

Low-Complexity Polynomial Approximation of Explicit MPC via Linear Programming

Michal Kvasnica, Johan Löfberg, Martin Herceg, Ľuboš Čirka, and Miroslav Fikar

Abstract—This paper addresses the issue of the practical implementation of Model Predictive Controllers (MPC) to processes with short sampling times. Given an explicit solution to an MPC problem, the main idea is to approximate the optimal control law defined over state space regions by a single polynomial of pre-specified degree which, when applied as a state-feedback, guarantees closed-loop stability, constraint satisfaction, and a bounded performance decay. It is shown how to search for such a polynomial by solving a single linear program.

I. INTRODUCTION

The task of on-line implementation of MPC in the Receding Horizon fashion boils down to repetitively solving a given optimization problem at each sampling instance. Success of a practical real-time implementation of such a control strategy therefore depends on whether enough computational resources are available at the target implementation platform to perform the optimization within the duration of one sampling period. If the sampling time decreases, or if less powerful control platforms are employed, additional care has to be taken to respect the hard real-time constraints.

Nowadays it is a standard practice to address this problem by solving the given MPC optimization problem explicitly [3], i.e. to “pre-calculate” the optimal solution for all feasible initial conditions. The result is an explicit representation of the control law in a form of a look-up table. Complexity of the real-time implementation of such solutions is determined by the amount of memory needed to describe the table and by the amount of CPU time needed to traverse it. Clearly, both of these figures grow proportionally with the table size.

Unfortunately, even in an ‘average’ case the complexity of the look-up table in the number of defining state space regions tends to be very large and above the storage or speed limits of most real-time control devices [3]. Therefore, it is often essential for a real-life implementation to find an appropriate approximation of the controller. Several authors approached this problem by either modifying the original MPC problem, retrieving a suboptimal solution, or by post-processing the computed

optimal controller, cf. e.g. [2], [5], [12]. However, a direct guarantee on the reduction of the complexity, closed-loop stability, or performance decay is mostly neglected.

This paper aims at approximating the given explicit MPC control law $\mu(x)$ by a single multivariate polynomial

$$\tilde{\mu}(x) = \sum_{i=0}^d \sum_{j=1}^{n_x} [\alpha_i]_j x_j^i \quad (1)$$

of pre-specified degree d in such a way that closed-loop stability, feasibility, and bounded performance decay are guaranteed. Here, $\alpha_i \in \mathbb{R}^{n_u \times n_x}$ are the coefficients to be determined, $[\alpha_i]_j$ denotes the j -th column of α_i , x_j^i is the i -th power of the j -th element of vector $x \in \mathbb{R}^{n_x}$, and n_x and n_u denote, respectively, the number of states and control inputs. Once calculated, the polynomial can replace the explicit MPC solution as a feedback controller, without negative impact on stability or constraint satisfaction. The added benefit is that evaluation of the polynomial feedback (1) for a given state measurements x can be done much faster compared to traversing the look-up table of the optimal MPC controller [9]. Memory footprint of the approximate controller is also significantly smaller compared to that of the optimal MPC feedback law.

The approximation is performed in two steps. First, given an explicit representation of the MPC control law $\mu(x)$ and a corresponding PWA Lyapunov function $V(x)$, the set of feedback laws which render $V(x)$ a Control Lyapunov Function is calculated using basic computational geometry tools. It is shown that any control law from this set asymptotically stabilizes the given system while also providing constraint satisfaction for all time. Then, in the second step, we show how to search for the coefficients of the approximation polynomial such that it is always contained in the set of stabilizing feedback laws by solving a single linear program (LP). We have proposed a similar procedure in [9] where the polynomial is searched for by solving a *sum-of-squares* (SOS) optimization. The SOS approach, however, quickly becomes prohibitive even for small-scale problems. As will be illustrated in Section IV, the LP-based procedure presented here outperforms the SOS-based approach in terms of performance by several orders of magnitude. Therefore the results of this paper open up the idea of polynomial approximation to a much wider class of problems. This is stressed in Section V where the discussed procedure is applied to devise a cheap control strategy for a real thermo-optical device.

Johan Löfberg is with the Automatic Control Laboratory, Linköpings universitet, Linköping, Sweden; johanl@isy.liu.se. All other authors are with the Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia; {michal.kvasnica,martin.herceg,lubos.cirka,miroslav.fikar}@stuba.sk

II. PRELIMINARIES

We consider the class of discrete-time, stabilizable Piecewise Affine (PWA) systems of the following form

$$\begin{aligned} x(t+1) &= f_{\text{PWA}}(x(t), u(t)) \\ &= A_d x(t) + B_d u(t) + a_d, \text{ if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{D}_d, \end{aligned} \quad (2)$$

where $t \geq 0$, the domain $\mathcal{D} := \cup_{d=1}^{N_{\mathcal{D}}} \mathcal{D}_d$ of $f_{\text{PWA}}(\cdot, \cdot)$ is a non-empty compact set in $\mathbb{R}^{n_x+n_u}$ with $N_{\mathcal{D}} < \infty$ the number of system dynamics, and $\{\mathcal{D}_d\}_{d=1}^{N_{\mathcal{D}}}$ denotes a polyhedral partition of the domain \mathcal{D} , i.e. the closure of \mathcal{D}_d is $\bar{\mathcal{D}}_d := \{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n_x+n_u} \mid D_d^x x + D_d^u u \leq D_d^0 \}$ and $\text{int}(\mathcal{D}_d) \cap \text{int}(\mathcal{D}_j) = \emptyset$ for all $d \neq j$. Note that linear state and input constraints can be naturally incorporated in the description of \mathcal{D}_d .

Assumption 2.1: The origin in the extended state-input space is an equilibrium point of the PWA system (2), i.e. $\mathbb{0}_{n_x+n_u} \in \mathcal{D}$ and $\mathbb{0}_{n_x} = f_{\text{PWA}}(\mathbb{0}_{n_x}, \mathbb{0}_{n_u})$, where $\mathbb{0}_n := [0 \ 0 \ \dots \ 0]' \in \mathbb{R}^n$. \square

We define for the aforementioned PWA system (2) the *constrained finite time optimal control* (CFTOC) problem

$$J_T^*(x(0)) = \min_{U_T} \ell_T(x(T)) + \sum_{t=0}^{T-1} \ell(x(t), u(t)) \quad (3a)$$

$$\text{s.t. } \begin{cases} x(t+1) = f_{\text{PWA}}(x(t), u(t)) \\ x(T) \in \mathcal{X}^f, \end{cases} \quad (3b)$$

where $\ell(\cdot, \cdot)$ is the *stage cost*, $\ell_T(\cdot)$ the *final penalty function*, U_T is the *optimization variable* defined as the input sequence $U_T := \{u(t)\}_{t=0}^{T-1}$, $T < \infty$ is the *prediction horizon*, and \mathcal{X}^f is a compact *terminal target set* in \mathbb{R}^{n_x} . The CFTOC problem (3) implicitly defines the set of feasible initial states $\mathcal{X}_T \subset \mathbb{R}^{n_x}$ ($x(0) \in \mathcal{X}_T$) and the set of feasible inputs $\mathcal{U}_{T-t} \subset \mathbb{R}^{n_u}$ ($u(t) \in \mathcal{U}_{T-t}$, $t = 0, \dots, T-1$). In the sequel we will consider linear cost functions of the form

$$\ell(x(t), u(t)) := \|Qx(t)\|_p + \|Ru(t)\|_p, \quad (4a)$$

$$\ell_T(x(T)) := \|Px(T)\|_p, \quad (4b)$$

where $\|\cdot\|_p$ with $p \in \{1, \infty\}$ denotes the standard vector 1-/∞-norm.

The goal in this section is to summarize results on explicit (closed-form) expression for $u^*(t) : \mathcal{X}_T \rightarrow \mathcal{U}_T$.

Theorem 2.2 (Solution to CFTOC [3]): The solution to the optimal control problem (3a)–(3b) with a linear performance index (4) is a time-invariant piecewise affine function of the initial state $x(0)$

$$\mu_{\text{RH}}(x(t)) := K_{T,i} x(t) + L_{T,i}, \quad (5)$$

if $x(t) \in \mathcal{P}_i$, with $u^*(t) = \mu_{\text{RH}}(x(t))$, and $\{\mathcal{P}_i\}_{i=1}^{N_{\mathcal{P}}}$ is a polyhedral partition of the set of feasible states $x(t)$, $\mathcal{X}_T = \cup_{i=1}^{N_{\mathcal{P}}} \mathcal{P}_i$, with the closure of \mathcal{P}_i given by $\bar{\mathcal{P}}_i = \{x \in \mathbb{R}^{n_x} \mid P_i^x x \leq P_i^0\}$. \blacksquare

Remark 2.3: The CFTOC problem (3) can be formulated using YALMIP [13] and the closed-form solution

in the form of (5) can be calculated e.g. using the freely available Multi-Parametric Toolbox (MPT) [10].

Assumption 2.4 (Stability, feasibility): Note that in the following it is assumed that the parameters T, Q, R, P , and \mathcal{X}^f are chosen in such a way that (5) is closed-loop stabilizing, feasible for all time [4] and that a polyhedral piecewise affine Lyapunov function of the form

$$V(x) = V_i^x x + V_i^0, \text{ if } x \in \mathcal{P}_i, \quad (6)$$

for the *closed-loop system*

$$f^{\text{CL}}(x(t)) := f_{\text{PWA}}(x(t), \mu_{\text{RH}}(x(t))), \quad (7)$$

$x(t) \in \mathcal{X}_T$, exists and is given. \square

This is not a restricting requirement but rather the aim of most (if not all) control strategies. Furthermore, we remark that if the parameters are chosen according to e.g. [14] one can simply take $V(\cdot)$ equal to the optimal cost $J_T^*(\cdot)$.

In order to present the complete result for the new controller approximation approach, the two underlying core ideas need to be explained. The first idea is based on the inherent freedom of the Lyapunov function (6):

Theorem 2.5 (Asymptotic/exponential stability [11]):

Let \mathcal{X}_T be a bounded positively invariant set in \mathbb{R}^{n_x} for the autonomous (closed-loop) system $x(t+1) = f^{\text{CL}}(x(t))$ with $x(t) \in \mathcal{X}_T$ and let $\underline{\alpha}(\cdot)$, $\bar{\alpha}(\cdot)$, and $\beta(\cdot)$ be K -class functions [17]. If there exists a non-negative function $V : \mathcal{X}_T \rightarrow \mathbb{R}_{\geq 0}$ with $V(\mathbb{0}_{n_x}) = 0$ such that

$$\underline{\alpha}(\|x\|) \leq V(x) \leq \bar{\alpha}(\|x\|), \quad (8a)$$

$$\Delta V(x) := V(f^{\text{CL}}(x)) - V(x) \leq -\beta(\|x\|), \quad (8b)$$

for all $x \in \mathcal{X}_T$, then the following results hold:

- (a) The equilibrium point $\mathbb{0}_{n_x}$ is *asymptotically stable* [17] in the Lyapunov sense in \mathcal{X}_T .
- (b) If $\underline{\alpha}(\|x\|) := \underline{a}\|x\|^\gamma$, $\bar{\alpha}(\|x\|) := \bar{a}\|x\|^\gamma$, and $\beta(\|x\|) := b\|x\|^\gamma$ for some positive constants $\underline{a}, \bar{a}, b, \gamma > 0$ then the equilibrium point $\mathbb{0}_{n_x}$ is *exponentially stable* [17] in the Lyapunov sense in \mathcal{X}_T . \blacksquare

Simply speaking, if all the prerequisites of Theorem 2.5 are fulfilled with a given controller $\mu_{\text{RH}}(\cdot)$, the resulting behavior of the closed-loop system is stabilizing. If, for the given function $V(\cdot)$, $\beta(\cdot)$ is now relaxed, one can (possibly) find a set of controllers that will render the closed-loop system stabilizing and feasible. These sets of controllers are denoted in the following as *stability tubes*. The concept and results of stability tubes – along with their computation – are elaborated in further detail in [4, Ch. 10].

Definition 2.6 (Stability tube): Let $V(\cdot)$ be a Lyapunov function for the general nonlinear, closed-loop system $x(t+1) = f(x(t), u(t))$, with $x(t) \in \mathcal{X}_T$, under the stabilizing control $u(t) = \mu(x(t))$ and constraints $\begin{bmatrix} x \\ u \end{bmatrix} \in \mathcal{D}$ and let the prerequisites of Theorem 2.5 be

fulfilled. Furthermore, let $\beta(\cdot)$ be a K -class function. Then the set

$$\mathcal{S}(V, \beta) := \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n_x \times n_u} \mid f(x, u) \in \mathcal{X}_T, \right. \\ \left. \begin{bmatrix} x \\ u \end{bmatrix} \in \mathcal{D}, V(f(x, u)) - V(x) \leq -\beta(\|x\|) \right\}$$

is called *stability tube*. \square

Theorem 2.7 ([4]): Let the assumptions of Definition 2.6 be fulfilled. Then every control law $u(t) = \tilde{\mu}(x(t))$, $x(t) \in \mathcal{X}_T$, (also any sequence of control samples $u(t)$) fulfilling

$$\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{S}(V, \beta) \quad (9)$$

asymptotically stabilizes the system $x(t+1) = f(x(t), u(t))$, where $x(t) \in \mathcal{X}_T$, to the origin. \blacksquare

Naturally, for general nonlinear systems, the stability tube $\mathcal{S}(V, \beta)$ can basically take any form. Note, however, that for the considered class of PWA systems (2), PWA control laws $u(x) = \mu_{RH}(x)$ of the form (5), and PWA Lyapunov functions of the form (6) with $\beta(\cdot)$ consisting of a sum of weighted vector 1-/∞-norms, the stability tube can be described by a collection of polytopic sets in the state-input space and can be computed with basic polytopic operations:

$$\mathcal{S}(V, \beta) := \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \mid f_{PWA}(x, \mu_{RH}(x)) \in \mathcal{X}_T, \right. \\ \left. V(f_{PWA}(x, \mu_{RH}(x))) - V(x) \leq -\beta(\|x\|) \right\}. \quad (10)$$

In the case considered here, the stability tube can be represented and ‘easily’ be obtained as a collection (or union) of polytopes of the form $\mathcal{S}(V, \beta) := \bigcup_{j=1}^{N_S} \mathcal{S}_j$, where the closure of \mathcal{S}_j is $\bar{\mathcal{S}}_j := \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n_x \times n_u} \mid \mathcal{S}_j^{xu} \begin{bmatrix} x \\ u \end{bmatrix} \leq \mathcal{S}_j^0 \right\}$.

Without going into details, by construction, we have the following properties: (a) for some index set $\mathcal{I}_i \subseteq \{1, \dots, N_S\}$, the union $\bigcup_{j \in \mathcal{I}_i} \mathcal{S}_j$ is defined over the controller region \mathcal{P}_i , and (b), $\sum_{i=1}^{N_P} |\mathcal{I}_i| = N_S$. This means that each \mathcal{S}_j is defined over a single region \mathcal{P}_i , i.e. if for some i_1 and j we have $\text{proj}_x(\mathcal{S}_j) \subseteq \mathcal{P}_{i_1}$ then there does not exist a $i_2 \neq i_1$ with $\text{proj}_x(\mathcal{S}_j) \subseteq \mathcal{P}_{i_2}$. We remark that simulations seem to indicate that most often $\mathcal{I}_i = 1$ for all i , i.e. only one \mathcal{S}_j is defined over \mathcal{P}_i .

III. MAIN RESULTS

Let us state the main aim of the paper:

Problem 3.1: Given $\mu_{RH}(x)$ as in (5) and $V(x)$ of the form (6) as an optimal closed-form solution to the CFTOC problem (3) for a PWA system (2) with $p = 1$ or $p = \infty$ in (4), find coefficients $\alpha_0, \dots, \alpha_d$ of the polynomial state-feedback law (1) of fixed degree d which approximates $\mu_{RH}(x)$ in such a way that closed-loop stability, constraint satisfaction, and a bounded performance decay are guaranteed.

Assumption 3.2: For the pair $\mu_{RH}(x)$, $V(x)$ there exists a stability tube $\mathcal{S}(V, \beta) = \bigcup \mathcal{S}_i(V, \beta)$ of the form (10) with \mathcal{S}_i defined over the i -th regions \mathcal{P}_i being convex (i.e. $|\mathcal{I}_i| = 1$), and the union $\bigcup \mathcal{S}_i(V, \beta)$ being connected.

Existence of $\mathcal{S}(V, \beta)$ hints at existence of control laws other than $\mu_{RH}(x)$ which would yield the same closed-loop properties (stability and constraint satisfaction). Connectivity is implied by the assumption that a single polynomial covers the whole space of interest and convexity is assumed in order to obtain a unique solution.

Theorem 2.7 provides a sufficient condition for existence of $\tilde{\mu}(x)$ which solves Problem 3.1:

Lemma 3.3: Let a stability tube $\mathcal{S}(V, \beta)$ satisfying Assumption 3.2 be given and denote by $p_i(\alpha, x)$ a set of polynomials

$$p_i(\alpha, x) := \mathcal{S}_i^0 - \mathcal{S}_i^{xu} \begin{bmatrix} x \\ \tilde{\mu}(x) \end{bmatrix}. \quad (11)$$

Then $\tilde{\mu}(x)$ as in (1) is a solution to Problem 3.1 if

$$p_i(\alpha, x) \geq 0, \quad \forall x \in \mathcal{P}_i, \quad \forall i \in [1, \dots, N_P]. \quad (12)$$

Proof: By assuming convexity of $\mathcal{S}_i(\cdot)$ we have $\mathcal{S}_i(\cdot) = \left\{ \begin{bmatrix} x \\ \tilde{\mu}(x) \end{bmatrix} \mid \mathcal{S}_i^{xu} \begin{bmatrix} x \\ \tilde{\mu}(x) \end{bmatrix} \leq \mathcal{S}_i^0 \right\}$. Hence (12) is equivalent to (9) with $p_i(\alpha, x)$ as in (11). From Theorem 2.7 it follows that any control law, i.e. also $u = \tilde{\mu}(x)$, satisfying $\begin{bmatrix} x \\ u \end{bmatrix} \in \bigcup \mathcal{S}_i(V, \beta)$ will provide closed-loop stability, constraint satisfaction, and a guaranteed worst-case performance decay of $\beta(\|x\|)$. \blacksquare

Lemma 3.3 suggests that finding $\tilde{\mu}(x)$ of the form (1) as a solution to Problem 3.1 can be cast as finding coefficients $\alpha_0, \dots, \alpha_d$ such that polynomials $p_i(\alpha, x)$ are non-negative for all points $x \in \mathcal{P}_i$, $\forall i \in [1, \dots, N_P]$. In [9] we have shown how to search, conservatively, for the coefficients of $\tilde{\mu}(x)$ satisfying (12) by formulating a sum-of-squares (SOS) problem, which can be solved using semidefinite programming (SDP) techniques. However, as will be illustrated later, from practical point of view complexity of the underlying SDP problem is often prohibitive even for small values of N_P .

In this work we suggest an alternative procedure of finding the coefficients $\alpha_0, \dots, \alpha_d$ of $\tilde{\mu}(x)$ which is, from a computational point of view, superior to the SOS-based approach of [9]. The proposed approach is based on the following theorem, due to Pólya [6]:

Theorem 3.4 (Pólya’s theorem): If a homogeneous polynomial $p_i(\alpha, x)$ is positive $\forall x \in \mathcal{P}_i$ with \mathcal{P}_i being a simplex, all the coefficients of $p_i^M(\alpha, x) = p_i(\alpha, x) \cdot (\sum_{j=1}^{n_x} x_j)^M$ are positive for a sufficiently large Pólya degree M .

Remark 3.5: Search for α such that $p_i^M(\alpha, x) \geq 0, \forall x \in \mathcal{P}_i$ can be performed by using the more obvious reverse of Pólya’s theorem, i.e. that positive coefficients of the extended polynomial imply positivity over the whole simplex.

Remark 3.6: The advantage of Theorem 3.4 over other conservative techniques for ensuring positivity of polynomials (such as the SOS formulation of [9]) stems from the fact that given a symbolic representation of $p_i^M(\alpha, x)$, the coefficients α can be found by solving a linear program (LP). To see this, observe that α enters (11) in a linear fashion and that all constraints (12) are linear in α .

Notice, however, that Theorem 3.4 is not directly applicable to find α from (12) as \mathcal{P}_i are not simplices, in general. To overcome this limitation, we observe that, by Theorem 2.2, we have $\mathcal{P}_i = \{x \mid P_i^x x \leq P_i^0\}$, which is a polytope described by an intersection of finitely many half-spaces. Given $\mathcal{V}_i = \text{vertices}(\mathcal{P}_i)$ being a set of extremal vertices of \mathcal{P}_i , the i -th region can be equivalently expressed as a convex combination of \mathcal{V}_i :

$$\mathcal{P}_i = \left\{ x \mid x = \sum_{j=1}^{|\mathcal{V}_i|} \lambda_j [\mathcal{V}_i]_j, \forall \lambda \in \Lambda_i \right\}, \quad (13)$$

$$\Lambda_i = \left\{ \lambda \mid 0 \leq \lambda_j \leq 1, \sum_{j=1}^{|\mathcal{V}_i|} \lambda_j = 1 \right\}, \quad (14)$$

where $|\mathcal{V}_i|$ stands for the number of extremal points of the i -th region, $[\mathcal{V}_i]_j$ denotes the j -th vertex of \mathcal{P}_i , and $\lambda = [\lambda_1, \dots, \lambda_{|\mathcal{V}_i|}]$. By substituting for $x = \sum_j \lambda_j [\mathcal{V}_i]_j$ into (12) we get

$$p_i(\alpha, \lambda) \geq 0, \quad \forall \lambda \in \Lambda_i, \forall i \in [1, \dots, N_{\mathcal{P}}]. \quad (15)$$

Notice that Λ_i in (15) are now simplices and Theorem 3.4 can therefore be applied to find α such that $p_i(\alpha, \lambda)$ is non-negative $\forall \lambda \in \Lambda_i$.

We can now state the main result of the paper, which is Theorem 3.7 and Algorithm 1 for calculating values of the coefficients $\alpha_0, \dots, \alpha_d$ of the polynomial feedback law $\tilde{\mu}(x)$ which is an admissible solution to Problem 3.1.

Algorithm 1 Polynomial approximation

INPUT: PWA system (2), parameters $T, Q, R, P, \mathcal{X}^f$ of the CFTOC problem (3), desired maximal performance decay $\beta(\|x\|)$, degree of the approximation polynomial d , Pólya degree M .

OUTPUT: Coefficients $\alpha_0, \dots, \alpha_d$ of the polynomial feedback (1) which, when applied as a state-feedback, asymptotically stabilizes the given PWA system.

- 1: Obtain a closed-form solution $\mu_{\text{RH}}(x)$, \mathcal{P}_i , $V(x)$ to the CFTOC problem (3) according to Theorem 2.2.
- 2: Calculate the stability tube $\mathcal{S}(V, \beta)$ per (10).
- 3: Calculate extremal vertices \mathcal{V}_i of all regions \mathcal{P}_i .
- 4: Formulate polynomials $p_i(\alpha, \lambda)$ per (15).
- 5: Homogenize $p_i(\alpha, \lambda)$ by multiplying single monomials by $(\sum_{j=1}^{|\mathcal{V}_i|} \lambda_j)$ until all monomials have the same degree.
- 6: Compute, symbolically, coefficients c_i^M of the Pólya's polynomial $p_i^M(\alpha, \lambda) = p_i(\alpha, \lambda) \cdot (\sum_{j=1}^{|\mathcal{V}_i|} \lambda_j)^M$.
- 7: Search for α by solving the following linear program:

$$\text{find} \quad \alpha_0, \dots, \alpha_d, \quad (16)$$

$$\text{s.t.} \quad c_i^M \geq 0, \quad \forall i \in [1, \dots, N_{\mathcal{P}}]. \quad (17)$$

8: **return** $\alpha_0, \dots, \alpha_d$.

Theorem 3.7: Let the input arguments of Algorithm 1 satisfy the conditions of Assumptions 2.1, 2.4, and 3.2. Then the polynomial feedback $\tilde{\mu}(x)$ of the form (1) calculated by Algorithm 1 is a solution to Problem 3.1.

Proof: Directly by Lemma 3.3 and Theorem 3.4. ■

Remark 3.8: All conditions of Assumption 2.4 will be satisfied (and hence $V(x) = J_T^*(x)$) for $T = \infty$ and P, \mathcal{X}^f calculated as shown in [1].

Remark 3.9: Computation in Steps 1 and 3 of Algorithm 1 can be carried out using Multi-Parametric Toolbox. The code for calculating $\mathcal{S}(V, \beta)$ can be obtained upon mail request from the authors. Steps 4–7 can be solved using YALMIP [13], which takes care of all symbolic and non-symbolic calculations.

Remark 3.10: Algorithm 1 is a non-iterative procedure and therefore it always terminates.

Remark 3.11: It is worth noting that Algorithm 1 always found a feasible approximation for $d = 5$ for all cases investigated in Section IV. However, since Theorem 3.4 is just a sufficient condition, it cannot be ruled out that Step 7 could fail. In such a case it is advised to repeat Steps 4–7 with an increased degree of the approximation polynomial.

Remark 3.12: Instead of a pure feasibility objective in (16), an alternative is to look for α which minimize the point-wise distance $\|\mu_{\text{RH}}(x_j) - \tilde{\mu}(x_j)\|_1$ with $x_j = [\mathcal{V}_i]_j$, $\forall j = [1, \dots, |\mathcal{V}_i|]$, $\forall i = [1, \dots, N_{\mathcal{P}}]$. Another approach is to try to aim for low-order polynomials by minimizing coefficients for higher-order terms. Alternatively, one can even aim for low-complexity controller by minimizing the number of non-zero coefficients, which would lead to a mixed-integer LP problem.

Example 3.13: To illustrate the results of Theorem 3.7, consider the following 1D PWA system [9]:

$$x(t+1) = \begin{cases} 4/5 x(t) + 2u(t) & \text{if } x > 0, \\ -6/5 x(t) + u(t) & \text{if } x \leq 0, \end{cases} \quad (18)$$

with $u(t) \in [-1, 1]$ and $x(t) \in [-4, 4]$. The CFTOC problem (3) was solved with $p = 1$, $Q = 1$, $R = 1$, $P = 0$, $T = \infty$ and the corresponding stability tube $\mathcal{S}(\cdot)$ was calculated for $\beta(\|x\|) = b\|x\|^\gamma$ with $b = 1 \cdot 10^{-6}$ and $\gamma = 1$. The closed-form solution consisted of 7 regions and the stability tube satisfied Assumption 3.2. The sets $\mathcal{S}(\cdot)$ are depicted in gray in Figure 1 along with the optimal feedback law $\mu_{\text{RH}}(x)$. Coefficients of three approximation polynomials with $d = 3, 5, 7$ have been subsequently calculated using Theorem 3.4 with $M = 1$ and are also depicted in Figure 1. The distance-minimization criterion suggested in Remark 3.12 was used when solving the LP in Step 7 of Algorithm 1.

IV. COMPLEXITY COMPARISON

The purpose of this section is to compare computational complexity of the LP-based procedure of Algorithm 1 against the SDP-based approach of [9]. As outlined in the previous section, both methods aim at showing positivity of (12) and therefore are equivalent from

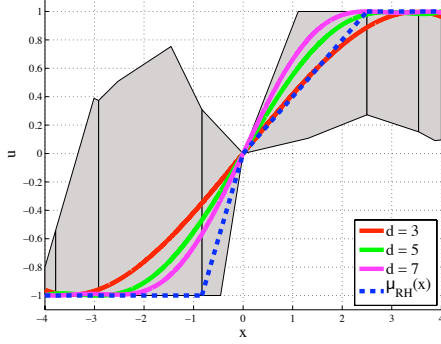


Fig. 1. Stability tubes $\mathcal{S}(\cdot)$ (gray sets), optimal control law $\mu_{RH}(x)$ (blue dashed line), and stabilizing polynomial approximations of different degrees.

a functional point of view, but they differ performance-wise.

To evaluate performance of both approaches, we have considered random PWA systems with 2 dynamics, 1 input, and a varying number of n_x states. All systems satisfied Assumption 2.1. For these systems we have subsequently calculated the closed-form solution to the CFTOC problem (3) with $Q = I$, $R = 1$, and $T = \infty$, which guarantees that the solution satisfies Assumption 2.4 [1]. Naturally, for different systems, different number of regions $N_{\mathcal{P}}$ was obtained. The stability tubes $\mathcal{S}(\cdot)$ have been calculated for $b = 1 \cdot 10^{-6}$ and $\gamma = 1$ and they always satisfied conditions of Assumption 3.2. The sets $\mathcal{S}(\cdot)$ were then used to find the approximation polynomial $\tilde{\mu}(x)$ of degree $d = 5$. The Pólya degree of $M = 1$ was sufficient to find a polynomial approximation in all considered cases using the LP-based procedure of Algorithm 1. The results summarizing the runtimes of the approximations are reported in Table I. All computations have been carried out on a Core2Duo CPU with 2.5GHz and 2GB of RAM running Matlab R2007b on Linux. The SDPT3 solver [15] was employed to solve the SDP formulation, while CPLEX 11.2 [8] was used as an LP solver. As can be observed, the LP-based procedure for finding the approximation polynomial clearly outperforms the SOS-based approach in terms of runtime.

It should be noted that although the calculation of extremal vertices in Step 3 is considered a hard problem in general, the physical runtime of this step is usually minute compared to the runtime of the LP solver in Step 7. In fact, calculation of the extremal vertices for all cases reported in Table I did not exceed 0.1 seconds using the MPT toolbox.

V. EXPERIMENTAL RESULTS

To further emphasize practical applicability of the proposed approximation scheme, we have applied it to design a polynomial controller for a thermo-optical device [7]. The plant consists of a lightbulb, a temperature sensor and a sensor measuring light intensity. The device is

TABLE I
COMPARISON OF RUNTIMES FOR STEP 7 OF ALGORITHM 1 (*Runtime LP*) AND THE SOS-BASED PROCEDURE OF [9] (*Runtime SOS*). † MEANS THAT THE PROCEDURE FAILED BECAUSE IT RAN OUT OF MEMORY.

n_x	$N_{\mathcal{P}}$	Runtime LP	Runtime SOS
2	34	0.1 secs	285 secs
2	50	0.2 secs	658 secs
2	60	0.2 secs	1390 secs
2	78	1.9 secs	3436 secs
2	92	4.9 secs	5475 secs
2	146	12.7 secs	†
2	170	22.8 secs	†
3	66	2.9 secs	†
3	122	4.6 secs	†

TABLE II
STATE-UPDATE MATRICES OF THE PWA MODEL (2).

A_1	B_1	a_1	0.813	0.936	-0.862	2.124
A_2	B_2	a_2	-0.239	0	0.455	-2.269
A_3	B_3	a_3	0.813	0.348	-0.522	0.391
			-0.644	0	0.247	-1.125
			0.813	0.246	-0.462	0
			-0.912	0	0.107	0

connected to a PC via an USB interface. From a physical point the plant is characterized by heat exchange from the lightbulb to ambient air with slow dynamics and a light emission with fast dynamics. Presence of physical constraints and a short sampling time of $T_s = 0.05$ seconds of the closed-loop system make control design challenging. The control task is to manipulate voltage to the bulb in such a way that a pre-scribed light intensity setpoint is reached without violating temperature limits.

The plant model was obtained by experimental identification techniques around multiple operating points using the IDTOOL toolbox [18]. Numerical values of the corresponding PWA model can be found in Table II. Moreover, physical constraints of the plant translate into state bounds $-2.57 \leq x_1(t) \leq 7.43$, $-6.78 \leq x_2(t) \leq 3.22$, and the input bound $-7 \leq u(t) \leq 3$.

We have then applied Algorithm 1 to obtain a polynomial control law (1). In Step 1 the CFTOC problem (3) was solved using Multi-Parametric Toolbox (MPT) with $T = \infty$, $Q = 2I$, and $R = 1$. The resulting closed-form solution consisted of 61 regions in the 2-dimensional state-space. Subsequently, the stability tube $\mathcal{S}(V, \beta) = \bigcup \mathcal{S}_j(V, \beta)$ was calculated for $b = 1 \cdot 10^{-6}$ and $\gamma = 1$ using the polytopic library of the MPT toolbox. We remark that $\mathcal{S}(\cdot)$ satisfied all conditions of Assumption 3.2. Finally, the coefficients of the polynomial control law (1) have been calculated by Algorithm 1. For $d = 3$ we have obtained the feedback $\tilde{\mu}(x) = 0.0006x_1^3 + 0.0041x_2^3 - 0.0627x_1^2 - 0.0031x_2^2 + 0.8272x_1 - 0.0366x_2$. Graphical representation of $\tilde{\mu}(x) \in \mathcal{S}(V, \beta)$ is shown in Figure 2. Interesting to notice is that the memory footprint of $\tilde{\mu}(x)$ is just 24 bytes (each of the 6 coefficients requires 4 bytes when stored as floating-point numbers) and the

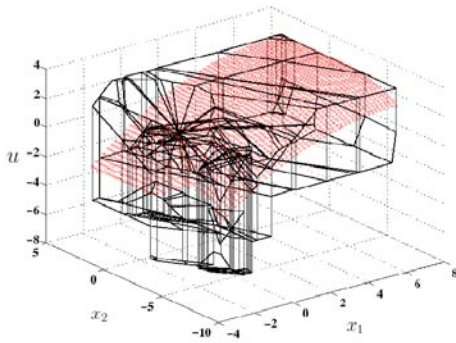


Fig. 2. Visualization of $\tilde{\mu}(x)$ inside of the stability tube.

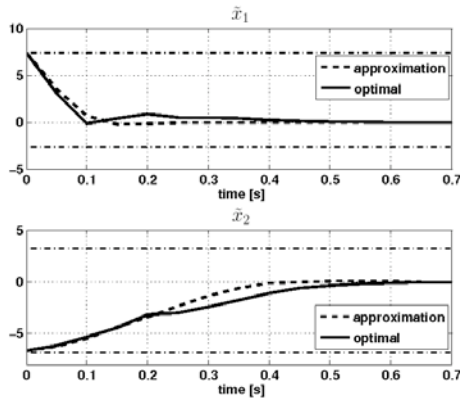


Fig. 3. Closed-loop evolution of plant's state under optimal control (solid lines) and under the approximate polynomial feedback (dashed lines).

polynomial can be evaluated for a given value of x in 14 FLOPS using Horner's scheme. Optimal closed-form solution $\mu_{RH}(x)$, defined over 61 regions, would require 2584 bytes of memory and 35 FLOPS to evaluate using the binary search tree of [16].

The optimal feedback $\mu_{RH}(x)$ as well as its polynomial approximation $\tilde{\mu}(x)$ of degree 3 have been implemented in real time and experimental data have been obtained. The case in Figure 3 shows closed-loop transition of system states starting from $x(0) = [7.43, -6.78]$ (lightbulb fully dimmed) to the origin. As can be observed, the approximate polynomial controller performs well, and provides guarantees of stability and constraint satisfaction.

VI. CONCLUSIONS

In this paper we have presented a novel way of deriving simple stabilizing feedback laws for the class of constrained PWA systems. Stability and feasibility of the presented scheme is guaranteed by employing the concept of stability tubes, which can be viewed as a parametrization of stabilizing feedback laws. Such sets are subsequently approximated by a polynomial of a fixed degree. The advantage of such approach, compared to other schemes, is that the on-line implementation of

such controller can be performed faster and requires less memory storage.

VII. ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/0537/10. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07.

REFERENCES

- [1] M. Baotić, F. J. Christophersen, and M. Morari. Constrained Optimal Control of Hybrid Systems With a Linear Performance Index. *IEEE Trans. on Automatic Control*, 51(12):1903–1919, December 2006.
- [2] A. Bemporad and C. Filippi. Suboptimal explicit RHC via approximate multiparametric quadratic programming. *Journal of Optimization Theory and Applications*, 117(1):9–38, April 2003.
- [3] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2003.
- [4] F. J. Christophersen. *Optimal Control of Constrained Piecewise Affine Systems*, volume 359 of *Lecture Notes in Control and Information Sciences*. Springer Verlag, 2007.
- [5] T. Geyer, F. D. Torrisi, and M. Morari. Optimal Complexity Reduction of Piecewise Affine Models Based on Hyperplane Arrangements. In *Proc. on the American Control Conference*, pages 1190–1195, Boston, Massachusetts, USA, June 2004.
- [6] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, second edition, 1952.
- [7] M. Huba, P. Kurčík, and M. Kamenský. *Thermo-optical device uDAQ28/LT*. STU Bratislava, Illkovičova 3, Bratislava, 2006. In Slovak.
- [8] ILOG, Inc. *CPLEX User Manual*. Gentilly Cedex, France. <http://www.ilog.fr/products/cplex/>.
- [9] M. Kvasnica, F. J. Christophersen, M. Herceg, and M. Fikar. Polynomial approximation of closed-form MPC for piecewise affine systems. In *Proceedings of the 17th IFAC World Congress*, pages 3877–3882, Seoul, Korea, July 6–11 2008.
- [10] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi-Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, pages 448–462, March 2004.
- [11] M. Lazar, D. Muñoz de la Peña, W.P.M.H. Heemels, and T. Alamo. On input-to-state stability of min-max nonlinear model predictive control. *Systems & Control Letters*, 57:39–48, 2008.
- [12] M. Lazar, W. P. M. H. Heemels, B. J. P. Roset, H. Nijmeijer, and P. P. J. van den Bosch. Input-to-state stabilizing sub-optimal nmppc with an application to dc-dc converters. *International Journal of Robust and Nonlinear Control*, 2007.
- [13] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.
- [15] K.C. Toh, M.J. Todd, and R.H. Tütüncü. SDPT3 – a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11/12:545–581, 1999.
- [16] P. Tøndel, T. A. Johansen, and A. Bemporad. Evaluation of Piecewise Affine Control via Binary Search Tree. *Automatica*, 39(5):945–950, May 2003.
- [17] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, 2nd edition, 1993.
- [18] Ľ. Čírka, M. Fikar, and P. Petruš. IDTOOL 4.0 - A Dynamical System Identification Toolbox for MATLAB/Simulink. In *14th Annual Conference Proceedings: Technical Computing Prague 2006*, pages 29–29, October 2006.