

# Explicit MPC of LPV Systems in the Controllable Canonical Form

Michal Kvasnica, Alexander Szücs, Miroslav Fikar, and Ján Drgoňa

**Abstract**—We exploit the controllable canonical form of single-input linear parameter-varying (LPV) systems to synthesize explicit Model Predictive Control (MPC) feedback laws. The nonlinear state-parameter dependence is first moved into the feedback term, followed by devising a suitable input constraint set. This allows the MPC problem to be formulated with a quadratic performance index, avoiding costly dynamic programming iterations. The resulting MPC optimization is shown to be convex and can be solved parametrically. The explicit solution takes a form of a piecewise affine (PWA) function which allows to implement the feedback law quickly even with limited computational resources.

## I. INTRODUCTION

Linear parameter-varying (LPV) systems describe a family of linear systems parametrized by a vector of (typically bounded) parameters. Such systems are popular especially due to their ability to capture time-dependent change of system's behavior and influence of nonlinearities [10]. LPV systems are typically controlled by gain-scheduling approaches, see [1] and references therein. However, if constraints on states, inputs and/or outputs are to be maintained by the controller, one has to resort to more elaborated strategies, such as to Model Predictive Control (MPC). Here, the challenge is represented by a nonlinear relationship between parameters of the LPV system and *predictions* of its states. Although the parameters are assumed to be measured and made known to the controller, they are allowed to fluctuate arbitrarily in time. Hence the controller has to satisfy constraints in a robust fashion while coping with lack of convexity. To formulate LPV-MPC problems as convex optimization problems, one can resort to quasi-min-max formulations [16], use interpolation-based strategies [19], apply gain-scheduling [10], or employ parameter-dependent control laws [22]. Although such approaches provide guarantees of closed-loop stability and constraint satisfaction, they only define the feedback law *implicitly* as a numerical solution to an optimization problem, typically in a linear matrix inequality (LMI) form. The optimization then needs to be carried out recursively at each sampling instant. Such a requirement is often a bottleneck if the system dynamics is fast, or if a slow, but cheap implementation hardware is used.

To overcome this difficulty, the seminal work [3] showed how to solve certain classes of convex MPC problems *explicitly* by employing parametric optimization [12], [7]. Here, the optimal solution to a given MPC optimization problem is pre-computed off-line for all feasible initial conditions. If the

prediction model is linear, the pre-computed solution takes a form of PWA function, which maps state measurements to optimal control inputs. The advantage of such explicit solutions stems from the fact that the on-line implementation of MPC boils down to a mere function evaluation.

The idea of explicit MPC was extended to LPV systems by [2], [4], [5], [6], [20]. Here, in order to provide robust satisfaction of state and input constraints for an arbitrary value of the LPV parameters, the explicit representation of the MPC feedback law is constructed in a dynamic programming (DP) fashion, iterating backwards in time. If the performance index to be minimized is given as a sum of 1- and/or  $\infty$ -norms, the explicit solution is a PWA function. Such explicit LPV-MPC controllers furthermore exploit measurements of system's parameters to improve performance. A common drawback of the DP-based methods is that they do not allow for quadratic terms in the performance index, or otherwise the feedback law would no longer be PWA over polytopes [7]. This has two principal implications. First, explicit MPC solutions with quadratic performance indices are typically significantly simpler compared to their counterparts based on  $1/\infty$  norms [7]. The difference in complexity is mainly due to the fact that additional variables and constraints need to be introduced to model the epigraph of a performance index based on  $1/\infty$  norms. Second, and more importantly, absence of support for quadratic terms does not allow to provide a-priori guarantees of closed-loop stability by standard arguments based on quadratic terminal penalties. Instead, stability can only be tested a-posteriori. The only exception is [6] where stability can be enforced a-priori by employing PWA Minkowski functions. However, the procedure involves robust optimization over polynomials and is computationally expensive.

In this work we show how to synthesize explicit LPV-MPC controllers without resorting to dynamic programming. As a consequence, our formulation allows to employ quadratic performance objectives, which are more natural in practice. To achieve this goal, we exploit the fact that for single-input LPV systems in the controllable canonical form [18], the nonlinear state-parameter dependence can be transferred into a nonlinear feed-forward term [9], [13]. By devising a specially-crafted input constraint set to account for the nonlinearity, we formulate the MPC optimization problem such that state and input constraints are satisfied in a robust fashion for arbitrary values of system's parameters. Moreover, we show how the controller exploits measurements of system's parameters to improve performance.

The authors are with Slovak University of Technology in Bratislava, Slovakia; {michal.kvasnica, alexander.szucs, miroslav.fikar, jan.drgona}@stuba.sk

## Notation and Definitions

For a vector  $z$ ,  $\|z\|_1 := \sum_i |z_i|$  and  $\|z\|_\infty := \max |z_i|$  denote respectively the vector 1- and  $\infty$ -norms. For  $\|z\|_2$  we adopt the notion that  $\|z\|_2 = z^T z$ . A polytope is the bounded convex intersection of a finite number of affine half-spaces, i.e.,  $\mathcal{R} := \{x \in \mathbb{R}^{n_x} \mid Fx \leq g\}$ . We call the collection of polytopes  $\{\mathcal{R}_i\}_{i=1}^R$  the *partition* of a polytope  $\mathcal{R}$  if  $\mathcal{R} = \bigcup_{i=1}^R \mathcal{R}_i$ , and  $\text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset$  for all  $i \neq j$ . Function  $\kappa(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$  with  $x \in \mathcal{R} \subset \mathbb{R}^{n_x}$  is called piecewise affine over polytopes if  $\{\mathcal{R}_i\}_{i=1}^R$  is the partition of a polytope  $\mathcal{R}$  and

$$\kappa(x) := K_i x + L_i \quad \forall x \in \mathcal{R}_i, \quad (1)$$

with  $K_i \in \mathbb{R}^{n_z \times n_x}$ ,  $L_i \in \mathbb{R}^{n_z}$ , and  $i = 1, \dots, R$ . PWA function  $\kappa(x)$  is continuous if  $K_i x + L_i = K_j x + L_j$  holds  $\forall x \in \mathcal{R}_i \cap \mathcal{R}_j$ ,  $i \neq j$ .

*Lemma 1.1 ([7]):* Consider an optimization problem

$$u^* = \arg \min_u \{J(u, x) \mid Gu \leq W + Ex\}, \quad (2)$$

with  $J(u, x)$  being either a convex quadratic, or a linear function of the optimization variables  $u \in \mathbb{R}^{n_u}$  and the parameters  $x \in \mathbb{R}^{n_x}$ . Then the optimizer  $u^*$  is a continuous PWA function of  $x$ , i.e.,  $u^* = \kappa(x)$  with  $\kappa(\cdot)$  as in (1). ■

## II. PROBLEM STATEMENT

We consider control of single-input discrete-time LPV systems of the form

$$x^+ = A(\lambda)x + Bu, \quad (3)$$

where  $x \in \mathbb{R}^{n_x}$  is the state vector,  $x^+$  is the successor state,  $u \in \mathbb{R}$  is the control input, and  $\lambda \in \mathbb{R}^{n_\lambda}$  are system's parameters. We assume that system (3) is in the controllable canonical form where

$$A(\lambda) := \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_{n_x} \end{bmatrix}, \quad B := \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (4)$$

It is assumed that system (3) operates under hard state and input constraints

$$x \in \mathcal{X}, \quad u \in \mathcal{U}, \quad (5)$$

where  $\mathcal{X} \subset \mathbb{R}^{n_x}$  and  $\mathcal{U} \subset \mathbb{R}$  are polytopes. The system's parameters  $\lambda$  are assumed to take arbitrary values from a polytopic set  $\Delta \subset \mathbb{R}^{n_\lambda}$ . Finally, we will assume that at each time instant we can measure the system's state  $x$  and its parameters  $\lambda$ .

*Remark 2.1:* Canonical LPV forms (4) arise naturally e.g. when the controlled process is modeled [21] or identified [23] as a discrete-time transfer function

$$\frac{Y(z)}{U(z)} = \frac{b_m z^m + b_1 z^{m-1} + \cdots + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_0}, \quad (6)$$

with known variations of the denominator's coefficients, e.g.  $\underline{a}_i \leq a_i \leq \bar{a}_i$ . Here,  $\lambda_i = -a_{i-1}$ , and  $\Delta$  is determined

from the bounds. Alternatively, any controllable system of the form

$$x^+ = \Gamma x + \Xi u \quad (7)$$

can be converted into (4) by a suitable choice of a transformation matrix [18]. □

For the system (3)–(4) we aim at devising an explicit representation of the closed-loop MPC feedback policy which will utilize measurements of  $x$  and  $\lambda$  (or some combination thereof) to achieve satisfaction of constraints in (5) and minimization of a selected performance index. Formally, the MPC control problem to be solved is given by

$$\min \|Q_N x_N\|_p + \sum_{k=0}^{N-1} \|Q_x x_k\|_p + \|Q_u u_k\|_p \quad (8a)$$

$$\text{s.t. } x_{k+1} = A(\lambda_k)x_k + Bu_k, \quad (8b)$$

$$u_k \in \mathcal{U}, \quad (8c)$$

$$x_k \in \mathcal{X}, \quad (8d)$$

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

$$\lambda_k \in \Delta, \quad (8f)$$

where  $x_k$  and  $u_k$  denote, respectively, the  $k$ -th step predictions of system's states and inputs over some finite prediction horizon  $N$ .  $\mathcal{T} \subset \mathcal{X}$  is a suitable polytopic terminal set, and  $Q_N$ ,  $Q_x$ ,  $Q_u$  are weighting matrices of appropriate dimensions. If  $p = 2$  in (8a), then we equivalently minimize  $z^T Q z$  instead of  $\|Q z\|_2$ . Note that in such a case  $Q_N \succeq 0$ ,  $Q_x \succeq 0$ ,  $Q_u \succ 0$  is required in order for (8a) to be strictly convex. The minimization is performed over  $u_0, \dots, u_{N-1}$ , initialized by measurements of  $x_0$  and  $\lambda_0$ . In the spirit of receding horizon closed-loop implementation of MPC [17] we are interested in finding an explicit representation of the function  $\kappa(\cdot)$  which maps measurements of  $x_0$  and  $\lambda_0$  to the first optimal control input  $u_0^*$ , i.e.,

$$u_0^* = \kappa(x_0, \lambda_0). \quad (9)$$

The main difficulty of obtaining  $\kappa(\cdot)$  in (9) follows from the fact that constraints (8b) are non-linear due to a multiplication between the predicted states and the parameters  $\lambda$ , both of which are considered as free variables in the optimization problem. Therefore, standard results in multi-parametric programming are not directly applicable to solve (8) in a parametric fashion via Lemma 1.1.

## III. MAIN RESULTS

To devise an explicit representation of the receding horizon feedback policy in (9) we exploit the canonical description in (4). In particular, it is easy to verify that system (3) with  $A$  and  $B$  as in (4) admits an equivalent representation [9], [13]

$$x^+ = \tilde{A}x + B\tilde{u} \quad (10)$$

with

$$\tilde{A} := \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad (11)$$

and

$$\tilde{u} = u + x^T \lambda. \quad (12)$$

*Remark 3.1:* We note that the matrix  $\tilde{A}$  in (11) is constant and does not depend on the bounds  $\Delta$ . The equivalence between (3) and (10) can be immediately observed by substituting for  $\tilde{u}$  the expression in (12) with  $B$  as in (4).  $\square$

The term  $x^T \lambda$  in (12) can be interpreted as a nonlinear feed-forward action. With such a description, the MPC problem (8) becomes

$$\min \|Q_N x_N\|_p + \sum_{k=0}^{N-1} \|Q_x x_k\|_p + \|Q_u(\tilde{u}_k - x_k^T \lambda_k)\|_p \quad (13a)$$

$$\text{s.t. } x_{k+1} = \tilde{A}x_k + B\tilde{u}_k, \quad (13b)$$

$$x_k \in \mathcal{X}, \quad (13c)$$

$$x_N \in \mathcal{T}, \quad (13d)$$

$$\tilde{u}_k - x_k^T \lambda_k \in \mathcal{U}, \quad \forall \lambda_k \in \Delta. \quad (13e)$$

The optimized variables are  $\tilde{u}_0, \dots, \tilde{u}_{N-1}$ . Note that (13a) and (13e) follow directly from (12) by expressing  $u = \tilde{u} - x^T \lambda$ . Due to the bilinearity, however, problem (13) is still non-convex.

To overcome this difficulty, and to reformulate (13) as a convex optimization problem with linear constraints, we propose to devise a special polytopic constraint set  $\mathcal{V} \subset \mathbb{R}^{n_x + n_u}$  in the joint state-input space such that

$$\left( \begin{bmatrix} x \\ \tilde{u} \end{bmatrix} \in \mathcal{V} \right) \Rightarrow (\tilde{u} - x^T \lambda \in \mathcal{U}, x \in \mathcal{X}, \forall \lambda \in \Delta). \quad (14)$$

The set  $\mathcal{V}$  thus serves to guarantee robust satisfaction of input and state constraints for any value of  $\lambda$  from  $\Delta$ .

*Theorem 3.2:* Suppose a polytope  $\mathcal{V}$  satisfying (14) is given. Reformulate (13) as

$$\min \|Q_N x_N\|_p + \sum_{k=0}^{N-1} \|Q_x x_k\|_p + \|Q_u \tilde{u}_k\|_p \quad (15a)$$

$$\text{s.t. } x_{k+1} = \tilde{A}x_k + B\tilde{u}_k, \quad (15b)$$

$$\begin{bmatrix} x_k \\ \tilde{u}_k \end{bmatrix} \in \mathcal{V}, \quad (15c)$$

$$x_N \in \mathcal{T}. \quad (15d)$$

Then:

P1: any feasible solution to (15) is a feasible solution to (8);

P2: (15) is a convex optimization problem with linear constraints;

P3: the explicit receding horizon feedback law is

$$u_0^* = \kappa(x_0) - x_0^T \lambda_0; \quad (16)$$

P4:  $\kappa(\cdot)$  in (16) is a PWA function of  $x_0$ .

*Proof:* For any  $x_k, \tilde{u}_k$  pair residing in  $\mathcal{V}$  we have that  $u_k \in \mathcal{U}$  and  $x_k \in \mathcal{X}$  for all  $\lambda_k \in \Delta$  by (14). Hence (15d) is an inner approximation of (8f). Moreover, (15b) is equivalent to (8b) via (10)–(12). Therefore any admissible solution to (15) is also feasible (though, not necessarily optimal) for (8).

P2 is imminent since (15b) is linear in  $x_k$  and  $\tilde{u}_k$ , and the sets  $\mathcal{V}$  and  $\mathcal{T}$  in (15c)–(15d) are polytopes, hence representable by a finite number of linear inequalities. Moreover, the performance index is convex for it being a sum of convex functions.

P3 follows directly from rewriting (12) as  $u^* = \tilde{u}^* - x^T \lambda$  and by representing  $\tilde{u}^* = \kappa(x)$ .

Finally, since (15) is either a quadratic program (if  $p = 2$ ) or a linear program (for  $p \in \{1, \infty\}$ ), P4 is a direct consequence of Lemma 1.1.  $\blacksquare$

*Remark 3.3:* In simple terms, the set  $\mathcal{V}$  is the inner polytopic approximation of the non-convex set

$$\mathcal{S} = \{x, \tilde{u} \mid \exists \lambda \in \Delta \text{ s.t. } \tilde{u} - x^T \lambda \in \mathcal{U}, x \in \mathcal{X}\}, \quad (17)$$

which describes all  $x, \tilde{u}$  pairs for which constraint satisfaction can be shown for *some*, but not necessarily for all, values of  $\lambda$ . Implication (14) means that  $\mathcal{V} \subset \mathcal{S}$ .  $\square$

In the subsequent sections we address two important aspects related to formulation of the optimization problem in (15). First, in Section III-A we describe a procedure for devising the set  $\mathcal{V}$  satisfying (14). Then, in Section III-B we discuss measures which allow to reduce suboptimality of (15).

#### A. Design of the set $\mathcal{V}$

Given are polytopic sets  $\mathcal{U}$ ,  $\mathcal{X}$ , and  $\Delta$ . We want to construct the set  $\mathcal{V}$  such that (14) holds. Since  $\Delta$  is assumed to be a polytope, it can always be represented [8] as a convex hull of its vertices  $\delta_i \in \mathbb{R}^{n_x}$ ,  $i = 1, \dots, n_\delta$ , i.e.,

$$\Delta = \left\{ \sum_{i=1}^{n_\delta} \mu_i \delta_i \mid \sum_{i=1}^{n_\delta} \mu_i = 1, \mu_i \geq 0, i = 1, \dots, n_\delta \right\}. \quad (18)$$

*Lemma 3.4:* Let

$$\mathcal{V}_i = \left\{ \begin{bmatrix} x \\ \tilde{u} \end{bmatrix} \mid x \in \mathcal{X}, \tilde{u} - x^T \delta_i \in \mathcal{U} \right\}, \quad i = 1, \dots, n_\delta. \quad (19)$$

Then

$$\mathcal{V} = \mathcal{V}_1 \cap \dots \cap \mathcal{V}_{n_\delta}. \quad (20)$$

is a polytope which satisfies (14).

*Proof:* First note that  $\mathcal{V}_i$  are convex polytopes because  $\mathcal{X}$  and  $\mathcal{U}$  in (19) are polytopes and  $\delta_i$  are fixed. It follows that  $\mathcal{V}$  is a polytope for it is given as an intersection of finitely many polytopes. Now take an arbitrary  $\begin{bmatrix} x \\ \tilde{u} \end{bmatrix} \in \mathcal{V}$  which by (20) means that  $\begin{bmatrix} x \\ \tilde{u} \end{bmatrix} \in \mathcal{V}_i$  for all  $i = 1, \dots, n_\delta$ . Moreover,  $x \in \mathcal{X}$  directly by (19). As in (18), represent  $\lambda = \sum_i \mu_i \delta_i$ ,  $\mu \in \Lambda$ ,  $\Lambda = \{\mu \in \mathbb{R}^{n_\delta} \mid \sum_i \mu_i = 1, \mu_i \geq 0\}$ . Then (14) holds if

$$\forall \mu \in \Lambda: \tilde{u} - x^T \sum_i^{n_\delta} \mu_i \delta_i \in \mathcal{U}. \quad (21)$$

Pick an arbitrary  $i$  and set  $\mu_i = 1$ ,  $\mu_j = 0$  for all  $j \neq i$ . Then (21) reduces to  $\tilde{u} - x^T \delta_i \in \mathcal{U}$ , which trivially holds for all  $i$  by (19) and (20). Moreover, since  $\mathcal{V}$  is convex, we have that (21) holds of an arbitrary  $\mu \in \Lambda$ , completing the proof.  $\blacksquare$

The procedure for computing  $\mathcal{V}$  therefore requires following steps:

- 1) Compute vertices  $\delta_i$  of the set  $\Delta$ .
- 2) Form polytopes  $\mathcal{V}_i$  per (19).
- 3) Construct  $\mathcal{V}$  via (20) by intersecting all polytopes  $\mathcal{V}_i$ .

Computationally-wise, enumeration of vertices in the first step is considered a hard task in general. However, for problems of moderate dimension (say, in dimensions smaller than 8), the task can be accomplished efficiently using off-the-shelf software, such as with the CDD library [11]. Formulation of polytopes and computing their intersection boils down to solving a series of linear programs, and can be accomplished e.g. by the MPT toolbox [14].

### B. Reduction of suboptimality

Although by P1 of Theorem 3.2 an optimal solution to MPC problem (15) is feasible for the original setup (8), it is not necessarily optimal with respect to the performance index (8a). The reason being that while the non-convex formulation (8) tackles the knowledge of  $\lambda$  in its performance index, the convex counterpart (15) only provides robust satisfaction of state and input constraints, combined with minimizing a “nominal” performance criterion (15a). In other words, while (8a) optimizes performance w.r.t. particular measurements of  $\lambda$ , (15a) does not account for them.

One popular way to mitigate the induced loss of performance is to account for the impact of  $\lambda$  on the first predicted state by means of a so-called *information variable* [2], [5]. Here, the prediction equation (15b) is split into two parts:

$$x_1 = A(\lambda_0)x_0 + B\tilde{u}_0, \quad (22a)$$

$$x_{k+1} = \tilde{A}x_k + B\tilde{u}_k, \quad k = 1, \dots, N-1, \quad (22b)$$

with  $\tilde{u}_0 \equiv u_0$ . Since both  $x_0$  and  $\lambda_0$  are assumed to be measured, the product  $A(\lambda_0)x_0$  can be evaluated at any time instant. Introduce a substitution  $z = A(\lambda_0)x_0$  and consider  $z \in \mathbb{R}^{n_x}$  a free variable. Then the MPC problem (15) can be rewritten as

$$\min \|Q_N x_N\|_p + \sum_{k=0}^{N-1} \|Q_x x_k\|_p + \|Q_u \tilde{u}_k\|_p \quad (23a)$$

$$\text{s.t. } x_1 = z + B\tilde{u}_0, \quad (23b)$$

$$x_{k+1} = \tilde{A}x_k + B\tilde{u}_k, \quad k = 1, \dots, N-1, \quad (23c)$$

$$\tilde{u}_0 \in \mathcal{U}, \quad (23d)$$

$$\begin{bmatrix} x_k \\ \tilde{u}_k \end{bmatrix} \in \mathcal{V}, \quad k = 1, \dots, N-1, \quad (23e)$$

$$x_k \in \mathcal{X}, \quad (23f)$$

$$x_N \in \mathcal{T}. \quad (23g)$$

Problem (23) is a parametric optimization problem with  $z$  representing the parameters. Since  $z$  influences  $x_1$ , which is penalized in (23a), we at least partially incorporate particular measurements of  $\lambda$  into the cost function. The receding horizon MPC policy for LPV system (3) then becomes

$$u_k^* = \kappa(z). \quad (24)$$

Here, the knowledge of the state and parameter measurements is already incorporated in the information variable  $z$ . Since all constraints of (23) are linear, and since the

optimization objective (23a) is either a convex quadratic (for  $p = 2$ ) or a linear function (for  $p \in \{1, \infty\}$ ), it follows directly from Lemma 1.1 that the function  $\kappa(\cdot)$  in (24) is a PWA function of the parameters  $z$ .

## IV. EXAMPLE

Consider a 2-states, single-input discrete-time LPV system

$$x^+ = \begin{bmatrix} 0 & 1 \\ \lambda & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad (25)$$

which is subject to constraints  $\mathcal{X} = \{x \in \mathbb{R}^2 \mid -5 \leq x_i \leq 5, i = 1, 2\}$ , and  $\mathcal{U} = \{u \in \mathbb{R} \mid -2 \leq u \leq 2\}$ . The parameter  $\lambda \in \Delta$  is bounded by  $\Delta = \{\lambda \in \mathbb{R} \mid -1.5 \leq \lambda \leq -0.5\}$ . It is worth noting that system (25) is open-loop unstable for  $-1.5 \leq \lambda \leq -1$ .

With vertices of  $\Delta$  being  $\delta_1 = -1.5$  and  $\delta_2 = -0.5$ , we computed the set  $\mathcal{V}$  from (20) as

$$\mathcal{V} = \left\{ \begin{bmatrix} x \\ \tilde{u} \end{bmatrix} \in \mathbb{R}^3 \mid \begin{aligned} -1 &\leq -0.25x_1 + 0.5x_2 - 0.5\tilde{u} \leq 1, \\ -1 &\leq -0.75x_1 + 0.75x_2 - 0.5\tilde{u} \leq 1, \\ -5 &\leq x_2 \leq 5 \end{aligned} \right\}. \quad (26)$$

The LPV-MPC problem (15) was formulated with  $Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ,  $Q_u = 1$ ,  $Q_N = \begin{bmatrix} 4.6714 & -0.7621 \\ -0.7621 & 9.3591 \end{bmatrix}$ , and with the terminal set

$$\mathcal{T} = \{x \mid -1 \leq 0.2160x_1 + 0.3741x_2 \leq 1, \\ -1 \leq 0.4655x_1 - 0.3796x_2 \leq 1\}. \quad (27)$$

Two explicit LPV-MPC controllers were then synthesized for a quadratic performance objective (i.e.,  $p = 2$  in (15a) and (23a)) and the prediction horizon  $N = 5$ . The first controller, denoted in the sequel as `reg1`, was obtained by solving (15), resulting in a receding horizon feedback (16). The second controller, designated as `reg2`, was derived by solving (23). Here, the feedback is given by (24) with  $z = A(\lambda_0)x_0$ . Advantage of `reg2` over `reg1` is that the former exploits measurements of  $\lambda$  to improve performance, cf. Section III-B. Both explicit controllers were obtained by solving the corresponding optimization problem parametrically by the MPT toolbox in less than one second. For `reg1`, the PWA function  $\kappa(\cdot)$  in (16) is defined over 67 polytopes in the 2-dimensional state space, which are depicted in Fig. 1. For `reg2` the function  $\kappa(\cdot)$  in (24) consisted of 65 regions in the space of the 2-dimensional  $z$  variable.

To validate the controllers, we have performed closed-loop simulations employing the system (25). The initial condition was chosen as  $x_0 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$ . In the simulation we have assumed that  $\lambda$  in (25) varies as in Fig. 2(d). To quantify performance of `reg1` and `reg2`, we have also constructed `reg_base`, which solves (8) numerically at each sampling instant, assuming a perfect knowledge of the future values of  $\lambda$ . The closed-loop profiles of states and inputs of (25) under these three controllers are shown in Fig. 2. As can be observed, both explicit controllers stabilize the plant to the origin despite varying  $\lambda$ . Moreover, the response under `reg2`, which utilizes measurements of  $\lambda$ , is practically identical to performance of `reg_base`, which solves the

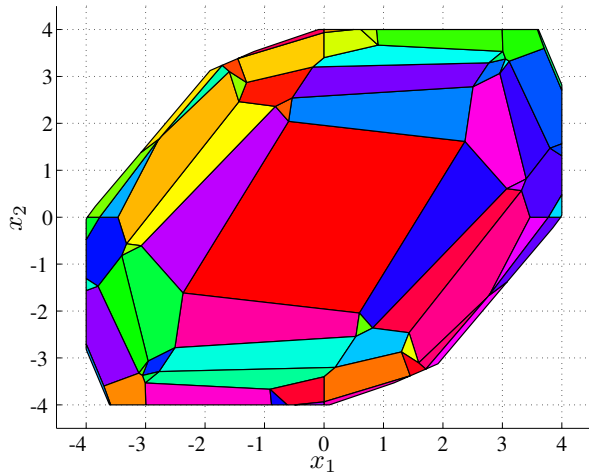


Fig. 1. Polytopes constituting the domain of  $\kappa(x)$  in (16) for the regulation controller `reg1`.

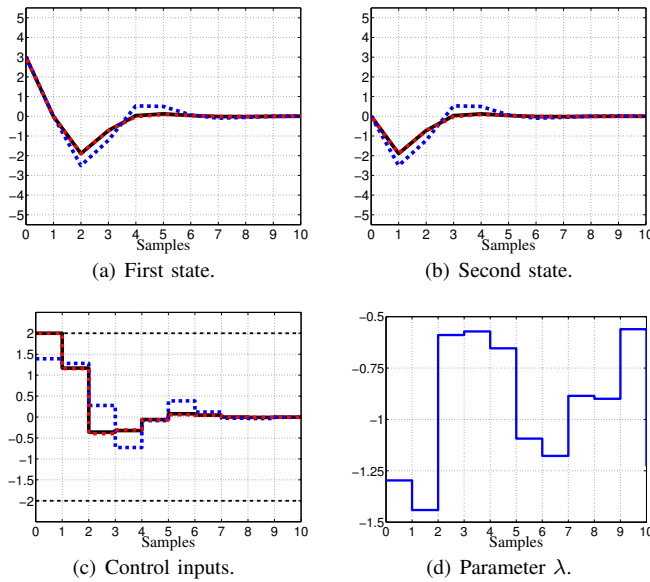
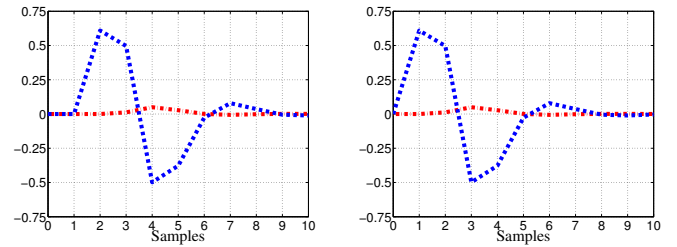


Fig. 2. Simulation results for the regulation scenario with  $x_0^T = (3, 0)$ . Solid black line represents response under `reg_base`, blue dashed line corresponds to `reg1`, and red dash-dotted line is for `reg2`. Note that the response of `reg2` is practically identical to `reg_base`.

original problem (8). The congruence is further stressed in Fig. 3, which shows the differences of state profiles under `reg1` and `reg2` versus the base-line controller `reg_base`. Here, `reg2` achieves a very good match. It is worth noting that computation of the optimal control input from (16) and (24) took on average just 2 microseconds by evaluating  $\kappa(\cdot)$  via (1).

#### A. Comparison to [6]

In [6] the authors proposed to obtain an explicit solution to (8) by devising an iterative dynamic-programming scheme. To guarantee closed-loop stability, a PWA Minkowski function induced by a polyhedral terminal set was employed. Furthermore, to guarantee robust satisfaction of constraints in (8f) for any value of  $\lambda$ , the nonlinear  $x$ - $\lambda$  dependence is



(a) Differences of the first state. (b) Differences of the second state.

Fig. 3. Differences of state profiles under `reg1` (blue dash-dotted line) and `reg2` (red dash-dotted line) versus the fully optimal controller `reg_base`.

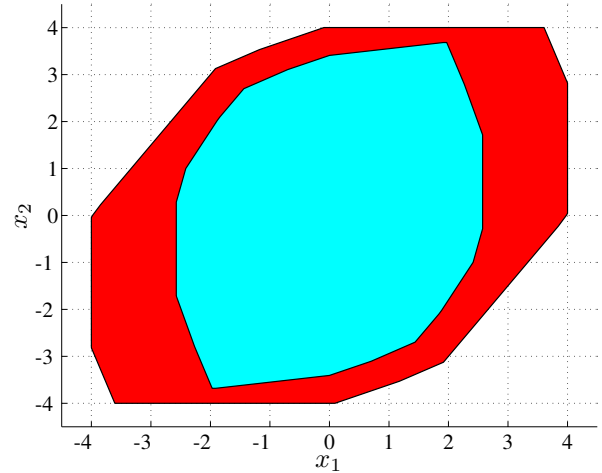


Fig. 4. Feasible set of `reg1` (red) versus the feasible set of [6], shown in cyan color.

relaxed by utilizing Polya polynomials. Then one can cast (8) as a series of parametric linear problems with  $p \in \{1, \infty\}$  in (8a). For the particular example of this paper, and with the code available in [15], the explicit representation of the feedback law in (24) was obtained in 119 seconds as a PWA function defined over 112 polytopes. This should be compared to 1 second and 67 polytopes reported for `reg2` above. Interesting to note is that the feasible set of our approach is significantly larger compared to the approach of [6], as shown graphically in Fig. 4.

## V. CONCLUSIONS

We have shown how to exploit the controllable canonical form of LPV systems to formulate an MPC optimization problem which does not require dynamic-programming iterations. The underlying idea lies in moving the nonlinear state-parameter dependence to the feedback term. Followed by devising a suitable input constraint set  $\mathcal{V}$  from (14) we can then guarantee robust satisfaction of state and input constraints for all possible values of the parameters  $\lambda$ . By resorting to convex polytopic sets  $\mathcal{V}$  we have furthermore showed that the MPC problem can be solved using parametric programming, resulting in an explicit representation of the feedback law. Knowledge of measurements of  $\lambda$  can be incorporated in the feedback law to improve its performance. The principal downside of the proposed approach follows from the fact

that only LPV systems with a single input can be treated in this way. Abolishing this limitation is currently under active research.

#### ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11 and of the Scientific Grant Agency of the Slovak Republic under the grant APVV 0551-11.

#### REFERENCES

- [1] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled  $H_\infty$  control of linear parameter-varying systems: A design example. *Automatica*, 31(9):1251–1261, 1995.
- [2] M. Baric, Sasa V. Rakovic, Th. Besselmann, and M. Morari. Max-min optimal control of constrained discrete-time systems. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea.
- [3] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002.
- [4] Th. Besselmann, J. Löfberg, and M. Morari. Explicit MPC for LPV systems. In *Conference on Decision and Control, CDC*, December 2008.
- [5] Th. Besselmann, J. Löfberg, and M. Morari. Explicit MPC for systems with linear parameter-varying state transition matrix. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008.
- [6] Th. Besselmann, J. Löfberg, and M. Morari. Explicit MPC for LPV systems: Stability and optimality. *IEEE Transactions on Automatic Control*, 57(9):2322–2332, 2012.
- [7] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2003.
- [8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [9] P. Brunovský. A classification of linear controllable systems. *Kybernetika*, 6(3):173–188, 1970.
- [10] L. Chisci, P. Falugi, and G. Zappa. Gain-scheduling MPC of nonlinear systems. *International Journal of Robust and Nonlinear Control*, 13(3-4):295–308, 2003.
- [11] K. Fukuda. *Cdd/cdd+ Reference Manual*, December 1997. Available from [www.cs.mcgill.ca/~fukuda/soft/cdd\\_home/cdd.html](http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html).
- [12] T. Gal and J. Nedoma. Multiparametric linear programming. *Management Science*, 18:406–442, 1972.
- [13] W.L. Jin. A method for computing quadratic Brunovsky forms. *Electronic Journal of Linear Algebra, ELA*, 13:40–55, 2005.
- [14] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. Available from <http://control.ee.ethz.ch/~mpt/>.
- [15] J. Löfberg. YALMIP, 2004. Available from <http://users.isy.liu.se/johanl/yalmip/>.
- [16] Y. Lu and Y. Arkun. Quasi-min-max MPC algorithms for LPV systems. *Automatica*, 36(4):527–540, 2000.
- [17] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.
- [18] J. Mikleš and M. Fikar. *Process Modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg, 2007.
- [19] B. Pluymers, J.A. Rossiter, J.A.K. Suykens, and B. de Moor. Interpolation based MPC for LPV systems using polyhedral invariant sets. In *American Control Conference, 2005*, pages 810–815. IEEE, 2005.
- [20] S.V. Rakovic and M. Baric. Parameterized robust control invariant sets for linear systems: theoretical advances and computational remarks. *IEEE Transactions on Automatic Control*, 55(7):1599–1614, 2010.
- [21] M. Steinbuch, R. Van De Molengraft, and A.J. Van Der Voort. Experimental modelling and LPV control of a motion system. In *2003 American Control Conference*, pages 1374–1379, 2003.
- [22] S. Yu, C. Bohm, H. Chen, and F. Allgower. Stabilizing model predictive control for LPV systems subject to constraints with parameter-dependent control law. In *Proceedings of the American Control Conference*, pages 3118–3123, 2009.
- [23] Y. Zhu and Z. Xu. A method of LPV model identification for control. In *Proceedings of the 17th IFAC World Congress*, pages 5018–5023, Seoul, Korea, 2008.