta_k \in \mathbb{R}_+^{n_\theta} \mid \sum_{j=1}^{n_\theta} \theta_{k,j} = 1 \right\}. \quad (1c)$$

The discrete time is denoted by $k \in \mathbb{Z}$, while the variables $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$ and $\theta_k \in \mathbb{R}_+^{n_\theta}$ denote the state, control input and time-varying scheduling parameter, respectively. The system matrices $A: \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x \times n_x}$ and $B: \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x \times n_u}$ are known to lie in polytopes, where $A_j \in \mathbb{R}^{n_x \times n_x}$, $B_j \in \mathbb{R}^{n_x \times n_u}$ correspond to the j th vertex of the parameter simplex Θ . This polytopic description is a common assumption in the LPV framework, see e.g. [3].

Furthermore, the system is constrained, $x_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$. The constraint sets \mathbb{X} and \mathbb{U} are assumed to be polytopes including the origin in its interior,

$$\mathbb{X} := \{x \in \mathbb{R}^{n_x} \mid E_x x \leq \mathbf{1}\}, \quad \mathbb{U} := \{u \in \mathbb{R}^{n_u} \mid E_u u \leq \mathbf{1}\}, \quad (2)$$

with E_x, E_u being real matrices of suitable dimensions.

Remark 1: We restrict ourselves to separate constraints on the state and inputs in (2) solely for ease of notation. It is straight-forward to modify the presented algorithms in this paper to the case of mixed constraints, i.e. $\tilde{E}_x x + \tilde{E}_u u \leq \mathbf{1}$.

It is assumed that the LPV paradigm holds, i.e. that the current scheduling parameter is known, while future values of the scheduling parameter are unknown. For the control problem to make sense, it is further assumed that the LPV system (1) is controllable (and observable) for all $\theta_k \in \Theta$, see [15], [16].

For the described class of systems we want to compute an explicit parameter-dependent state-feedback control law,

$$u_k = \mu(x_k, \theta_k), \quad (3)$$

which makes use of knowledge of the current scheduling parameter θ_k . In order to compute this control law (3) within a Model Predictive Control scheme, a cost function based on finite-horizon predictions is to be minimized. These predictions over a horizon of length N depend on the unknown sequence of future scheduling parameters $\mathbf{T}_k := \{\theta_{k+1}, \dots, \theta_{k+N-1}\} \in \Theta \times \dots \times \Theta =: \Theta^{N-1}$ and the sequence of control laws $\boldsymbol{\pi}_N := \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$. Following the *receding horizon control* strategy, only the first control law $\mu_0 = \mu(x_k, \theta_k)$ in this sequence is applied to the real system, whereas the control laws $\{\mu_1, \dots, \mu_{N-1}\}$ are only used for predictions. The control objective is to stabilize the LPV system (1) to the origin. According to standard MPC, our cost function is defined as

$$J(\boldsymbol{\pi}_N, \mathbf{T}_k; x_k, \theta_k) = \|Px_{k+N}\|_p + \sum_{i=0}^{N-1} \|Qx_{k+i}\|_p + \|Ru_{k+i}\|_p, \quad (4)$$

where P, Q and R are real, full-column rank matrices of

appropriate dimensions, and where p denotes a polyhedral norm, usually the 1-norm or the ∞ -norm. Polyhedral norms enable a parametric solution to the stated problem using dynamic programming. Quadratic cost functions are not possible, since our procedure relies on epigraph reformulations, which would render the original problem a quadratically constrained parametric quadratic program, for which no efficient solution techniques are available.

In a closed-loop MPC approach, one assumes that each future control action u_{k+i} , $i = 1, \dots, N-1$, is not determined until x_{k+i} and θ_{k+i} are available, [17]. Since the future values of the scheduling parameters are still unknown, it has to be ensured that the constraints are satisfied for all possible \mathbf{T}_k . Moreover, the worst-case cost should be considered in order to assure that the actual cost function will be less or equal to the computed one, no matter how the scheduling parameters evolve. The optimization problem to solve in a closed-loop MPC approach is thus

$$\mu^*(x_k, \theta_k) = \arg \min_{\mu_0} \max_{\theta_{k+1}} \min_{\mu_1} \dots \max_{\theta_{k+N-1}} \min_{\mu_{N-1}} J(\boldsymbol{\pi}_N, \mathbf{T}_k; x_k, \theta_k) \quad (5a)$$

$$\text{s.t. } \forall i \in \{0, \dots, N-1\}$$

$$x_{k+i+1} = A(\theta_{k+i})x_{k+i} + B(\theta_{k+i})\mu_i(x_{k+i}, \theta_{k+i}), \quad (5b)$$

$$\mu_i(x_{k+i}, \theta_{k+i}) \in \mathbb{U} \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (5c)$$

$$x_{k+N} \in \mathbb{X}_T \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (5d)$$

$$x_{k+i} \in \mathbb{X} \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (5e)$$

$$\theta_{k+i} \in \Theta, \quad (5f)$$

with the polytopic terminal state constraints $\mathbb{X}_T \in \mathbb{R}^{n_x}$.

III. EXPLICIT MPC FOR LPV-A SYSTEMS

In this section we restrict ourselves to LPV-A systems, linear discrete-time systems with a parameter-varying state transition matrix. In the following, we will refer to (1) as LPV-A system, if the input matrix is constant, i.e., if $B_j = B \forall j \in \{1, \dots, n_\theta\}$. LPV-A systems thus possess the state-update equation,

$$x_{k+1} = \sum_{j=1}^{n_\theta} A_j \theta_{k,j} x_k + B u_k. \quad (6)$$

The reason for considering LPV-A systems separately is that they allow for a simpler computation of the state feedback control laws (3), and, as will be shown later, the considered closed-loop MPC problem can be solved to optimality.

Here we propose a *dynamic programming* (DP) procedure to solve (5), [18]. The DP procedure is started at the prediction horizon N with

$$J_N^*(x_{k+N}) := \|Px_{k+N}\|_p, \quad \mathbb{X}_N := \mathbb{X}_T. \quad (7)$$

Then we iterate backwards in time, with the iteration index i decreasing from $N-1$ to 1. At each iteration of the dynamic programming procedure, the parametric optimization problem

$$J_i^*(x_{k+i}) = \|Qx_{k+i}\|_p + \max_{\theta_{k+i}} \min_{\mu_i} \|R\mu_i(x_{k+i}, \theta_{k+i})\|_p + J_{i+1}^*(x_{k+i+1}) \quad (8a)$$

$$\text{s.t. } x_{k+i+1} = A(\theta_{k+i})x_{k+i} + B\mu_i(x_{k+i}, \theta_{k+i}), \quad (8b)$$

$$\mu_i(x_{k+i}, \theta_{k+i}) \in \mathbb{U} \quad \forall \theta_{k+i} \in \Theta, \quad (8c)$$

$$x_{k+i+1} \in \mathbb{X}_{i+1} \quad \forall \theta_{k+i} \in \Theta, \quad (8d)$$

$$x_{k+i} \in \mathbb{X}, \quad (8e)$$

$$\theta_{k+i} \in \Theta, \quad (8f)$$

is solved. The set \mathbb{X}_i , which denotes the polytopical set of states for which (8) is feasible at iteration i , describes the constraints (8d) on the successor state at the subsequent iteration $i-1$. The constraint (8e) on the parameter of the parametric optimization problem ensures the satisfaction of the state constraints in (2). For the parametric optimization problem at hand it is not a constraint on the optimization variable, but determines the set of states where a solution is wanted.

Note that we are not interested in states x_{k+i} , and associated control laws $\mu_i(x_{k+i}, \theta_{k+i})$, which are feasible for some θ_{k+i} (a nonconvex set), but in states x_{k+i} , which are feasible for *all* $\theta_{k+i} \in \Theta$ (a convex set). The feasible states are thus robustly feasible with respect to θ_{k+i} .

Theorem 3.1 (Solution properties of the DP iterations):

Consider optimization problem (8). The following statements hold:

- (i) The set of feasible states \mathbb{X}_i is a polytope in \mathbb{R}^{n_x} , and \mathbb{X}_i is partitioned into polytopical critical regions.
- (ii) The optimal value function $J_i^*(x_{k+i})$ is continuous, convex and piecewise affine over \mathbb{X}_i , and affine in each critical region.
- (iii) The optimal solution $\mu_i^*(x_{k+i}, \theta_{k+i})$ is a continuous piecewise affine function of the state x_{k+i} and an affine function¹ of the scheduling parameter θ_{k+i} , i.e. of the form

$$\mu_i^*(x_{k+i}, \theta_{k+i}) = \sum_{j=1}^{n_\theta} \theta_{k+i,j} \mu_{i,j}^*(x_{k+i}) \quad (9)$$

with $\mu_{i,j}^*(x_{k+i})$ being continuous and polyhedral piecewise affine over \mathbb{X}_i .

Proof: For the time being assume that the set \mathbb{X}_{i+1} is a polyhedron and the optimal cost function of the previous DP iteration J_{i+1}^* is a continuous, convex and polyhedral piecewise affine function. Introducing the epigraph variable t_i , an epigraph reformulation is applied to obtain

$$J_i^*(x_{k+i}) = \|Qx_{k+i}\|_p + \max_{\theta_{k+i}} \min_{\{\mu_i, t_i\}} t_i(x_{k+i}, \theta_{k+i}) \quad (10a)$$

$$\text{s.t. } x_{k+i+1} = A(\theta_{k+i})x_{k+i} + B\mu_i(x_{k+i}, \theta_{k+i}), \quad (10b)$$

$$\|R\mu_i(x_{k+i}, \theta_{k+i})\|_p + J_{i+1}^*(x_{k+i+1}) \leq t_i(x_{k+i}, \theta_{k+i}) \quad \forall \theta_{k+i} \in \Theta, \quad (10c)$$

$$\mu_i(x_{k+i}, \theta_{k+i}) \in \mathbb{U} \quad \forall \theta_{k+i} \in \Theta, \quad (10d)$$

$$x_{k+i+1} \in \mathbb{X}_{i+1} \quad \forall \theta_{k+i} \in \Theta, \quad (10e)$$

$$x_{k+i} \in \mathbb{X}, \quad (10f)$$

$$\theta_{k+i} \in \Theta. \quad (10g)$$

The constraints (10c) - (10e) describe a polyhedron in the space spanned by μ_i, t_i and x_{k+i+1} . Hence by inserting the state-update equation (6), the optimization problem can be written as

$$J_i^*(x_{k+i}) = \|Qx_{k+i}\|_p + \max_{\theta_{k+i}} \min_{\{\mu_i, t_i\}} t_i(x_{k+i}, \theta_{k+i}) \quad (11a)$$

$$\text{s.t. } C_{\mu t} \begin{bmatrix} \mu_i(x_{k+i}, \theta_{k+i}) \\ t_i(x_{k+i}, \theta_{k+i}) \end{bmatrix} \quad (11b)$$

$$\leq c - C_x \left(\sum_{j=1}^{n_\theta} A_j \theta_{k+i,j} x_{k+i} \right) \quad \forall \theta_{k+i} \in \Theta,$$

$$x_{k+i} \in \mathbb{X}, \quad (11c)$$

$$\theta_{k+i} \in \Theta, \quad (11d)$$

with $C_{\mu t}, C_x$ and c being matrices and a vector of appropriate dimensions describing the hyperplanes of this polyhedron. When solving the optimization problem (11) parametrically, we are only interested in regions which are full-dimensional with respect to the state x_{k+i} , i.e. where the constraints (11c) are inactive. Thus the set of active constraints (in the following denoted by the subscript \mathcal{A}) in each region will be a subset of the constraints (11b) and one can conclude on the structure of the optimization variables

$$\begin{aligned} & \begin{bmatrix} \mu_i(x_{k+i}, \theta_{k+i}) \\ t_i(x_{k+i}, \theta_{k+i}) \end{bmatrix} \\ &= (C_{\mu t, \mathcal{A}})^{-1} \left(c_{\mathcal{A}} - C_{x, \mathcal{A}} \left(\sum_{j=1}^{n_\theta} A_j \theta_{k+i,j} x_{k+i} \right) \right) \\ &= \sum_{j=1}^{n_\theta} \theta_{k+i,j} \left((C_{\mu t, \mathcal{A}})^{-1} (c_{\mathcal{A}} - C_{x, \mathcal{A}} A_j x_{k+i}) \right) \\ &=: \sum_{j=1}^{n_\theta} \theta_{k+i,j} \begin{bmatrix} \mu_{i,j}(x_{k+i}) \\ t_{i,j}(x_{k+i}) \end{bmatrix}. \end{aligned} \quad (12)$$

Consequently, the optimization variables $\mu_i(x_{k+i}, \theta_{k+i})$ and $t_i(x_{k+i}, \theta_{k+i})$ depend affinely on θ_{k+i} . We continue with the remaining properties of Theorem 3.1. Inserting the input parametrization (12) in (11) (and recalling that $\mu_{i,j}$ and $t_{i,j}$ are parameter-independent) yields

$$J_i^*(x_{k+i}) = \|Qx_{k+i}\|_p + \min_{\{\mu_i, t_i, \theta_{k+i}\}} \max_{\theta_{k+i}} \sum_{j=1}^{n_\theta} \theta_{k+i,j} t_{i,j}(x_{k+i}) \quad (13a)$$

¹Since $\theta_{k+i} \in \Theta$, any affine function can be written in the linear form (9) by multiplying the constant term with $\sum_{j=1}^{n_\theta} \theta_{k+i,j}$.

$$\begin{aligned}
\text{s.t. } & \sum_{j=1}^{n_\theta} \theta_{k+i,j} C_{\mu t} \begin{bmatrix} \mu_{i,j}(x_{k+i}) \\ t_{i,j}(x_{k+i}) \end{bmatrix} \\
& \leq \sum_{j=1}^{n_\theta} \theta_{k+i,j} (c - C_x A_j x_{k+i}) \quad \forall \theta_{k+i} \in \Theta, \quad (13b) \\
& x_{k+i} \in \mathbb{X}, \quad (13c) \\
& \theta_{k+i} \in \Theta. \quad (13d)
\end{aligned}$$

The constraints (13b) are satisfied if and only if they are satisfied at the vertices of the parameter simplex. Moreover, the maximum is attained at a vertex of the parameter simplex, such that the optimization problem (13) can be stated as

$$\begin{aligned}
J_i^*(x_{k+i}) &= \|Qx_{k+i}\|_p + \min_{\{\mu_{i,j}, t_{i,j}\}} \max_j t_{i,j}(x_{k+i}) \quad (14a) \\
\text{s.t. } & C_{\mu t} \begin{bmatrix} \mu_{i,j}(x_{k+i}) \\ t_{i,j}(x_{k+i}) \end{bmatrix} \leq c - C_x A_j x_{k+i} \quad \forall j \in \{1, \dots, n_\theta\}, \quad (14b) \\
& x_{k+i} \in \mathbb{X}, \quad (14c) \\
& j \in \{1, \dots, n_\theta\}. \quad (14d)
\end{aligned}$$

The parametric optimization problem (14) is a parametric linear program, which implies properties (i) and (ii) of Theorem 3.1. It also implies the remaining property of (iii), namely that the optimal solution $\mu_{i,j}^*(x_{k+i})$ is continuous and polyhedral piecewise affine in x_{k+i} . Moreover, starting with (7), it follows by induction that \mathbb{X}_{i+1} is polytopic and J_{i+1}^* convex and polyhedral piecewise affine as assumed initially. ■

The *final step of the dynamic programming procedure* differs from the previous steps. As the scheduling parameter θ_k is measured and known, this information can, and should, be taken into account instead of considering the worst case.

$$\begin{aligned}
\mu^*(x_k, \theta_k) &= \\
& \arg \min_{\mu_0} \|Qx_k\|_p + \|R\mu_0(x_k, \theta_k)\|_p + J_1^*(x_{k+1}) \quad (15a) \\
\text{s.t. } & x_{k+1} = A(\theta_k)x_k + B\mu_0(x_k, \theta_k), \quad (15b) \\
& \mu_0(x_k, \theta_k) \in \mathbb{U}, \quad (15c) \\
& x_{k+1} \in \mathbb{X}_1, \quad (15d) \\
& x_k \in \mathbb{X}, \quad (15e) \\
& \theta_k \in \Theta, \quad (15f)
\end{aligned}$$

Unfortunately, bilinear constraints appear in the optimization problem, which prevents a standard parametric solution strategy. One way around this is to solve the optimization problem not parametrically in the measured state x_k , but in the *uncontrolled successor state*,

$$z_k := \left(\sum_{j=1}^{n_\theta} A_j \theta_{k,j} \right) x_k, \quad (16)$$

which was introduced in [12]. By substituting the uncontrolled successor state z_k , by omitting the current state cost $\|Qx_k\|_p$ (which as an additive offset has no influence on $\mu^*(x_k, \theta_k)$) and by omitting the constraints on the current state and current scheduling parameter (yielding a parametric solution for a larger set of initial states and scheduling parameters, but not changing the solution within $\mathbb{X} \times \Theta$) the optimization problem

in the final DP step becomes

$$J^*(z_k) = \min_{\{\mu_0, t\}} t(z_k) \quad (17a)$$

$$\text{s.t. } C_{\mu t}[\mu_0(z_k), t(z_k)]^T \leq c - C_x z_k, \quad (17b)$$

which is a standard parametric linear program with respect to the uncontrolled successor state. The explicit solution is a piecewise affine control law, $u_k = \mu^*(z_k)$, which is stored in a look-up table, [9]. Online, all we have to do is to compute the uncontrolled successor state z_k , which is completely determined by the measured state x_k and scheduling parameter θ_k , and evaluate the look-up table to obtain the optimal control input u_k .

Theorem 3.2 (Optimality of the DP solution): The iterative solution of (8) for i decreasing from $N - 1$ to 1, initiated with (7) and followed by solving (15) is equivalent to solving (5) directly.

Proof: The theorem and its proof is an adaptation of the well-known *principle of optimality*, see e.g. [18]. We show equivalence here by back-substitution, starting by noting that (15) is equivalent to

$$\begin{aligned}
\mu^*(x_k, \theta_k) &= \\
& \arg \min_{\mu_0} \|Qx_k\|_p + \|R\mu_0(x_k, \theta_k)\|_p + J_1^*(x_{k+1}) \quad (18a) \\
\text{s.t. } & x_{k+1} = A(\theta_k)x_k + B\mu_0(x_k, \theta_k), \quad (18b) \\
& \mu_0(x_k, \theta_k) \in \mathbb{U} \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (18c) \\
& x_{k+1} \in \mathbb{X}_1 \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (18d) \\
& x_k \in \mathbb{X} \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (18e) \\
& \theta_k \in \Theta, \quad (18f)
\end{aligned}$$

because (15c) – (15e) are independent of \mathbf{T}_k . By substituting $J_1^*(x_{k+1})$ using (8) with $i = 1$ and by shifting the optimization operators we obtain

$$\begin{aligned}
\mu^*(x_k, \theta_k) &= \arg \min_{\mu_0} \max_{\theta_{k+1}} \min_{\mu_1} \sum_{i=0}^1 \|Qx_{k+i}\|_p \\
& + \|R\mu_i(x_{k+i}, \theta_{k+i})\|_p + J_2^*(x_{k+2}) \quad (19a)
\end{aligned}$$

$$\begin{aligned}
\text{s.t. } & \forall i \in \{0, 1\} \\
& x_{k+i+1} = A(\theta_{k+i})x_{k+i} + B(\theta_{k+i})\mu_i(x_{k+i}, \theta_{k+i}), \quad (19b) \\
& \mu_i(x_{k+i}, \theta_{k+i}) \in \mathbb{U} \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (19c) \\
& x_{k+2} \in \mathbb{X}_2 \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (19d) \\
& x_{k+i} \in \mathbb{X} \quad \forall \mathbf{T}_k \in \Theta^{N-1}, \quad (19e) \\
& \theta_{k+i} \in \Theta, \quad (19f)
\end{aligned}$$

Again we redundantly require constraint satisfaction for all future scheduling parameters \mathbf{T}_k , although for $i = 1$ the constraints (8c) – (8e) do only depend on θ_k and θ_{k+1} . Note that $x_{k+1} \in \mathbb{X}_1$ is omitted, because it is now redundant (\mathbb{X}_1 is implicitly defined by the other constraints coming from (8)). By iteratively substituting $J_i^*(x_{k+i})$ using (8) with $i = 1, \dots, N - 1$ in an analogue fashion, and by finally inserting (7) and (4), the equivalence to (5) is shown.

The DP procedure finishes with the optimal cost $J^*(z_k)$, defined over a set \mathbb{Z}_f of feasible uncontrolled successor states z_k . While a parameter-dependent state-feedback control law (3) is obtained already, one might additionally be interested in the set \mathbb{X}_f of states x_k , which are feasible for all possible current scheduling parameter values $\theta_k \in \Theta$, and, for a given state x_k , the worst-case costs $\bar{J}(x_k)$ with respect to the current scheduling parameter. From convexity reasons it is possible to compute both by solving the parametric linear program

$$\bar{J}(x_k) = \|Qx_k\|_p + \min_t t(x_k) \quad (20a)$$

$$\text{s.t. } J^*(A_j x_k) \leq t(x_k), \quad j = 1, \dots, n_\theta, \quad (20b)$$

$$A_j x_k \in \mathbb{Z}_f, \quad j = 1, \dots, n_\theta, \quad (20c)$$

$$x_k \in \mathbb{X}. \quad (20d)$$

The set \mathbb{X}_f of states x_k , for which (20) is feasible, is polyhedral.

IV. EXPLICIT MPC FOR LPV SYSTEMS

This section is concerned with the computation of explicit state feedback control laws for the more general class of discrete-time LPV systems (1). Contrary to the previous section, the input matrix is now assumed to depend affinely on the scheduling parameter, i.e. $\exists i, j \in \{1, \dots, n_\theta\} : B_i \neq B_j$.

Following the dynamic programming procedure from Section III, one would reach an optimization problem similar to (11), but with the scheduling parameter appearing in the constraint matrix, $C_{\mu t} = C_{\mu t}(\theta_{k+i})$. Parametric optimization problems with a parameter-dependent constraint matrix are known to be difficult problems, and are far less well understood than parametric linear programs, [6]. Some interesting aspects were presented in [19] for the case of a single parameter. The optimal solution within one critical region was computed by solving an extended linear program. This solution however turns out to be a rational function of the parameter with a degree equal to the number of active constraints. Each coefficient of this rational function is an optimization variable in the linear program, which would render the optimal control laws already for tiny LPV systems very complex. Moreover, the critical regions are generally no polyhedra, and the optimal cost function is not a convex piecewise affine function. These obstacles cumber the computation, the storage and the efficient evaluation of the optimal solution and render the optimal solution to (5) impractical, if not impossible.

Consequently we are proposing a *suboptimal* solution by restricting the parametrization of the input. In the following we will consider the control law to be an affine function of the scheduling parameter,

$$u_{k+i} = \mu_i(x_{k+i}, \theta_{k+i}) = \sum_{j=1}^{n_\theta} \theta_{k+i,j} \mu_{i,j}(x_{k+i}), \quad (21)$$

where $\mu_{i,j}(x_{k+i})$ corresponds to the control law in the j th vertex of the parameter simplex (1c).

Remark 2: Note that in principle also input parametrizations, which are polynomial or rational in the scheduling parameter θ_k , are possible with our proposed method. We restrict

ourselves to affine input parametrizations for complexity and notational reasons.

Analogue to Section III we solve (5) by a *dynamic programming* (DP) procedure iterating backwards in time. The DP procedure is initialized at the prediction horizon N with (7). Then at each iteration an epigraph reformulation is applied to the DP optimization problem in order to transfer the parameter dependence to the constraints. This leads to semi-infinite optimization problems of the form

$$J_i^*(x_{k+i}) = \|Qx_{k+i}\|_p + \min_{\{\mu_{i,j}, t_i\}} t_i(x_{k+i}) \quad (22a)$$

$$\text{s.t. } C_{\mu t}(\theta_{k+i}) \begin{bmatrix} \mu_i(x_{k+i}, \theta_{k+i}) \\ t_i(x_{k+i}) \end{bmatrix} \leq (c - C_x A(\theta_{k+i})x_{k+i}) \quad \forall \theta_{k+i} \in \Theta, \quad (22b)$$

$$x_{k+i} \in \mathbb{X}, \quad (22c)$$

$$\theta_{k+i} \in \Theta. \quad (22d)$$

The constraints (22b) are quadratic in the scheduling parameter and a vertex enumeration is *not* sufficient to ensure constraint satisfaction over the whole simplex. However, the constraint satisfaction of the semi-infinite optimization problem (22) can be ensured, conservatively, over the whole parameter simplex due to Pólya's theorem:

Theorem 4.1 (Pólya's theorem): If a homogeneous polynomial $p(\theta)$ is positive on the standard simplex Θ , the coefficients C_{N_p} of the extended polynomial $p_{N_p}(\theta) = p(\theta) \cdot (\sum_{j=1}^{n_\theta} \theta_j)^{N_p}$ are positive for a sufficiently large *Pólya degree* N_p .

Proof: See [20], [21].

We will make use of the more obvious reverse of Pólya's theorem, i.e. positive coefficients of the extended polynomial mean positivity over the whole simplex. We will call this reverse *Pólya's relaxation*², since it is a relaxation in the max direction.

The following design procedure describes the relaxation of the parameter-dependent constraints (22b) into constraints which are piecewise affine in the state and inputs and independent of the scheduling parameter:

- 1) Reformulate (22b) into a positivity constraint of a polynomial p , $p(\theta) \geq 0 \forall \theta \in \Theta$.
- 2) Homogenize the polynomial $p(\theta)$ by multiplying single monomials with $\sum_{j=1}^{n_\theta} \theta_j$ (which equals one on the standard simplex) 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 $p_{N_p}(\theta) = p(\theta) \cdot (\sum_{j=1}^{n_\theta} \theta_j)^{N_p}$.
- 4) Replace (22b) by $C_{N_p} \geq 0$.

If all coefficients C_{N_p} are non-negative, so is the polynomial $p(\theta)$. Hence the semi-infinite optimization problem (22) can be transformed into the following parametric linear program:

$$J_i^*(x_{k+i}) = \|Qx_{k+i}\|_p + \min_{\{\mu_{i,j}, t_i\}} t_i(x_{k+i}) \quad (23a)$$

²Pólya's relaxation is implemented in YALMIP as one of the so-called filters in the robust optimization framework, [22].

$$\text{s.t. } \mathbf{C}_{N_p}(\mu_{i,1}, \dots, \mu_{i,n_\theta}, t_i; x_{k+i}) \geq 0, \quad (23b)$$

$$x_{k+i} \in \mathbb{X}. \quad (23c)$$

Note that the coefficients \mathbf{C}_{N_p} lie in a cone which is spanned by the coefficients of (22b), such that the polyhedral dependence on $\mu_{i,1}, \dots, \mu_{i,n_\theta}, t_i, x_{k+i}$ is preserved.

Remark 3: The choice of the Pólya degree N_p for the Pólya relaxation is a tradeoff between solution complexity and introduced conservatism. The larger the Pólya degree, the less conservative the relaxation, but also the greater the number of constraints in (23b).

Remark 4: Instead of using an affine input parametrization, it is also possible to select a *parameter-independent control law* in all but the final dynamic programming iterations, following a robust approach. This approach typically results in less complex control laws, since instead of Pólya's relaxation a vertex enumeration as in the case of LPV-A systems is sufficient, and since (21) simplifies to a single control law. On the other hand, this approach will result in a more conservative control law, since the predicted future inputs are assumed not to be able to react on the scheduling parameter at the same prediction time.

The *final step of the dynamic programming procedure* differs from the preceding steps, since knowledge of the current scheduling parameter values can be exploited to improve control performance. The problem we want to solve is

$$\mu^*(x_k, \theta_k) = \arg \min_{\mu_0} \|R\mu_0(x_k, \theta_k)\|_p + J_1^*(x_{k+1}) \quad (24a)$$

$$\text{s.t. } x_{k+1} = A(\theta_k)x_k + B(\theta_k)\mu_0(x_k, \theta_k), \quad (24b)$$

$$\mu_0(x_k, \theta_k) \in \mathbb{U} \quad \forall \theta_k \in \Theta, \quad (24c)$$

$$x_{k+1} \in \mathbb{X}_1 \quad \forall \theta_k \in \Theta, \quad (24d)$$

$$x_k \in \mathbb{X}, \quad (24e)$$

$$\theta_k \in \Theta, \quad (24f)$$

which is again a parametric optimization problem with a parametric constraint matrix. Apart from solving (24) on-line, when the current x_k, θ_k are known, there exist several possibilities for suboptimal explicit solutions: (i) Grid-and-interpolate regarding the parameter simplex, (ii) minimize the average cost w.r.t. some grid points of the parameter simplex, and (iii) minimize the worst-case cost w.r.t. the scheduling parameter. The latter two possibilities require the selection of an appropriate input parametrization such as (21), and the application of Pólya's relaxation. They can be supplemented by solving (24) parametrically in the uncontrolled successor state (16) as described in Section III.

In our test simulations there was no distinctively superior approach. A worst-case minimization approach using the uncontrolled successor state turned out to be a good tradeoff between solution complexity and control performance. It also allows one to determine the feasible set \mathbb{X}_f and the worst-case cost for a given state, $\bar{J}(x_k)$, easily, by the solution of (20). Note however that it is merely an heuristic, and for specific systems other approaches might be more successful.

The procedure proposed above involves two approximation steps, namely the fixing of the input parametrization and

Pólya's relaxation. Hence it might be interesting to know a bound on the introduced suboptimality. Such a bound can be determined with respect to the worst-case cost $\bar{J}(x_k)$. A lower bound on the worst case cost, $\underline{J}(x_k)$, and thus a bound on the suboptimality, can be computed by a DP procedure based on gridding, similar to [23]. At first the parameter simplex is gridded to obtain a number of grid points $\hat{\theta}_1, \dots, \hat{\theta}_{n_g}$. Then, at each DP step, the following procedure is performed:

- 1) For each grid point $\hat{\theta}_g$, solve the optimization problem (22) with $\theta_{k+i} = \hat{\theta}_g$, to obtain $\underline{J}_{i,g}(x_{k+i})$.
- 2) Determine the optimal cost-to-go as $\underline{J}_i(x_{k+i}) = \max_g \{\underline{J}_{i,g}(x_{k+i}) \mid g = 1, \dots, n_g\}$.

Since the proposed procedure does not consider the entire parameter simplex Θ , but only the grid points $\hat{\theta}_1, \dots, \hat{\theta}_{n_g}$, the resulting cost function $\underline{J}(x_k)$ is a lower bound on the worst-case cost $\bar{J}(x_k)$. A bound on the suboptimality of the worst-case costs can then be determined as $\max_{x_k} \{\bar{J}(x_k) - \underline{J}(x_k) \mid x_k \in \mathbb{X}_f\}$.

Remark 5: In the case when the rate of parameter variation of an LPV system is known to be limited, the procedure described above is conservative. A possibility to mitigate this conservatism is to subdivide the parameter simplex into several sub-simplices, and to solve a parametric optimization problem for each sub-simplex separately. Further material on such an approach can be found in [14], [24].

V. STABILITY

This section is concerned with stability of the resulting closed-loop system, when explicit control laws are applied to LPV systems. One can guarantee (i) asymptotic stability, (ii) constraint satisfaction, and (iii) recursive feasibility *a-priori* for all feasible states, by considering a *dual mode* approach and by choosing the terminal state constraints \mathbb{X}_T and the polyhedral terminal cost $L_N(x_{k+N})$ appropriately, [25]. From [25] we have the following conditions for asymptotic stability:

A1: $\mathbb{X}_T \subseteq \mathbb{X}$, \mathbb{X}_T closed and contains the origin.

A2: $\mu_T(x_k, \theta_k) \in \mathbb{U} \quad \forall x_k \in \mathbb{X}_T \quad \forall \theta_k \in \Theta$.

A3: $x_{k+1} = A(\theta_k)x_k + B(\theta_k)\mu_T(x_k, \theta_k) \in \mathbb{X}_T \quad \forall x_k \in \mathbb{X}_T \quad \forall \theta_k \in \Theta$

A4: $L_N(x_k) - L_N(x_{k+1}) \leq \|Qx_k\|_p + \|R\mu_T(x_k, \theta_k)\|_p \quad \forall x_k \in \mathbb{X}_T \quad \forall \theta_k \in \Theta$.

Furthermore, it is a well-known fact, that stability is preserved in the case of a suboptimal solution, as long as the suboptimality of the cost function does not exceed one stage cost, [26]. Consider the following procedure, which is based on [14] and [9]:

- 1) Compute an asymptotically stabilizing terminal region parameter-varying state-feedback controller

$$u_k = \mu_T(x_k, \theta_k) = K(\theta_k)x_k = \sum_{j=1}^{n_\theta} K_j \theta_{k,j} x_k \quad (25)$$

for the unconstrained system (1), e.g. by the procedure in [14].

- 2) Determine a polytopic λ -contractive terminal region \mathbb{X}_T by pre-image computations, such that

$$\begin{aligned} \forall x_k \in \mathbb{X}_T, \forall \theta_k \in \Theta \exists \mu_T(x_k, \theta_k) \in \mathbb{U} : \\ A(\theta_k)x_k + B(\theta_k)\mu_T(x_k, \theta_k) \in \lambda\mathbb{X}_T \end{aligned} \quad (26)$$

holds for some $\lambda \in [0, 1)$ (see e.g. [14]).

- 3) Scale the Minkowski function

$$\psi_{\mathbb{X}_T}(x_k) := \min_{\alpha} \{\alpha \in \mathbb{R}_+ \mid x_k \in \alpha\mathbb{X}_T\}, \quad (27)$$

induced by the terminal region \mathbb{X}_T , by a factor $\beta^* \in \mathbb{R}_+$, which can be determined by the linear program

$$\beta^* = \min_{\beta} \beta \quad (28a)$$

$$\begin{aligned} \text{s.t. } \beta(1 - \lambda) &\geq \|Qv_i\|_p + \|RK_j v_i\|_p \\ &\forall v_i \in \text{vert}(\mathbb{X}_T) \forall j \in \{1, \dots, n_{\theta}\}. \end{aligned} \quad (28b)$$

- 4) Define $L_N(x_{k+N}) := \beta^* \psi_{\mathbb{X}_T}(x_{k+N})$.

Theorem 5.1: Assume that there exists a terminal region control $\mu_T(x_k, \theta_k)$ of the form (25), which renders the polytope \mathbb{X}_T λ -contractive as in (26) with $\lambda \in [0, 1)$. Then, for the terminal region \mathbb{X}_T and the terminal cost $L_N(x_k) = \beta^* \psi_{\mathbb{X}_T}(x_k)$, as defined in (27)-(28), the conditions A1 to A4 are satisfied, such that we have asymptotic stability, constraint satisfaction and recursive feasibility for all feasible points.

Proof: Conditions A1 to A3 follow immediately from the properties of the λ -contractive terminal set \mathbb{X}_T . Condition A4 follows from

$$\begin{aligned} \beta\psi_{\mathbb{X}_T}(x_k) - \beta\psi_{\mathbb{X}_T}(x_{k+1}) &\geq \|Qx_k\|_p + \|RK(\theta_k)x_k\|_p \\ &\forall x_k \in \mathbb{X}_T \forall \theta_k \in \Theta \\ \Leftrightarrow \beta(1 - \lambda)\psi_{\mathbb{X}_T}(x_k) &\geq \|Qx_k\|_p + \|RK(\theta_k)x_k\|_p \\ &\forall x_k \in \mathbb{X}_T \forall \theta_k \in \Theta \\ \Leftrightarrow \beta(1 - \lambda)\psi_{\mathbb{X}_T}(x_k) &\geq \|Qx_k\|_p + \sum_{j=1}^{n_{\theta}} \theta_{k,j} \|RK_j x_k\|_p \\ &\forall x_k \in \mathbb{X}_T \forall \theta_k \in \Theta \\ \Leftrightarrow \sum_{j=1}^{n_{\theta}} \theta_{k,j} \beta(1 - \lambda)\psi_{\mathbb{X}_T}(x_k) &\geq \sum_{j=1}^{n_{\theta}} \theta_{k,j} \|Qx_k\|_p \\ &\quad + \sum_{j=1}^{n_{\theta}} \theta_{k,j} \|RK_j x_k\|_p \quad \forall x_k \in \mathbb{X}_T \forall \theta_k \in \Theta \\ \Leftrightarrow \beta(1 - \lambda)\psi_{\mathbb{X}_T}(x_k) &\geq \|Qx_k\|_p + \|RK_j x_k\|_p \\ &\forall x_k \in \mathbb{X}_T \forall j \in \{1, \dots, n_{\theta}\} \\ \Leftrightarrow \beta(1 - \lambda) &\geq \|Qv_i\|_p + \|RK_j v_i\|_p \\ &\forall v_i \in \text{vert}(\mathbb{X}_T) \forall j \in \{1, \dots, n_{\theta}\} \end{aligned}$$

■

Note that a symmetric terminal set \mathbb{X}_T implies a Minkowski function which can be expressed as $L_N(x_{k+N}) = \|Px_{k+N}\|_{\infty}$ with some matrix P . However, the described procedures work also for convex polyhedral piecewise affine terminal cost $L_N(x_{k+N})$. For more results on stability of LPV-A systems, see e.g. [27].

The drawback of adding terminal state constraints is that in general they lead to a loss of performance, a smaller

feasible space and an increase in complexity of the resulting control law. While the former two effects can be mitigated by extending the prediction horizon, this typically leads to a further increase in the complexity of the control law. In order to avoid these downsides in practical implementations, a possibility is to omit the terminal state constraints and to verify stability a-posteriori, following the theory described in [27].

The closed-loop system in the space of the uncontrolled successor state is given by:

$$\begin{aligned} z_{k+1} = \\ A(\theta_{k+1})x_{k+1} = A(\theta_{k+1})\{z_k + B(\theta_k)\mu(z_k, \theta_k)\} = \\ A(\theta_{k+1})(I + B(\theta_k)F_r(\theta_k))z_k + A(\theta_{k+1})B(\theta_k)g_r(\theta_k), \\ \text{if } z_k \in \mathbb{D}_r, \quad r = 1, \dots, n_r. \end{aligned} \quad (29)$$

In the following, we will describe an approach to verify stability of the closed-loop system (29). The a-posteriori stability verification can also be applied, in simplified form, to controllers for LPV-A systems, and to the case of bounded rate of scheduling parameter variation.

A polytope \mathbb{T} containing the origin in its interior can be written in standard form $\mathbb{T} = \{z \in \mathbb{R}^n \mid E_t z \leq \mathbf{1}\}$, and induces the *gauge function* (also known as the Minkowski functional of \mathbb{T})

$$\Psi_{\mathbb{T}}(z) := \max E_t z. \quad (30)$$

Definition 1: A set \mathbb{T} is said to be λ -contractive w.r.t. the closed-loop system (29), if $z_{k+1} \in \lambda\mathbb{T} \forall z_k \in \mathbb{T}, \forall \theta_k, \theta_{k+1} \in \Theta$.

Definition 2: Let \mathbf{I} be the index set of all controller regions containing the origin,

$$\mathbf{I} := \{r \in \{1, \dots, n_r\} \mid 0 \in \mathbb{D}_r\}.$$

The index set \mathbf{I} is single-valued if the origin is contained in the interior of a controller region, and multi-valued if the origin lies on the facet of several controller regions.

The stability analysis can be performed in three steps; the first verifies the origin to be an equilibrium point, the second considers the stability of the target region \mathbb{T} , and in the third a reachability analysis is performed.

- 1) *Invariance of the origin*

At first we require the origin to be invariant, i.e. to be an equilibrium of the closed-loop system. This is the case, if the condition

$$g_r(\theta_k) = 0 \quad \forall \theta_k \in \Theta, \quad r \in \mathbf{I} \quad (31)$$

holds. Since \mathbb{X} and \mathbb{U} include the origin, there always exists a controller which fulfils this condition. If Condition (31) is violated, the origin is *not* an equilibrium point, and at most ultimate boundedness to a target set can be present.

- 2) *Contractiveness of the target region*

We infer asymptotic stability of the origin from the existence of a λ -contractive target region \mathbb{T} . The *contractive preset* of a set $\mathbb{Z}_i \subset \mathbb{R}^{n_x}$ w.r.t. to the closed-loop system

(29) is given by

$$\Omega(\mathbb{Z}_i, \mathbb{D}_r) := \left\{ z_k \in \mathbb{D}_r \mid \begin{array}{l} A_j(I + B(\theta_k)F_r(\theta_k))z_k + A_jB(\theta_k)g_r(\theta_k) \in \lambda\mathbb{Z}_i, \\ j = 1, \dots, n_\theta, \quad \forall \theta_k \in \Theta \end{array} \right\}, \quad (32)$$

where $\lambda \in [0, 1)$ denotes the *contraction ratio*. For an efficient treatment of the sets \mathbb{Z}_i we require them to be polytopes. Since the contractive presets $\Omega(\mathbb{Z}_i, \mathbb{D}_r)$ are convex but not polytopes, we determine the polytopic preset $\Omega_{N_p}(\mathbb{Z}_i, \mathbb{D}_r)$ of \mathbb{Z}_i by applying Pólya's relaxation to (32). If the index set \mathbf{I} of regions containing the origin is single-valued, a robust invariant target set can be determined by repeatedly computing polytopic presets, $\Omega_{N_p}(\dots(\Omega_{N_p}(\mathbb{Z}_0, \mathbb{D}_r), \mathbb{D}_r) \dots, \mathbb{D}_r)$, starting with $\mathbb{Z}_0 = \mathbb{D}_r$. If the index set \mathbf{I} is multi-valued, the computation runs in parallel for each regarded region $r \in \mathbf{I}$, starting with a polytope $\mathbb{Z}_0 \subseteq \bigcup_{r \in \mathbf{I}} \mathbb{D}_r$ containing the origin in the interior and determining the largest polytope contained in the union of the polytopic presets after each iteration $\mathbb{Z}_{i+1} \subseteq \bigcup_{r \in \mathbf{I}} \Omega(\mathbb{Z}_i, \mathbb{D}_r)$.

If $\mathbb{Z}_i \subseteq \mathbb{Z}_{i+1}$, the algorithm is terminated with $\mathbb{T} := \mathbb{Z}_i$, and the resulting target region \mathbb{T} is λ -contractive with regard to the closed-loop system (29). It follows from the succeeding proposition that the existence of a λ -contractive polytope \mathbb{T} induces asymptotic stability of the origin.

Proposition 1: Let $\mathbb{T} \subseteq \bigcup_{r \in \mathbf{I}} \mathbb{D}_r \subset \mathbb{R}^{n_x}$ be a polytope containing the origin in its interior and let Condition (31) hold. If $\forall r \in \mathbf{I}$, all vertices v_r^i of $\mathbb{T} \cap \mathbb{D}_r$ are mapped into $\lambda\mathbb{T}$, $0 \leq \lambda < 1$, then $\mu\mathbb{T}$ is λ -contractive $\forall \mu \in [0, 1]$.

Proof: Consider any $\hat{z}_k \in \mu(\mathbb{T} \cap \mathbb{D}_r) \Rightarrow \tilde{z}_k = \hat{z}_k/\mu \in (\mathbb{T} \cap \mathbb{D}_r) \Leftrightarrow \exists \alpha_r^i \in \mathbb{R}_+, \sum_i \alpha_r^i = 1 : \tilde{z}_k = \sum_i \alpha_i v_r^i \Rightarrow \hat{z}_k = \mu \sum_i \alpha_i v_r^i \Rightarrow \hat{z}_{k+1} \in \mu \sum_i \alpha_i \lambda \mathbb{T} = \mu \lambda \mathbb{T}$. ■

Proposition 1, together with the properties of the gauge function $\psi_{\mathbb{T}}(z)$ induced by \mathbb{T} suffices to establish asymptotic stability inside \mathbb{T} by using $\psi_{\mathbb{T}}(z)$ as a Lyapunov function.

3) Reachability analysis

A reachability analysis can be performed to check which states are mapped into the target set \mathbb{T} under the computed control law. This reachability analysis is also based on repeated preset computations, using the polytopic preset arising from applying Pólya's relaxation to (32). We start with $\mathbb{Z}_0 = \mathbb{T}$. In each iteration i we compute $\Omega_{N_p}(\mathbb{Z}_i, \mathbb{D}_r)$, $r = 1, \dots, n_r$ and merge the resulting presets to $\mathbb{Z}_{i+1} = \bigcup_{r=1}^{n_r} \Omega_{N_p}(\mathbb{Z}_i, \mathbb{D}_r)$.

The iterations terminate when $\mathbb{Z}_{i+1} \subseteq \mathbb{Z}_i$ or when the entire feasible space is covered. The resulting region of attraction is denoted by \mathbb{Z}_∞ . All uncontrolled successor states $z_k \in \mathbb{Z}_\infty$ are controlled to the target set \mathbb{T} and eventually to the origin by construction. The required computations boil down to polytopic manipulations and can be adapted from the algorithms given in the refer-

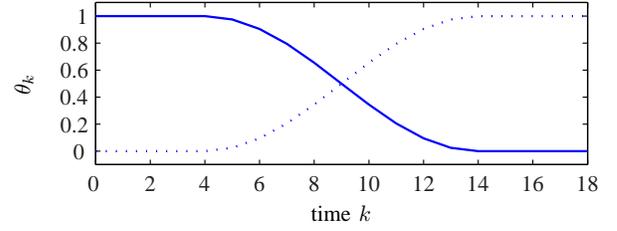


Fig. 1. Evolution of the scheduling parameter, $\theta_{k,1}$ (—), $\theta_{k,2}$ (···), in Example 1.

ences mentioned above.

VI. NUMERICAL EXAMPLES

A. Example 1: Explicit MPC of an LPV-A system

In the first example the potential of Explicit LPV-A MPC in reducing the online computational effort is demonstrated. The example system is taken from [28]. It represents an unstable LPV-A system of the form (6) with the system matrices

$$A_1 = \begin{bmatrix} 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 0.1 & 0 & 0 & 0 \\ 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 0.1 & 0 & 0 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 1 & 0 & 1 & 0 \\ 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

$$B = [1 \ 0 \ 0 \ 0].$$

The states are constrained to be $x_{k,i} \leq 1.14$, $i \in [1, 4]$, and the input is constrained to be $u_k \leq 4.15$. In this example, the states of the LPV-A system shall be regulated from the initial state

$$x_0 = [-0.3964 \ 0.4377 \ -1.0905 \ 1.1137]$$

to the origin by means of two control methods: The explicit MPC scheme presented in Section III, and the Quasi-Min-Max MPC scheme proposed together with the original example in [28]. The evolution of the scheduling parameter is depicted in Fig. 1.

The Multi-Parametric Toolbox (MPT) and YALMIP were used to compute the explicit control law, [22], [29]. The weight matrices $Q = \text{diag}([1 \ 1 \ 1 \ 1])$, $R = 0.1$, $P = Q$, and a prediction horizon of $N = 4$ were chosen. The ∞ -norm was used in the cost function (4). The control law was computed for all states in the hyperbox $-10 \leq x_{k,i} \leq 1.14$, $i = 1, \dots, 4$. Afterwards, the resulting controller regions were merged using a greedy merge, resulting in a total number of 740 regions. For this control law a binary search tree was generated, which can be evaluated under C with small computational effort, [30]. MPT supports the exportation of binary search trees into C code, which can then be compiled as MEX functions callable from within MATLAB. In this example the MEX function requires 37 kB of memory. A pure C implementation would even further decrease the required computation times.

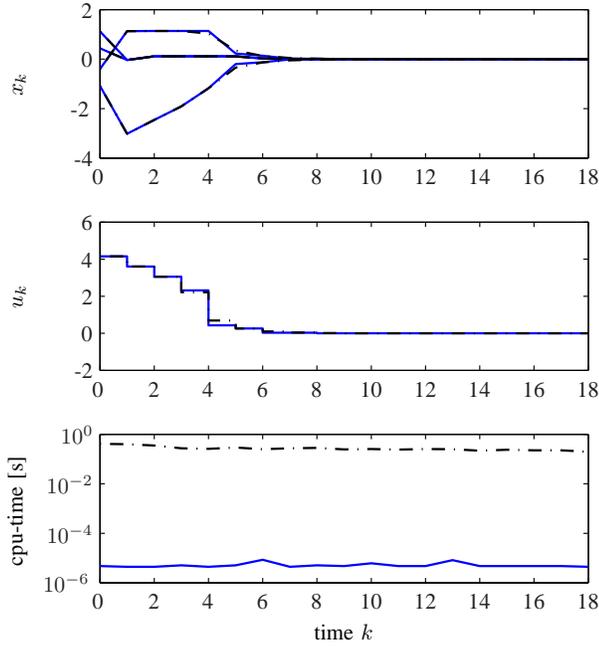


Fig. 2. Closed-loop trajectories of the states and the input, and the required online CPU time in Example 1. Comparison of Explicit LPV-A MPC (—) and Quasi-Min-Max MPC (---).

The solution of the semi-definite programs within the Quasi-Min-Max MPC scheme was performed by SEDUMI, [31], interfaced via YALMIP, [32].

The simulation results for both control methods are depicted in Fig. 2. It can be seen that the closed-loop trajectories of the states and the inputs are nearly the same, although two different objective functions were minimized (quadratic upper bound on the infinite horizon costs vs. piecewise linear finite-horizon costs). Under both control methods, the required input and state constraints are satisfied. The actual difference lies in the required online computational effort. While explicit MPC requires a computation time of less than 0.02 ms in each step, Quasi-Min-Max MPC requires between 0.2 s and 0.3 s, which is more than 4 orders of magnitude difference³.

B. Example 2: A-priori stability guarantees

In the second example we want to illustrate possible consequences of ensuring a-priori stability guarantees on the complexity of the explicit solution and on the size of the region of attraction. This example shall demonstrate that in some cases it is beneficial to verify stability a-posteriori instead. We consider an unstable LPV system with the system matrices

$$A_1 = \begin{bmatrix} 1.1 & 0 \\ 0.2 & 1.1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}, \quad (33a)$$

$$A_2 = \begin{bmatrix} 1.1 & 0 \\ 0.4 & 1.1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad (33b)$$

³All computations were performed on a 3 GHz Pentium 4 processor using MATLAB 7.

TABLE I
NUMBER OF CONTROLLER REGIONS WITH THE APPROACHES A-PRIORI AND A-POST FOR DIFFERENT PREDICTION HORIZONS IN EXAMPLE 2.

Prediction horizon N	2	3	4	5
No. of regions, a-priori	106	182	268	388
No. of regions, a-post	69	115	149	204

under the constraints

$$x_k \in \mathbb{X} = \{x_k \in \mathbb{R}^2 \mid \|x_k\|_\infty \leq 10\},$$

$$u_k \in \mathbb{U} = \{u_k \in \mathbb{R} \mid |u_k| \leq 1\}.$$

The weight matrices $Q = \text{diag}([1 \ 1])$, $R = 0.01$, and the ∞ -norm were employed in the cost function (4). The explicit control laws were computed for the prediction horizons $N = 2, \dots, 5$, and the worst-case costs were minimized in the final DP step. In the first approach, denoted by a-priori, terminal cost and terminal constraints were determined by means of the procedure described in Section V in order to guarantee stability a-priori for all feasible states. In the second approach, denoted by a-post, the terminal weight $P = Q$ and no terminal constraints were used, such that no a-priori stability guarantee can be given. Instead the region of attraction was determined afterwards by a reachability analysis.

The resulting complexities of the computed control laws are reported in Table I. The number of regions for a-priori is higher than the number of regions for a-post, which comes to no surprise, since terminal region constraints typically add to the solution complexity.

The size of the region of attraction for the different closed-loop systems is illustrated in Fig. 3. In the approach with a-priori stability guarantees, the region of attraction coincides with the set of feasible states. This set is indicated in black in Fig. 3. The region of attraction increases with increasing prediction horizon, but only slowly. A-post on the other hand results in larger regions of attraction. By omitting a-priori stability guarantees, the region of attraction is only a subset of the set of feasible states, but larger than the region of attraction of a-priori. Moreover one would require a much longer prediction horizon with a-priori to obtain similar regions of attraction, which typically results in even more controller regions.

C. Example 3: Explicit MPC of an LPV system

The third example system is a marginally stable LPV system of the form (1), and was taken from [33]. Contrary to [33], we will assume that the LPV paradigm holds, i.e. that the current scheduling parameter value is known to the controller. The system matrices of the LPV system are stated as

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (34a)$$

$$A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \quad (34b)$$

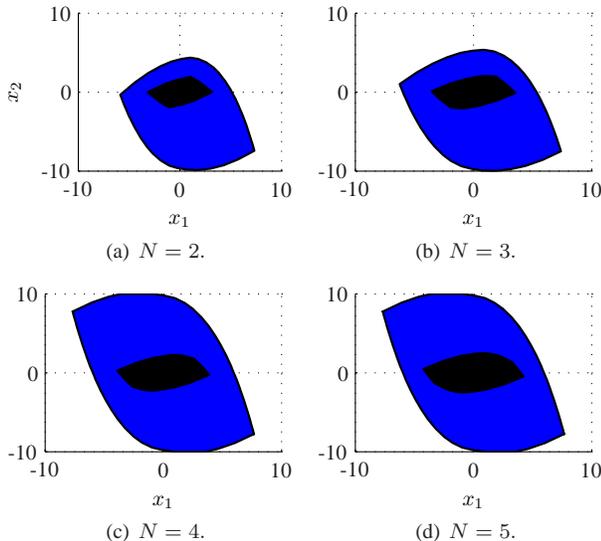


Fig. 3. Region of attraction of the closed-loop systems with the approaches a-priori (black) and a-post (blue) for different prediction horizons in Example 2.

and the system is subject to state and input constraints,

$$\begin{aligned} x_k \in \mathbb{X} &= \{x_k \in \mathbb{R}^2 \mid \|x_k\|_\infty \leq 10\}, \\ u_k \in \mathbb{U} &= \{u_k \in \mathbb{R} \mid |u_k| \leq 1\}. \end{aligned}$$

The weight matrices $Q = \text{diag}([1 \ 1])$, $R = 0.1$, $P = Q$ and a prediction horizon of $N = 4$ were chosen. The ∞ -norm was used in the cost function (4). Four methods are compared by means of control performance and online computational effort, when regulating the LPV system (34) to the origin.

- 1) The first method is the Quasi-Min-Max MPC method from [34]. Since the system has a parameter-varying input matrix, the predicted future control laws were chosen to be independent of the scheduling parameter in order to keep the semi-definite program linear in the scheduling parameter.
- 2) The second considered method is a workaround. In order to avoid a parameter-varying input matrix, an input delay is introduced and the system is augmented to the LPV-A system

$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} A(\theta) & B(\theta) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \tilde{u}_k. \quad (35)$$

Afterwards the arising parametric optimization problem for the augmented system (35) is solved optimally, resulting in a control law with 189 regions.

- 3) The computation of an explicit control law for the unaugmented LPV system (34) following the procedure in Section IV lead to an inappropriate high number of regions during the final DP step. Therefore the final DP step was adapted and instead of the worst-case costs, the average of the vertex costs at the vertices of the parameter simplex are minimized. Thus an explicit control law comprising 230 regions was computed.
- 4) As the last considered control method, the DP procedure is terminated before the last DP step, and the final DP

step (24) is solved online instead.

All four control methods were tested in simulations by controlling the system from 400 initial points, uniformly distributed over the hyperbox $\|x\|_\infty \leq 10$, to the origin. To account for the varying scheduling parameter, these test were repeated ten times, each time with different random scheduling parameter values. During the simulations the actually sustained quadratic costs,

$$J_{\text{act}} = \sum_{k=1}^{50} x_k^T Q x_k + u_k^T R u_k, \quad (36)$$

were collected over 50 time steps. Note that only the Quasi-Min-Max MPC approach minimizes a quadratic cost function, while all other tested approaches minimize the ∞ -norm, i.e. the performance criterion (36) is in favor of the Quasi-Min-Max MPC approach. Additionally, the average computation times per step were taken.

Table II presents this data for initial points which were feasible under all control methods. The average actual costs during the simulations show the drawback of introducing an input delay to the LPV system: The controller cannot react as quickly to the variations in the scheduling parameter, leading to a performance degradation. This degradation will in practice even be more severe, since disturbances were not considered during the simulations, which would have to be compensated for with delay. Although the actual costs were measured in quadratic norm, the Quasi-Min-Max MPC is less performant than the Explicit LPV-MPC approaches. This becomes more comprehensible, when one considers that the Quasi-Min-Max MPC scheme introduces some conservatism by considering a quadratic upper bound on the predicted costs, and moreover uses parameter-independent state feedback laws in the predictions. In this example this conservatism is on average more severe than the conservatism introduced by the explicit LPV-MPC schemes.

The computation times confirm the observations already made in Example 1 for the case of LPV-A systems: The evaluation of the control laws is orders of magnitude faster than the solution of the semi-definite programs in the Quasi-Min-Max MPC scheme. Computing the final step of the Explicit LPV-MPC scheme online reduces also the computation times compared to Quasi-Min-Max MPC, but with a factor of six not as significant as the completely explicit solution. The solution of the semi-definite programs within the Quasi-Min-Max MPC scheme was performed by SEDUMI, [31], interfaced via YALMIP, [32].

REFERENCES

- [1] W. Rugh and J. Shamma, "Research on Gain Scheduling," *Automatica*, vol. 36, no. 10, pp. 1401 – 1425, September 2000.
- [2] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. SIAM Studies in Applied Mathematics. Philadelphia, Pennsylvania: SIAM, 1994.
- [3] P. Apkarian, P. Gahinet, and G. Becker, "Self-Scheduled \mathcal{H}_∞ Control of Linear Parameter-Varying Systems: A Design Example," *Automatica*, vol. 31, no. 9, pp. 1251 – 1261, 1995.
- [4] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.
- [5] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, June 2001.

TABLE II

AVERAGE ACTUAL QUADRATIC COSTS AND ONLINE COMPUTATION TIMES FOR THE FOUR MPC APPROACHES IN EXAMPLE 3. NUMBER OF REGIONS AND MEMORY REQUIREMENTS OF THE EXPLICIT SOLUTIONS.

MPC controller	Quasi-Min-Max	Explicit (LPV-A)	Explicit	Semi-explicit
Average costs	376.14	445.03	366.23	363.58
t_{CPU} (mean)	148 ms	5.5 μ s	4.8 μ s	23.6 ms
Number of regions	–	189	230	128
Size of the explicit solution	–	20 kB	12 kB	7 kB

[6] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Non-Linear Parametric Optimization*. Berlin: Akademie-Verlag, 1982.

[7] A. Beccuti, S. Mariéthoz, S. Cliquenois, S. Wang, and M. Morari, “Explicit Model Predictive Control of DC-DC Switched Mode Power Supplies with Extended Kalman Filtering,” *IEEE Trans. on Industrial Electronics*, vol. 56, no. 6, pp. 1864 – 1874, Jun. 2009.

[8] T. Geyer, G. Papafotiou, and M. Morari, “Model Predictive Direct Torque Control - Part I: Concept, Algorithm and Analysis,” *IEEE Trans. on Industrial Electronics*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.

[9] A. Bemporad, F. Borrelli, and M. Morari, “Model Predictive Control Based on Linear Programming – The Explicit Solution,” *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974 – 1985, December 2002.

[10] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag, 2003, vol. 290.

[11] A. Bemporad, F. Borrelli, and M. Morari, “Min-Max Control of Constrained Uncertain Discrete-Time Linear Systems,” *IEEE Transactions on Automatic Control*, vol. 48, no. 9, pp. 1600 – 1606, September 2003.

[12] T. Besselmann, J. Löfberg, and M. Morari, “Explicit Model Predictive Control for Systems with Parameter-Varying State Transition Matrix,” in *IFAC World Congress*, Seoul, Korea, July 2008.

[13] —, “Explicit Model Predictive Control for Linear Parameter-Varying Systems,” in *Proceedings of the IEEE Conference on Decision & Control*, Cancun, Mexico, December 2008.

[14] —, “Explicit LPV-MPC with Bounded Rate of Parameter Variation,” in *IFAC Symposium on Robust Control Design*, Haifa, Israel, 2009.

[15] L. Silverman and H. Meadows, “Controllability and Observability in Time-Varyable Linear Systems,” *SIAM Journal on Control*, vol. 5, no. 1, pp. 64 – 73, 1967.

[16] G. Balas, J. Bokor, and Z. Szabó, “Invariant Subspaces for LPV Systems and Their Applications,” *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 2065 – 2069, Nov. 2003.

[17] J. Löfberg, “Approximations of closed-loop MPC,” in *IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2003, pp. 1438–1442.

[18] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[19] M. Kon-Popovska, “On Some Aspects of the Matrix Data Perturbation in Linear Program,” *Yugoslav Journal of Operations Research*, vol. 13, no. 2, pp. 153 – 164, 2003.

[20] 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.

[21] 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, no. 1 – 2, pp. 221 – 229, October 2001.

[22] J. Löfberg, “Automatic Robust Convex Programming,” *Optimization Methods and Software*, 2012, accepted for publication.

[23] A. Bemporad and C. Filippi, “An Algorithm for Approximate Multi-parametric Convex Programming,” vol. 35, pp. 87 – 108, 2006.

[24] T. Besselmann, “Constrained Optimal Control - Piecewise Affine and Linear Parameter-Varying Systems,” Ph.D. dissertation, Automatic Control Laboratory, ETH Zurich, 2010.

[25] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained Model Predictive Control: Stability and Optimality,” *Automatica*, vol. 36, no. 6, pp. 789 – 814, June 2000.

[26] P. Scokaert, D. Mayne, and J. Rawlings, “Suboptimal Model Predictive Control (Feasibility Implies Stability),” *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648 – 654, March 1999.

[27] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, T. Basar, Ed. Boston: Birkhäuser, 2008.

[28] Y. Lu and Y. Arkun, “A Scheduling Quasi-Min-Max MPC for LPV Systems,” in *Proceedings of the American Control Conference*, San Diego, USA, June 1999.

[29] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari, “Multi-Parametric Toolbox (MPT),” in *Proceedings of the International Workshop on Hybrid Systems: Computation and Control*, Philadelphia, USA, Mar. 2004, pp. 448–462.

[30] P. Tøndel, T. Johansen, and A. Bemporad, “Evaluation of Piecewise Affine Control via Binary Search Tree,” *Automatica*, vol. 39, no. 5, pp. 945 – 950, May 2003.

[31] J. Sturm, “Using SeDuMi 1.02, a Matlab Toolbox for Optimization over Symmetric Cones,” *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.

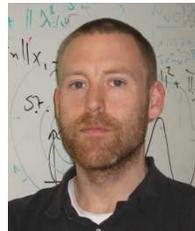
[32] J. Löfberg, “YALMIP : A Toolbox for Modelling and Optimization in MATLAB,” in *CCA/ISIC/CACSD*, Taipei, Taiwan, Sep. 2004.

[33] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor, “Interpolation Based MPC for LPV Systems Using Polyhedral Invariant Sets,” in *Proceedings of the American Control Conference*, Portland, OR, USA, June 2005.

[34] Y. Lu and Y. Arkun, “Quasi-Min-Max MPC Algorithms for LPV Systems,” *Automatica*, vol. 36, no. 4, pp. 527 – 540, 2000.



Thomas Besselmann was born 1980 in Thuine, Germany. In 2003 he received his B.Sc. degree in General Engineering Science and 2005 his diploma degree in Mechatronics from Hamburg University of Technology in Hamburg, Germany. In 2010 he received his Ph.D. degree at the Automatic Control Laboratory, ETH Zurich, Switzerland. Currently he is employed as Scientist at ABB Corporate Research, Baden-Dättwil, Switzerland. His research interests include high-speed control of constrained systems, in particular model predictive control methods, and their application to automotive and power electronic systems.



Johan Löfberg was born in Jönköping, Sweden in 1974. He received his M.Sc. degree in Mechanical Engineering in 1998 and the Ph.D. degree in Automatic Control in 2003, both from Linköping University, Linköping, Sweden. He held a post-doctoral position at the Automatic Control Laboratory at ETH Zurich, Switzerland 2003-2006. From 2006, he works as an Assistant Professor at Linköping University. His research interests include optimal and robust control in a model predictive control framework, and general aspects related to the use of optimization in control. He is the author and developer of the modeling language YALMIP.



Manfred Morari was appointed head of the Department of Information Technology and Electrical Engineering at ETH Zurich in 2009. He was head of the Automatic Control Laboratory from 1994 to 2008. Before that he was the McCollum-Corcoran Professor of Chemical Engineering and Executive Officer for Control and Dynamical Systems at the California Institute of Technology. He obtained the diploma from ETH Zurich and the Ph.D. from the University of Minnesota, both in chemical engineering. His interests are in hybrid systems and the control of biomedical systems. In recognition of his research contributions he received numerous awards, among them the Donald P. Eckman Award, the John R. Ragazzini Award and the Richard E. Bellman Control Heritage Award of the American Automatic Control Council, the Allan P. Colburn Award and the Professional Progress Award of the AIChE, the Curtis W. McGraw Research Award of the ASEE, Doctor Honoris Causa from Babes-Bolyai University, Fellow of IEEE, IFAC and AIChE, the IEEE Control Systems Technical Field Award, and was elected to the National Academy of Engineering (U.S.). Manfred Morari has held appointments with Exxon and ICI plc and serves on the technical advisory boards of several major corporations.