Ultra-Fast Stabilizing Model Predictive Control via Canonical Piecewise Affine Approximations

Alberto Bemporad, *Fellow, IEEE*, Alberto Oliveri, Tomaso Poggi, and Marco Storace, *Member, IEEE*

***Abstract*—This paper investigates the use of canonical piecewise affine (PWA) functions for approximation and fast implementation of linear MPC controllers. The control law is approximated in an optimal way over a regular simplicial partition of a given set of states of interest. The stability properties of the resulting closed-loop system are analyzed by constructing a suitable PWA Lyapunov function. The main advantage of the proposed approach to the implementation of MPC controllers is that the resulting stabilizing approximate MPC controller can be implemented on chip with sampling times in the order of tens of nanoseconds.**

***Index Terms*—Hardware synthesis, model predictive control (MPC), piecewise affine (PWA) approximations.**

I. Introduction

MODEL predictive control (MPC) is an increasingly popular technique in industry for feedback control of multivariable systems subject to constraints on manipulated and controlled variables [1]. In MPC, the desired performance is specified by selecting the value of a set of weights on states and inputs, as in classical linear quadratic regulation, and complemented with constraint specifications on system variables, such as input saturations and desired ranges for state and output variables. Performance is repeatedly optimized under constraints at each sampling time of the controller, by solving a finite-horizon open-loop optimal control problem. This is based on a given dynamical model of the process under control by taking the current (measured or estimated) state as the initial state of the problem. The first sample of the resulting optimal sequence of future control moves is applied to the process, and at the next time step a new optimal control problem is solved again, starting from the new state vector. For this reason MPC is also referred to as "receding-horizon" control.

To overcome the aforementioned computational issues, *explicit* MPC techniques were developed during the last ten years to preprocess the MPC control law off line and convert it into a piecewise affine (PWA) law. In this way, online operations reduce to the evaluation of a lookup table of linear control gains. We refer the reader to [2] for a recent survey on explicit MPC.

Although successfully applied in several practical applications, especially to automotive systems [3] and power converters [4], explicit MPC tends to generate a large set of controller gains. The number of gains depends roughly exponentially on the number of constraints included in the MPC optimization problem. Explicit MPC is usually applied in fast-sampling problems with sampling time in the order of 1–50 ms and of relatively small size (say 1–2 manipulated inputs, 5–10 parameters). The main reason for the excessive number of regions in the explicit MPC law is usually due to the desire of the designer of solving the multiparametric programming problem *exactly*. On the other hand, trying to solve exactly an optimization problem in which the cost function is usually the result of a rough trial and error weight-tuning procedure is often an unjustified approach. To simplify the complexity of explicit MPC controllers, *approximate* explicit MPC techniques were addressed recently [2], [5]–[11]. By sacrificing the optimality of the control action with respect to the selected performance index to a tolerable level, approximate explicit MPC simplifies the control law, so that sampling frequencies can be pushed up considerably. For example, in [12] the authors showed that explicit MPC solutions can be implemented in an application specific integrated circuit (ASIC) with about 20 000 gates, leading to computation times in the order of 1 μs. Field-programmable gate array (FPGA) implementations of explicit MPC with sampling frequencies up to 2.5 MHz were reported recently in [13].

An alternative route to synthesize approximate explicit MPC controllers is to treat the MPC control law as a generic nonlinear function and use general purpose offline *function approximation* techniques. Ideas in this direction were pursued in [14] using artificial neural networks and in [15] using set-membership identification. One of the main drawbacks of function approximation approaches is the difficulty in proving the stabilization properties of the synthesized controller.

In this paper, we adopt a special class of basis functions, the *canonical piecewise-affine (PWA) functions* described in [16]–[18], to approximate a given linear MPC controller and impose constraints in the approximation procedure so as to be able to analyze closed-loop stability properties using PWA Lyapunov functions. The choice of canonical PWA functions has both a theoretical and a technological motivation. On the theoretical side, the closed-loop system under approximate explicit MPC belongs to the class of discrete-time PWA dynamical systems, and here we propose a procedure based on the synthesis of PWA Lyapunov functions and linear programming

Manuscript received May 28, 2010; revised November 16, 2010, March 11, 2011; accepted April 02, 2011. Date of publication April 07, 2011; date of current version December 07, 2011. This work was supported in part by the European Commission through project MOBY-DIC "Model-based synthesis of digital electronic circuits for embedded control" (FP7-INFSO-ICT-248858), http://www.mobydic-project.eu. Recommended by Associate Editor H. Ito

A. Bemporad is with the IMT (Institutions, Market, Technologies) Institute for Advanced Studies, Lucca, 55100 Lucca, Italy (e-mail: alberto.bemporad@imtlucca.it).

A. Oliveri, T. Poggi, and M. Storace are with the Biophysical and Electronic Engineering Department, University of Genoa, I-16145 Genova, Italy (e-mail: marco.storace@unige.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2011.2141410

0018-9286/$26.00 © 2011 IEEE

to analyze its stability properties. On the technological side, PWA functions expressed as linear combinations of canonical PWA functions have a direct circuit implementation on chip [19], [20], which allows an extremely fast computation of the control law, as we show in this paper.

Summing up, the approximated controller can be obtained 1) by calculating the exact explicit MPC and 2) by approximating the exact solution using canonical PWA functions. Even if this approach brings to a fast implementation of explicit MPC, its main drawback is the memory occupation: indeed, the number of coefficients needed to represent the approximated solution grows exponentially with the number of dimensions of the state space. Nevertheless, the number of coefficients can be easily tuned by the control designer in order to meet a tradeoff between controller complexity and performance. We point out that the calculation of the exact explicit MPC is not required if only input constraints are considered.

This paper is organized as follows. In Section II, we formulate the linear MPC law to be approximated using the canonical PWA functions, described in Section III. Section IV describes the procedure for approximating the MPC law, whose closed-loop stability properties are analyzed in Section V. Simulation and circuit implementation results are reported in Section VI.

A. Notation

All vectors are intended as column vectors. The symbol $'$ denotes transposition, I the identity matrix, $\mathbb{1}$ a vector of all ones, $\leq$ *et similia* denote component-wise inequalities, and $\succeq$ (*et similia*) positive semi-definiteness. Given a matrix H (a vector K), $H^{(i)}$ denotes the ith row of H and H_{ij} denotes the (i,j)th entry of H (K_i the ith element of K). Given a set $\Omega \subset \mathbb{R}^n$, $\mathring{\Omega}$ denotes its interior and $\partial\Omega$ its boundary. Given a box $S \subset \mathbb{R}^n$, $L^\infty[S]$ is the space of bounded functions defined over S, $L^2[S]$ the space of Lebesgue square integrable functions ($L^\infty[S] \subset L^2[S]$), and $\mathrm{PWA}[S]$ the space of piecewise affine functions defined over a polyhedral partition of S ($\mathrm{PWA}[S] \subset L^\infty[S]$). $C^0[S]$ is the space of continuous functions over S. We denote by u^* the exact (optimal) MPC control signal and by $\hat{u}$ its suboptimal PWA approximation. The operators $\|\cdot\|$ and $\|\cdot\|_2$ denote the Euclidean norm, $\|\cdot\|_1$ the 1-norm, $\|\cdot\|_\infty$ the infinity norm. Given a set $\Omega \subseteq \mathbb{R}^n$ and a scalar λ, $\lambda\Omega$ denotes the set $\{y \in \mathbb{R}^n : y = \lambda x, x \in \Omega\}$. Given a finite set of vectors $x^1, x^2, \ldots, x^m \in \mathbb{R}^n$, $\mathrm{conv}(x^1, \ldots, x^m)$ denotes their convex hull $\{x \in \mathbb{R}^n : x = \sum_{i=1}^m \alpha_i x^i, \alpha_i \geq 0, \sum_{i=1}^m \alpha_i = 1\}$.

II. Model Predictive Controller

Consider a MPC algorithm based 1) on the linear discrete-time prediction model

$$x(t+1) = Ax(t) + Bu(t) \tag{1}$$

of the open-loop process, in which $x(t) \in \mathbb{R}^n$ is the state vector at sampling time t and $u(t) \in \mathbb{R}^m$ is the vector of manipulated variables, and 2) on the solution of the finite-time optimal control problem

$$\min_U \quad x_N' P x_N + \left(\sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k\right) + \rho\epsilon^2 \tag{2a}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1,\ x_0 = x(t) \tag{2b}$$

$$u_k = Kx_k,\ k = N_u, \ldots, N-1 \tag{2c}$$

$$E_u u_k \leq G_u,\ k = 0, \ldots, N_u - 1 \tag{2d}$$

$$E_u u_k \leq G_u + V_u\epsilon,\ k = N_u, \ldots, N-1 \tag{2e}$$

$$E_x u_k + F_x x_k \leq G_x + V_x\epsilon,\ k = 0, \ldots, N-1 \tag{2f}$$

$$\epsilon \geq 0 \tag{2g}$$

where N is the prediction horizon, N_u is the control horizon, $U \triangleq [u_0' \ \ldots \ u_{N_u-1}' \ \epsilon]' \in \mathbb{R}^{mN_u+1}$ is the vector of variables to be optimized, $Q = Q' \succeq 0$, $R = R' \succ 0$, $P = P' \succeq 0$ are weight matrices of appropriate dimensions defining the performance index, ϵ is a slack variable relaxing the constraints, and $\rho > 0$ is a (large) weight penalizing constraint violations. In (2d)–(2f), E_u, G_u, V_u and E_x, F_x, G_x, V_x are matrices of appropriate dimensions defining constraints on input variables, and on input and state variables, respectively. We also assume that $G_u > 0$, i.e., that the constraint set of $\mathbb{R}^m$ defined by (2d) contains $u = 0$ in its interior, and similarly that $G_x > 0$. A typical instance of (2d) are saturation constraints $E_u = [I \ -I]'$, $G_u = [u_{\max}' \ -u_{\min}']'$, $u_{\min} < 0 < u_{\max}$. In (2f), vector $V_x > 0$ defines the degree of softening of the mixed input/state constraints. Similarly, constraints (2e) are softened through $V_u > 0$.[1] In (2c), K is a terminal gain defining the remaining control moves after the expiration of the control horizon N_u; for instance $K = 0$, or K is the linear quadratic regulator gain associated with matrices Q and R, and P is the corresponding Riccati matrix.

By substituting $x_k = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$, (2) can be recast as the quadratic programming (QP) problem

$$U^*(x(t)) \triangleq \arg\min_U \ \frac{1}{2}U'HU + x'(t)F'U + \frac{1}{2}x'(t)Yx(t) \tag{3a}$$

$$\text{s.t.} \quad GU \leq W + Dx(t) \tag{3b}$$

where $U^*(x(t)) = [u_0'^*(x(t)) \ \ldots \ u_{N_u-1}'^*(x(t)) \ \epsilon^*(x(t))]'$ is the optimal solution, $H = H' \succ 0$ and F, Y, G, W, D are matrices of appropriate dimensions [21]–[23]. Note that Y is not needed to compute $U^*(x(t))$, it only affects the optimal value of (3a).

The MPC control law is

$$u^*(x) = [I \ 0 \ \ldots \ 0]U^*(x) \tag{4}$$

corresponding to solving the QP problem (3) at each time t, applying the first move $u(t) = u_0^*(x(t))$ to the process, discarding the remaining optimal moves, and repeating the procedure again at time $t+1$ for the next state $x(t+1)$. A few extensions of the MPC setup (2) are reported in [24].

One of the drawbacks of the MPC law (4) is the need to solve the QP problem (3) online, which has traditionally labeled MPC as a technology for slow processes.

To get rid of online QP, an alternative approach to evaluate the MPC law (4) was proposed in [21]. Rather than solving the QP problem (3) on line for the current vector $x(t)$, the idea is to solve (3) *off line* for all vectors x within a given range and make the dependence of u on x *explicit*. The key idea is to treat (3) as

[1]To guarantee that problem (2) is feasible for all $x(t) \in \mathbb{R}^n$, we avoid that V_x, V_u have zero components, which would define hard constraints.

a *multiparametric* quadratic programming problem, where $x(t)$ is the vector of parameters. It turns out that $u^*(x) : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a piecewise affine and continuous function, and consequently the MPC controller defined by (4) can be represented explicitly as

$$u^*(x) = \begin{cases} F_0 x & \text{if } H_0 x \leq K_0 \\ F_1 x + g_1 & \text{if } H_1 x \leq K_1 \\ \vdots & \quad \vdots \\ F_{n_r-1} x + g_{n_r-1} & \text{if } H_{n_r-1} x \leq K_{n_r-1} \end{cases} \tag{5}$$

where $F_i \in \mathbb{R}^{m\times n}$, $g_i \in \mathbb{R}^m$; H_i and K_i are matrices and vectors, respectively, of appropriate dimensions; n_r is the number of polyhedral regions $\mathcal{X}_i = \{x : H_i x \leq K_i\}$, $i = 0, \ldots, n_r - 1$ defining the domain partition. $\mathcal{X}_0$ is the region corresponding to the unconstrained solution $F_0 = -[I\ 0\ \ldots\ 0]H^{-1}F$ of (3), where $H_0 = GF_0 - D$ and $K_0 = W$. Note that $G_x, G_u > 0$ imply that $0 \in \mathring{\mathcal{X}}_0$.

Once in the form (5), the evaluation of the MPC controller (4) can be carried out by a very simple piece of control code. As mentioned in the introduction, experiments on the implementation of explicit MPC on FPGA and application specific integrated circuits (ASIC) with sampling times around 1 μs have been recently reported in [12], [25]. Although they are both based on the binary search tree proposed in [26], the design strategies proposed in [12] and [25] are different. In [25], the controller approximation is obtained through a bottom-up approach, which is more suitable for FPGA implementation: the building blocks that compose the circuit are directly defined at low level, thus keeping the design simple and saving area occupancy. The solution proposed in [12] implements instead multivariate PWA functions for control purposes and is designed for VLSI implementation using a top-down method by describing its behavior in the C language and by optimizing the specific circuit architecture through the C-to-silicon compiler.

In this paper, we aim at pushing the sampling time in the nanoseconds range by adopting a special class of PWA approximating functions, described in the next section, that have a direct implementation counterpart on electronic circuits.

III. PWA SIMPLICIAL FUNCTIONS

We consider a compact domain $S \subset \mathbb{R}^n$ that admits a *polyhedral partition*. In other terms, $S = \bigcup_{i=0}^{L-1} \mathcal{X}_i$, where $\{\mathcal{X}_i\}_{i=0}^{L-1}$ are (possibly non-closed) polytopes such that $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$, $\forall i, j = 0, \ldots, L-1$, $i \neq j$. A (possibly discontinuous) PWA vector function $u : S \subset \Re^n \rightarrow \Re^m$ is defined as

$$u(x) = F_i x + g_i, \quad x \in \mathcal{X}_i. \tag{6}$$

If function u is continuous, one can always define the regions $\mathcal{X}_i$ of the partition as closed polyhedra $\mathcal{X}_i = \{x \in \mathbb{R}^n : H_i x \leq K_i\}$ as in (5) by only requesting that $\mathring{\Omega}_i \cap \mathring{\Omega}_j = \emptyset$, as no ambiguity in the definition of u would arise on possibly overlapping boundaries of different sets $\mathcal{X}_i$, $\mathcal{X}_j$.

Equation (6) defines PWA functions defined over S that have a general structure and, indeed, it can express every function, even discontinuous, that belongs to the function space $\mathrm{PWA}[S]$. A generic PWA function belongs to both $L^\infty[S]$ and to $L^2[S]$. For modeling and circuit implementation reasons we restrict our attention to the subclass of *continuous & regular* PWA functions, i.e., functions defined over a regular partition of the domain S into a set of *simplices* having the same shape, and assume that $S = \{x \in \mathbb{R}^n : x_{minj} \leq x_j \leq x_{maxj}, j = 1, \ldots, n\}$, i.e., S is a box (hyper-rectangle). The elements of this class, called *PWA Simplicial* (PWAS) functions, can be formally defined by introducing a simplicial partition of the domain and a set of basis functions.

In the following, we define PWAS functions starting from the basic notion of simplex and simplicial partitions.

A. Domain Partition and Basis Functions Definition

Definition 1: Given a set of $n+1$ vectors $x_i^0, x_i^1, \ldots, x_i^n \in \Re^n$ in common position, a *simplex* S_i in $\Re^n$ is their convex combination

$$S_i\left(x_i^0, \ldots, x_i^n\right) = \mathrm{conv}\left\{x_i^0, \ldots, x_i^n\right\}.$$

Vectors $x_i^0, x_i^1, \ldots, x_i^n$ are called the *vertices* of S_i. A simplex can also be represented by the hyperplanes that define its boundary, i.e., by a set of inequalities: $S_i(x_i^0, \ldots, x_i^n) = \{x : \hat{H}_i x \leq \hat{K}_i\}$. As shown in [27], matrices $\hat{H}_i$, $\hat{K}_i$ are defined directly by the inequalities

$$\begin{bmatrix} 1 & \ldots & 1 \\ x_i^0 & \ldots & x_i^n \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0 \tag{7}$$

that are a minimal system of linear inequalities representing S_i. In this work, we exploit both vertex and hyperplane representations of S_i.

The box S is partitioned into simplices as follows. Every dimensional component $x_j \in [x_{minj}, x_{maxj}]$ of S is divided into p_j subintervals of length $(x_{maxj} - x_{minj}/p_j)$, $j = 1, \ldots, n$. Consequently, the domain is divided into $\prod_{j=1}^n p_j$ hyper-rectangles and contains $N_v = \prod_{j=1}^n (p_j+1)$ vertices v_k, collected into the set $\mathcal{V}_s$. Each rectangle is further partitioned into $n!$ simplices with non-overlapping interiors and thus S contains $L = n! \prod_{j=1}^n p_j$ simplices S_i, such that $S = \cup_{i=0}^{L-1} S_i$ and $\mathring{S}_i \cap \mathring{S}_j = \emptyset$, $\forall i, j = 0, \ldots, L-1$. The resulting partition is called *simplicial partition* or type-1 triangulation and is univocally identified by the vector $p = [p_1\ \ldots\ p_n]'$. The corresponding class of continuous functions that are affine over each simplex constitutes an N_v-dimensional linear space $PWAS_p[S] \subset PWAS[S] \subset PWA[S] \cap C^0[S]$ [16]. Therefore, it is possible to define different bases, made up of N_v linearly independent functions belonging to $PWAS_p[S]$. By choosing some (arbitrary) ordering of the functions of any of these bases, we can regard them as an N_v-length vector, say $\phi(x)$. Then a scalar PWAS function $\hat{u} \in PWAS_p[S]$ is defined as a linear combination of the basis functions as follows:

$$\hat{u}(x) = \sum_{k=1}^{N_v} w_k\ \phi_k(x) = w'\phi(x). \tag{8}$$

Vector $w = [w_1\ \ldots\ w_{N_v}]'$ determines $\hat{u}$ uniquely for each given $x \in S$.

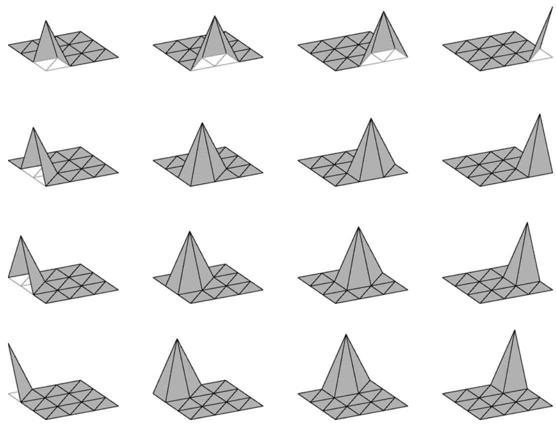

Fig. 1. The α-basis for a uniformly partitioned two-dimensional domain, with $p_1 = 3$ and $p_2 = 3$ (then $N_v = 16$).

A PWAS vector function $\hat{u} : \Re^n \to \Re^m$ is defined by the weights $w = [(w^1)' \ (w^2)' \ \ldots \ (w^m)']' \in \Re^{mN_v}$

$$\hat{u}(x) = \begin{bmatrix} \hat{u}_1(x) \\ \vdots \\ \hat{u}_m(x) \end{bmatrix} \triangleq \begin{bmatrix} (w^1)'\phi(x) \\ \vdots \\ (w^m)'\phi(x) \end{bmatrix} = \begin{bmatrix} \phi'(x) & 0 & \cdots & 0 \\ 0 & \phi'(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \phi'(x) \end{bmatrix} w = \Phi(x)w$$

Different types of basis functions can be defined; in this work we refer to the so-called α-basis [17]. Fig. 1 shows the functions of the α-basis defined over a uniformly-partitioned two-dimensional domain. See [18] and [28] for other basis functions. The kth function α is a PWAS hyper-pyramid (a pyramid if $n = 2$), which takes the value 1 at v_k and 0 at all the other vertices. In other words, every element of the basis is affine over each simplex and satisfies the condition

$$\alpha_k(v_h) = \begin{cases} 1, & \text{if } h = k \\ 0, & \text{if } h \neq k \end{cases}.$$

This definition implies an important property: when the α-basis is used, the coefficients w_k are the values of the PWAS function at the vertices of the simplicial partition, $w_k = \hat{u}(v_k)$. This fact establishes a one-to-one correspondence between each vertex v_k and a coefficient w_k.

B. Circuit Implementations of PWAS Functions

PWAS functions can be implemented in a circuit by using linear interpolators. In fact, the value of a PWAS function can be obtained, for any n-dimensional input vector, by linearly interpolating only the $n+1$ values assumed by the function at the vertices that define the simplex S_i the input vector belongs to

$$\hat{u}(x) = \sum_{j=0}^{n} \mu_j \hat{u}\left(x_i^j\right). \tag{9}$$

As a consequence, the circuit realization of a PWAS function contains three functional elements:

1) a memory where the N_v weights w_k are stored;
2) a block that, for any given point x, a) solves the point location problem, i.e., finds the simplex S_i such that $x \in S_i$, and b) computes the coefficients μ_j (this step requires a sorting algorithm);
3) a block performing the weighted sum (9).

The algorithm usually adopted to locate the simplex S_i is based on Kuhn's lemmas [29]–[31] and is optimal with respect to the number of inputs [32]. This algorithm exploits the regularity of the simplicial partition; thus, it is very simple, compared to other methods used to solve the point location problem.

By implementing this algorithm on a digital circuit, it turns up that the latency, i.e., the time required to evaluate the function, depends linearly on the number n of dimensions of the state space and on the number of bits used to code the input vectors. We remark that the latency is independent of the number of divisions p_j that define the simplicial partition. Some examples of digital circuit solutions for fast piecewise-linear interpolation can be found in [19] and [33]–[35].

IV. PWAS Approximations of MPC

In this section, we discuss some metrics and optimization techniques to find PWAS approximations of MPC in $L^2[S]$ and $L^\infty[S]$.

As shown in [19], circuits implementing PWAS functions are faster and simpler than those realizing general PWA functions. Thus, we want to find a PWAS control vector function $\hat{u} : \mathbb{R}^n \to \mathbb{R}^m$ that approximates the optimal control $u^* : \mathbb{R}^n \to \mathbb{R}^m$ fulfilling the constraints (3b) (and thus also constraints (2b), (2d), (2e), and (2f)). In other words, $\hat{u}(x)$ must be a feasible control input. We assume that the simplicial partition (i.e., vector p and vertex set $\mathcal{V}_s$) is fixed, and look for a (vector) function $\hat{u} \in PWAS_p[S]$ as close as possible to u^* according to two alternative functionals, whose minimization leads to an approximation of u^* in $L^2[S]$ or in $L^\infty[S]$. Depending on the chosen functional, we show in this section that the coefficients w that define $\hat{u}$ can be found by solving a QP or an LP problem.

Lemma 1: Let $\mathcal{V}$ be the set of vertices of the partition of the domain S of a given PWA scalar function $u : S \subset \mathbb{R}^n \to \mathbb{R}$. Then

$$\arg\max_{x \in S} u(x) \cap \mathcal{V} \neq \emptyset.$$

In other words, a maximum of u is always attained at one of the vertices in $\mathcal{V}$.

Proof: The proof is a simple consequence of the linearity of the PWA function inside each polytope of its partition. Suppose that the maximum of u lies in the ith polytope $\mathcal{X}_i$. Since u is affine over $\mathcal{X}_i$, its maximum lies on one of the vertices of $\mathcal{X}_i$, which is in turn a vertex of the set $\mathcal{V}$ of vertices of the domain partition.[2] □

The following corollary of Lemma 1 will be used in the sequel.

Corollary 1: For a given PWA function $u : S \subset \mathbb{R}^n \to \mathbb{R}^m$, $u(x) \leq 0, \forall x \in S$ if and only if $u(v) \leq 0, \forall v \in \mathcal{V}$, where $\mathcal{V}$ is the set of vertices of the partition.

[2]The maximum may not be unique in general, but at least one of the vertices is always a maximizer.

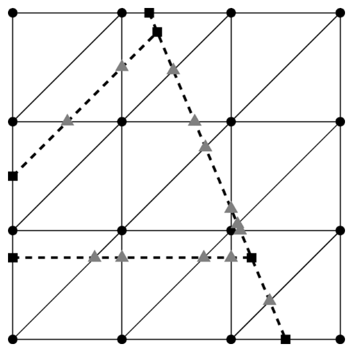

Fig. 2. Mixed partition of a two-dimensional domain. Dashed lines: irregular partition. Solid lines: simplicial partition. Black dots: vertices ($\in \mathcal{V}_s$) of the simplicial partition. Black squares: vertices ($\in \mathcal{V}_i$) of the irregular partition. Gray triangles: vertices $\in \mathcal{V}_m$.

A. Constraints on the Approximate Controller: Feasibility

In order to obtain an approximate MPC control law $\hat{u}$ enforcing the inequality constraints in (2), we need to define some constraints on the weights w. We distinguish two cases: A) only input constraints (2d) are present, B) constraints (2e) and/or (2f) are also present.

In case A), feasibility of $\hat{u}(x)$ is simply enforced by imposing

$$E_u \Phi(v) w \leq G_u, \; \forall v \in \mathcal{V}_s. \tag{10}$$

In this case the explicit MPC control law (5) is not required to determine a set of coefficients w satisfying (3b). Case B) requires instead that the exact explicit MPC control law u^* and its associated (possibly non-regular) PWA partition is available, in order to impose constraints on w that enforce (2e) and/or (2f). Let $\mathcal{V}_i$ be the set of vertices of the PWA partition defining u^*. We call *mixed partition* the partition (i.e., all the non-empty polytopes) of a given domain S induced by the facets of both simplicial and irregular partitions ($\hat{H}_i x \leq \hat{K}_i$ and $H_i x \leq K_i$). A mixed partition is composed by convex polytopes and is completely defined by the sets of vertices $\mathcal{V}_i$, $\mathcal{V}_s$, and $\mathcal{V}_m$, the latter being the set of vertices resulting from the intersections of the hyperplanes of the simplicial and irregular partitions and not belonging to $\mathcal{V}_i$ and $\mathcal{V}_s$. Fig. 2 shows a two-dimensional example of mixed partition.

By remembering that

$$U^*(x) = [u_0'^*(x) \; \ldots \; u_{N_u-1}'^*(x) \; \epsilon^*(x)]'$$

, constraints (3b) in general involve all the controls from instant k to $k + N_u - 1$. Since only $u^*(x) = u_0^*(x)$ is applied to system (1)—the other controls and ϵ are discarded—u^* is the only function we need to approximate and store. We define the following matrices and vectors:

- G_1 is the matrix collecting the first m columns of G [see (3b)];
- G_2 collects the columns of G related to $u_1^*, \ldots, u_{N_u-1}^*$;
- $-G_3$ is the column of G related to the slack variable ϵ for soft constraints, $G_3 > 0$ since $V_u, V_x > 0$;
- $U_{1,N-1}^*(x) = [u_1'^*(x) \; \ldots \; u_{N-1}'^*(x)]'$ is the vector of discarded future control inputs.

Any null row in G_1 must be discarded, together with the corresponding rows in W, D, G_2, G_3 as they do not involve u^*. We denote by $\bar{G}_1$, $\bar{W}$, $\bar{D}$, $\bar{G}_2$, $\bar{G}_3$ the reduced matrices.

Using the definitions above we have

$$[\bar{G}_1 \quad \bar{G}_2 \quad -\bar{G}_3] \begin{bmatrix} u^*(x) \\ U_{1,N-1}^*(x) \\ \epsilon^*(x) \end{bmatrix} \leq \bar{W} + \bar{D}x$$

i.e.,

$$\bar{G}_1 u^*(x) \leq \bar{W} + \bar{D}x - \bar{G}_2 U_{1,N-1}^*(x) + \bar{G}_3 \epsilon^*(x)$$

which defines the constraints fulfilled by u^*. Since $\hat{u}$ must be feasible too, the same set of constraints is considered for the approximated control. Then

$$\bar{G}_1 \hat{u}(x) - \bar{W} - \bar{D}x + \bar{G}_2 U_{1,N-1}^*(x) \leq \bar{G}_3 \epsilon^*(x), \; \forall x \in S. \tag{11}$$

The left-hand side of (11) is a PWA function defined over the mixed partition obtained by joining the simplicial partition of $\hat{u}$ and the irregular partition of $U_{1,N-1}^*$. Thus, from Lemma 1, the PWA constraints are fulfilled for all $x \in S$ if and only if

$$\bar{G}_1 \hat{u}(v) \leq \bar{W} + \bar{D}v - \bar{G}_2 U_{1,N-1}^*(v) + \bar{G}_3 \epsilon^*(v), \quad \forall v \in (\mathcal{V}_s \cup \mathcal{V}_i \cup \mathcal{V}_m) \tag{12}$$

By using a PWAS basis to represent $\hat{u}$, we have $\hat{u}(x) = \Phi(x) w$, so that (12) can be written as

$$\bar{G}_1 \Phi(v) w - \bar{G}_3 \sigma(v) \leq \bar{W} + \bar{D}v - \bar{G}_2 U_{1,N-1}^*(v) + \bar{G}_3 \epsilon^*(v), \quad \forall v \in (\mathcal{V}_s \cup \mathcal{V}_i \cup \mathcal{V}_m) \tag{13}$$

where $\sigma : S \to \mathbb{R}$, $\sigma(x) = w_\sigma' \phi(x)$, $\sigma(x) \geq 0$, is a PWAS function that allows softening constraints (2e), (2f) further than the softening induced by ϵ^*. Thus, $\sigma(x)$ should be ideally zero everywhere. The elements of vector $w_\sigma \in \Re^{N_v}$, that are the coefficients of the PWAS slack function σ, are additional decision variables. The reason for introducing σ on top of ϵ^* is twofold. First, to compensate the need for possible extra softening due to the different PWA partitions over which the approximating function $\hat{u}$ and the "optimal" slack ϵ^* are defined. A one-dimensional example showing the need for such an extra softening is depicted in Fig. 3: no PWAS function $\hat{u}(x)$ exists such that $\underline{u}(x) \leq \hat{u}(x) \leq \overline{u}(x)$, where $\overline{u}$ and $\underline{u}$ are PWA functions defined over the same partition of u^*. Second, the original "optimal" softening induced by ϵ^* would be lost if only σ is used to soften the constraints enough to find a feasible $\hat{u}$. Hence, keeping both σ and ϵ^* allows introducing a minimal perturbation from the optimal softening that guarantees feasibility of $\hat{u}$.

B. Constraints on the Approximate Controller: Local Optimality

Assumption 1: The simplex S_0 is such that $0 \in \mathring{S}_0 \subseteq \mathcal{X}_0$.

Since, as observed earlier, $0 \in \mathring{\Omega}_0$, a simplicial partition S satisfying Assumption 1 can always be determined. Under Assumption 1, the further set of constraints

$$\hat{u}(v) = \Phi(v) w = u^*(v), \; \forall \text{ vertex } v \text{ of } S_0 \tag{14}$$

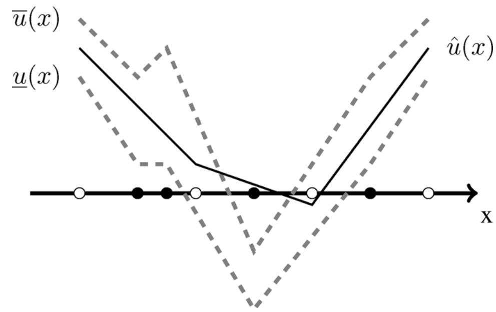

Fig. 3. PWA constraints in a one-dimensional case. The vertices of the simplicial partition are denoted by empty circles, whereas the vertices of the irregular partition are denoted by black dots.

imposes the optimality of $\hat{u}$ around the origin. In particular, since u^* is linear in $\mathcal{X}_0$, the constraints in (14) impose that

$$\hat{u}(x) = u^*(x), \quad \forall x \in S_0 \tag{15}$$

which makes the approximate and exact MPC control laws equal for small signals around the origin, a desirable condition to inherit local stability and local frequency response properties of the original MPC controller [36].

In order to meet the requirement of Assumption 1, the simplicial partition must be fine enough. This fact may have implications on the complexity of the approximated controller. As a partial solution, one may apply a change of coordinates to cope with a slanting and flattened $\mathcal{X}_0$ and/or resort to a non-uniform simplicial partition (see [37]).

C. Constraints on the Approximate Controller: Invariance

The approximate control law $\hat{u}$ is only defined over the selected simplicial partition of S. The set S may be chosen arbitrarily large so that, for the set of initial conditions $x(0)$ of interest, the control law is always defined on the closed-loop trajectory $x(t)$. Rather than looking for the subset of initial states in S for which $x(t) \in S, \forall t \geq 0$, we impose additional constraints on $\hat{u}$ such that the state vector always remains within a polyhedral invariant set $\Omega_{\mathcal{E}} = \{x \in \mathbb{R}^n : H_{\mathcal{E}} x \leq K_{\mathcal{E}}\}$ containing S, that will be determined in Section V-B:

$$H_{\mathcal{E}}\left(Av + B\Phi(v)w\right) \leq K_{\mathcal{E}}, \quad \forall v \in \mathcal{V}_s \tag{16}$$

imposes that for all $x \in S$, the updated state $Ax + B\hat{u}(x) \in \Omega_{\mathcal{E}}$. Roughly speaking, constraints (16) should mainly affect the resulting control law $\hat{u}(x)$ only towards the boundaries of S. Note that imposing invariance of S instead of $\Omega_{\mathcal{E}}$ may be very stringent, especially if S is large and input constraints are tight.

D. Definition of the Approximation Functionals

The first functional is defined working in the infinite-dimensional Hilbert space L^2 and using the metrics induced by the usual L^2 inner product extended to vector functions

$$\mathcal{F}_2(\hat{u}, w_\sigma) = \int_S \|u^*(x) - \hat{u}(x)\|^2 \, dx + \tau \|w_\sigma\|^2$$

where $\tau \gg 0$ is a weighting factor chosen in order to weight the constraints violation due to the presence of the PWAS slack function $\sigma(x)$ in (11). In particular, the second addend with $\tau \gg 0$ tends to make the PWAS slack function σ as close to zero as possible. Considering that $\hat{u} \in PWAS_p[S]$, $\mathcal{F}_2$ reduces to a cost function F_2:

$$\begin{aligned}\mathcal{F}_2(\hat{u}, w_\sigma) &= F_2(w, w_\sigma) \\ &= \sum_{q=1}^{m} \int_S \left[u_q^*(x) - (w^q)'\phi(x)\right]^2 dx + \tau \|w_\sigma\|^2.\end{aligned}$$

F_2 can be expressed as

$$F_2(w, w_\sigma) = \|\Psi w - d\|^2 + \tau \|w_\sigma\|^2 \tag{17}$$

where

$$\Psi = \begin{bmatrix} \hat{\Psi} & 0 & \cdots & 0 \\ 0 & \hat{\Psi} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\Psi} \end{bmatrix} \qquad d = \begin{bmatrix} d^1 \\ d^2 \\ \vdots \\ d^m \end{bmatrix}.$$

$\hat{\Psi} \in \Re^{N_v \times N_v}$ is the square matrix of the L^2 inner products between basis functions, $[\hat{\Psi}_{ij}] = \langle \phi_i, \phi_j \rangle$, and d^q is given by: $d_i^q = \langle u_q^*, \phi_i \rangle$, $q = 1, \ldots, m$.

The approximated control can be found by solving the QP problem

$$\begin{aligned} \min_{w, w_\sigma} \quad & F_2(w, w_\sigma) && \text{(18a)} \\ \text{s.t.}: \quad & w_\sigma \geq 0 \\ & \text{constraints (13), (14), (16).} && \text{(18b)} \end{aligned}$$

The second functional is based on the usual metrics in $L^\infty[S]$

$$\mathcal{F}_\infty(\hat{u}, w_\sigma) = \max_{q=1,\ldots,m} \sup_{x \in S} \left\{\left|u_q^*(x) - \hat{u}_q(x)\right|\right\} + \tau \|w_\sigma\|_1$$

where $\tau \gg 0$ as before. In this case, the approximated control is given by the solution of the following LP problem:

$$\begin{aligned} \min_{w, w_\sigma, \eta} \quad & \eta + \tau \sum_{k=1}^{N_v} w_{\sigma k} && \text{(19a)} \\ \text{s.t.}: \quad & \eta \geq \pm \left[(w^q)'\phi(v) - u_q^*(v)\right], \; q = 1, \ldots, m, \\ & \quad \forall v \in (\mathcal{V}_s \cup \mathcal{V}_i \cup \mathcal{V}_m) && \text{(19b)} \\ & w_\sigma \geq 0 \\ & \text{constraints (13), (14), (16)} && \text{(19c)} \end{aligned}$$

In both the QP (18) and LP (19) formulations, the simplicial partition determines the approximation accuracy (i.e., the p_j's), since the number of vertices equals the number N_v of basis functions, and the location of the vertices influences the structure of the PWAS function.

The resulting PWAS function $\hat{u}$ is not affected by the choice of the basis functions, because each basis spans $PWAS_p[S]$. However, the numerical complexity of problems (18) and (19) is strongly affected by the basis choice. Indeed, by using the α-basis, the matrices of constraints (13) and (14) are sparse, as well as matrix Ψ in (17) and constraints (19c), thus reducing memory requirements and computational times. Moreover, once the set of coefficients w that represents the PWAS approximated control has been calculated, the implementation of $\hat{u}$ by means of linear interpolators is straightforward, since the coefficients of the α-basis represent the values of $\hat{u}$ at the vertices of the simplicial partition, as pointed out in Section III-A.

From now on, we will refer to the approximate controller $\hat{u}(x)$ as

$$\hat{u}(x) = \begin{cases} F_0 x, & \text{if } x \in S_0 \\ \hat{F}_1 x + \hat{g}_1, & \text{if } x \in S_1 \\ \vdots & \quad\vdots \\ \hat{F}_L x + \hat{g}_L, & \text{if } x \in S_{L-1} \\ F_{\mathcal{E}} x, & \text{if } x \notin S, \end{cases} \tag{20}$$

where $S_i \triangleq \{x : \hat{H}_i x \le \hat{K}_i\}$, $i = 0, \ldots, L$, $S_0 \subseteq \mathcal{X}_0$, $0 \in \overset{\circ}{S}_0$, and where we have added a backup gain $F_{\mathcal{E}}$ outside S, that will be defined in Section V-B. Note that each linear gain $\hat{F}_i$, $\hat{g}_i$ is obtained by solving the linear system

$$\begin{bmatrix} x_i^0 & \ldots & x_i^n \\ 1 & \ldots & 1 \end{bmatrix}' \begin{bmatrix} \hat{F}_i' \\ \hat{g}_i \end{bmatrix} = \begin{bmatrix} \hat{u}\left(x_i^0\right) \\ \vdots \\ \hat{u}\left(x_i^n\right) \end{bmatrix} \tag{21}$$

where, according to Section III-A, x_i^j denotes a vertex of the simplex S_i, $i = 0, \ldots, L-1$, $j = 0, \ldots, n$.

E. Suboptimality Analysis

The control law (20) provides an approximation of the exact MPC control law (5). The degree of approximation clearly depends on the coarseness of the partitions in S. In this section, we aim at quantifying such a dependence.

For each region $\mathcal{X}_i$ in (5), $i = 0, \ldots, n_r - 1$, denote by $\Gamma(i) \subseteq \{0, \ldots, L-1\}$ the set of indices j of the simplices S_j intersecting $\mathcal{X}_i$. Rather than computing L LPs to determine the elements of $\Gamma(i)$, for all $i = 0, \ldots, n_r - 1$, due to the regularity of the simplicial partition one can restrict the number of LPs by computing the bounding box $\mathcal{B}_i$ of $\mathcal{X}_i$ and only check intersections with the simplices S_j whose bounding box intersects $\mathcal{B}_i$, an operation which only requires comparisons. It is immediate to prove the following lemma.

Lemma 2: The maximum approximation error

$$M = \max_{x \in S} \|u(x) - \hat{u}(x)\|_\infty \tag{22}$$

is given by

$$\max_{\substack{q=1,\ldots,m \\ i=0,\ldots,n_r-1 \\ j \in \Gamma(i)}} \left\{M_{ijq}^-, M_{ijq}^+\right\}$$

where

$$M_{ijq}^{\pm} = \begin{cases} \max_x & \pm\left(\left(F_i^{(q)} - \hat{F}_j^{(q)}\right)x + \left(g_{i_q} - \hat{g}_{j_q}\right)\right) \\ \text{s.t.} & H_i x \le K_i \\ & \hat{H}_j x \le \hat{K}_j \end{cases}$$

for $i = 0, \ldots, n_r - 1$, $j \in \Gamma(i)$, $q = 1, \ldots, m$.

V. Stability Analysis

As the proposed MPC approximation procedure does not impose conditions for closed-loop stability, such properties must be analyzed *a posteriori*.

As pointed out in [38], the intimate relationship between closed-loop MPC systems and piecewise affine (PWA) dynamical systems allows one use stability analysis tools developed for hybrid systems for checking closed-loop stability properties of (explicit) MPC. Stability methods for PWA systems based on piecewise quadratic Lyapunov functions and linear matrix inequality relaxations [39], [40] were applied in [10] for analyzing closed-loop stability of reduced-complexity explicit MPC controllers. Piecewise linear Lyapunov functions were considered in [40]–[42] for stability of PWA closed-loop systems.

The approach taken in this paper for analyzing closed-loop stability under the approximate MPC law (20) consists of synthesizing 1) a piecewise linear (PWL) Lyapunov function around the origin;[3] 2) a backup gain $F_{\mathcal{E}}$ and the corresponding polyhedral invariant set $\Omega_{\mathcal{E}}$ containing S; 3) a PWA Lyapunov function on $\Omega_{\mathcal{E}}$ by solving an LP problem, tailored to the special structure of controller (20) and exploiting the double description (hyperplane and vertex representations) of simplices S_i, which is immediately available.

A. Inner λ-Contractive Set

In this section, we exploit contractive polyhedral sets and Minkowski functions to synthesize a PWL Lyapunov function, following ideas in the spirit of [43].

Definition 2: For a given $0 < \lambda < 1$, a set $\Omega \subseteq \mathbb{R}^n$ with $\lambda\Omega \subset \Omega$ and $0 \in \overset{\circ}{\Omega}$ is called a *λ-contractive set* for system $x_{k+1} = A_0 x_k$ if for all $x \in \Omega$, it holds that $A_0 x(k) \in \lambda\Omega$.[4]

A λ-contractive and finitely-generated polyhedral set Ω can be constructed as in [43], [44], or alternatively as suggested in [45]. In [43] and [44], the maximal λ-contractive set Ω contained in a polyhedron $\mathcal{P}$ is constructed recursively as

$$\mathcal{X}_0 = \mathcal{P},\ \mathcal{X}_{k+1} = \{x \in \mathbb{R}^n : A_0 x \in \lambda\mathcal{X}_k\} \cap \mathcal{P},\ \Omega = \bigcap_{k=0}^{\infty} \mathcal{X}_k. \tag{23}$$

The following Lemmas 3 and 4 provide a constructive method to synthesize such a PWL function for the local closed-loop system $x_{k+1} = A_0 x_k$, $A_0 \triangleq A + BF_0$, based on a connection with the theory of maximum output admissible sets [46].

Lemma 3: The largest λ-contractive set Ω contained in a polytope $\mathcal{P} = \{x : H_P x \le K_P\}$, with $K_P > 0$, is the maximum output admissible invariant set for the linear closed-loop system $\tilde{x}_{k+1} = (1/\lambda)A_0\tilde{x}_k$ under the constraint $x \in \mathcal{P}$

$$\Omega = \left\{x \in \mathbb{R}^n : H_P \left(\frac{1}{\lambda}A_0\right)^k x \le K_P,\ \forall k \ge 0\right\}. \tag{24}$$

[3]In this paper, we use the term "PWL function" to indicate a PWA function that is null at the origin.

[4]One can extend the definition to the case $\lambda = 0$ (contraction in the origin in one step), and $\lambda = 1$, in which case $\Omega = \lambda\Omega$ is called an *invariant* set.

Proof: Since $\lambda\mathcal{P} = \{y : y = \lambda x, H_P x \le K_P\} = \{y : H_P(1/\lambda)y \le K_P\} = \{y : H_P y \le \lambda K_P\}$, the sequence of sets

$$\begin{aligned}\mathcal{X}_0 &= \mathcal{P}\\ \mathcal{X}_1 &= \{x \in \mathbb{R}^n : A_0 x \in \lambda\mathcal{P}\} \cap \mathcal{P}\\ &= \left\{x \in \mathbb{R}^n : \begin{bmatrix} H_P A_0 \\ H_P \end{bmatrix} x \le \begin{bmatrix} \lambda K_P \\ K_P \end{bmatrix}\right\}\\ \mathcal{X}_2 &= \{x \in \mathbb{R}^n : A_0 x \in \lambda\mathcal{X}_1\} \cap \mathcal{P}\\ &= \left\{x \in \mathbb{R}^n : \begin{bmatrix} H_P A_0^2 \\ H_P A_0 \\ H_P \end{bmatrix} x \le \begin{bmatrix} \lambda^2 K_P \\ \lambda K_P \\ K_P \end{bmatrix}\right\}\\ &\vdots\\ \mathcal{X}_k &= \left\{x \in \mathbb{R}^n : H_P A_0^j x \le \lambda^j K_P,\ \forall j = 0, \ldots, k\right\}\end{aligned}$$

is such that $\bigcap_{k=0}^{h} \mathcal{X}_k = \mathcal{X}_h$, and therefore (23) and (24) are equivalent, as $\Omega = \bigcap_{k=0}^{\infty} \mathcal{X}_k = \lim_{h\to\infty} \mathcal{X}_h$.

Next Lemma 4 provides a constructive method to synthesize a λ-contractive polyhedral set by determining an upperbound N_0 to the number of steps k in (24), and therefore to represent Ω as a polytope generated by a finite (possibly non-minimal) number of hyperplanes.

Lemma 4: Consider the linear system $x_{k+1} = A_0 x_k$ and a polytope $\mathcal{P} = \{x : H_P x \le K_P\} = \operatorname{conv}(x^1, \ldots, x^{q_v})$ with $K_P > 0$, $K_P \in \mathbb{R}^{q_h}$, and let $A_0 \in \mathbb{R}^{n\times n}$ such that its spectral radius $\mu < 1$. Fix any scalar λ such that $\mu < \lambda \le 1$, and let $P \in \mathbb{R}^{n\times n}$, $P = P' \succ 0$, be the solution of the Lyapunov equation

$$P - \frac{1}{\lambda^2} A_0' P A_0 = I \tag{25}$$

Let

$$\gamma_1 \triangleq \max_{j=1,\ldots,q_v} \left\{(x^j)' P x^j\right\} \tag{26a}$$

$$\gamma_2 \triangleq \min_{i=1,\ldots,q_h} \left\{\frac{(K_{Pi})^2}{H_P^{(i)} P^{-1} \left(H_P^{(i)}\right)'}\right\} \tag{26b}$$

and denote by μ_P the spectral radius of P, $\mu_P \in \mathbb{R}$, $\mu_P > 0$. Then:

1) the sets $\mathcal{E}_1 \triangleq \{x \in \mathbb{R}^n : x'Px \le \gamma_1\}$ and $\mathcal{E}_2 \triangleq \{x \in \mathbb{R}^n : x'Px \le \gamma_2\}$ are, respectively, the smallest λ-contractive ellipsoidal set containing $\mathcal{P}$ and the largest λ-contractive ellipsoidal set contained in $\mathcal{P}$;
2) the largest λ-contractive polyhedral set Ω contained in $\mathcal{P}$ is finitely generated by at most $\bar{n}_\Omega \triangleq q_h N_0$ linear inequalities

$$\begin{aligned}\Omega &= \left\{x \in \mathbb{R}^n : H_P A_0^k x \le \lambda^k K_P,\ \forall k = 0, \ldots, N_0\right\}\\ &\triangleq \{x \in \mathbb{R}^n : H_\Omega x \le K_\Omega\}\end{aligned} \tag{27}$$

where

$$N_0 = \left\lceil \frac{\log \frac{\gamma_2}{\gamma_1}}{\log\left(1 - \frac{1}{\mu_P}\right)} \right\rceil \tag{28}$$

and $H_\Omega \in \mathbb{R}^{\bar{n}_\Omega \times n}$, $K_\Omega \in \mathbb{R}^{\bar{n}_\Omega}$.

Proof:

1) As $\lambda > \mu$, the linear system $\tilde{x}_{k+1} = (1/\lambda)A_0\tilde{x}_k$, $\tilde{x}_0 = x_0$, is asymptotically stable, so (25) admits a solution $P = P' \succ 0$. It is easy to prove that γ_1 is the smallest scalar and γ_2 is the largest scalar such that $\mathcal{E}_2 \subset \mathcal{P} \subset \mathcal{E}_1$. Since $K_P > 0$, $\mathcal{P}$ is a polyhedron such that $0 \in \mathring{\mathcal{P}}$, and therefore $0 < \gamma_2 < \gamma_1$. We next prove λ-contractiveness of any ellipsoid $\mathcal{E} = \{x : x'Px \le \gamma\}$, $\gamma > 0$. Take a generic vector $x \in \mathcal{E}$ and set $y = A_0 x$; to prove that $A_0 x \in \lambda\mathcal{E}$ we need to find a vector $z \in \mathcal{E}$ such that $y = \lambda z$. By (25) we get $z'Pz = (1/\lambda^2)y'Py = (1/\lambda^2)(A_0x)'P(A_0x) = (1/\lambda^2)\lambda^2 x'(P - I)x \le x'Px \le \gamma$, and hence $y = A_0 x \in \lambda\mathcal{E}$, which proves in particular λ-contractiveness of $\mathcal{E}_1$, $\mathcal{E}_2$.
2) Since $0 \in \mathring{\mathcal{P}}$, by [46] it follows that a finite integer N_0 exists such that Ω is finitely generated by the intersection of $q_h N_0$ half-spaces. As $\tilde{x}'P\tilde{x} \le \mu_P \tilde{x}'\tilde{x}$ and by (25) it follows that for any $\tilde{x} \in \mathbb{R}^n$

$$\tilde{x}'\left(P - \frac{1}{\lambda^2}A_0'PA_0\right)\tilde{x} = \tilde{x}'\tilde{x} \ge \frac{1}{\mu_P}\tilde{x}'P\tilde{x}. \tag{29}$$

Take any $x_0 = \tilde{x}_0 \in \mathcal{P}$, which implies $\tilde{x}_0 \in \mathcal{E}_1$. Then, for all $k \ge 0$, it follows by (29) that $\tilde{x}_k'P\tilde{x}_k = (1/\lambda^2)\tilde{x}_{k-1}'(A_0'PA_0)x_{k-1} \le (1 - (1/\mu_P))\tilde{x}_{k-1}'P\tilde{x}_{k-1} \le (1 - (1/\mu_P))^k \tilde{x}_0'P\tilde{x}_0 \le [(1 - (1/\mu_P))]^k\gamma_1 \le \gamma_2$ for $k = N_0$, that is $\tilde{x}_{N_0} \in \mathcal{E}_2 \subset \mathcal{P}$. Note that N_0 in (28) is a well-defined positive integer, as by (25) $P \succeq I$, which in turn implies $\mu_P \ge 1$, or, equivalently, $0 \le (1 - (1/\mu_P)) < 1$, so that N_0 in (28) rounds the ratio of two negative numbers. By (25) it immediately follows that $\mathcal{E}_2$ is an invariant set for $\tilde{x}_{k+1} = (1/\lambda)A_0\tilde{x}_k$, so that $\tilde{x}_k \in \mathcal{E}_2 \subset \mathcal{P}$, $\forall k \ge N_0$. This proves that (27) is an invariant set for $\tilde{x}_{k+1} = (1/\lambda)A_0\tilde{x}_k$, and, as proved in Lemma 3, a λ-contractive set for $x_{k+1} = A_0 x_k$. Hence, (24) and (27) are equivalent definitions of Ω. □

By setting $\mathcal{P} = S_0$, $q_v = n + 1$, $q_h = n + 1$, Lemma 4 provides a constructive method to synthesize a λ-contractive set Ω contained in S_0, for all $\lambda > \mu$. A possible choice for a contraction factor $\lambda < 1$ is for instance $\lambda = \sqrt{\mu}$. A minimal hyperplane representation of Ω is obtained by removing possible further redundant inequalities, which can be done by solving the following $\bar{n}_\Omega$ LP problems:

$$\begin{aligned}\sigma_i = \max_x \quad & H_\Omega^{(i)} x\\ \text{s.t.} \quad & H_\Omega^{(j)} x - K_{\Omega_j} \le 0,\ j = 1, \ldots, \bar{n}_\Omega,\ j \ne i\end{aligned} \tag{30}$$

and removing all inequalities i for which $\sigma_i \le K_{\Omega_i}$. We denote by $\Omega = \{x \in \mathbb{R}^n : G_0 x \le \mathbb{1}\}$ the resulting minimal representation of Ω, where $\mathbb{1} \in \mathbb{R}^{n_\Omega}$, $n_\Omega \le \bar{n}_\Omega$, $G_0 \in \mathbb{R}^{n_\Omega \times n}$. The following lemma will be useful in the sequel (cf. [47, Theorem 3.3(d)].

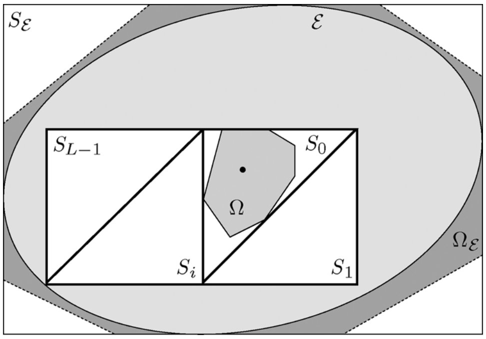

Fig. 4. Construction of the λ-contractive ellipsoidal set $\mathcal{E}$ and polytopic set $\Omega_{\mathcal{E}}$ ($\varphi = 1$).

Lemma 5: Let $\Omega = \{x \in \mathbb{R}^n : G_0 x \leq \mathbb{1}\}$ be a λ-contractive invariant set for the linear system $x_{k+1} = A_0 x_k$. Then, the Minkowski function $V_0 : \mathbb{R}^n \to \mathbb{R}$

$$V_0(x) = \max_{j=1,\dots,n_\Omega} \left\{ G_0^{(j)} x \right\} = \begin{cases} G_0^{(1)} x, & \text{if } \left(G_0^{(1)} - G_0^{(h)}\right) x \geq 0, \\ & \forall h = 1,\dots,n_\Omega \\ \vdots & \vdots \;\; \vdots \\ G_0^{(n_\Omega)} x, & \text{if } \left(G_0^{(n_\Omega)} - G_0^{(h)}\right) x \geq 0, \\ & \forall h = 1,\dots,n_\Omega \end{cases}$$

is a PWL Lyapunov function.

Proof: Let $x \in \Omega$, and define vector $z \triangleq (1/V_0(x))x$. As $x \in \Omega$, it follows that $\max_{j=1,\dots,n_\Omega}\{G_0^{(j)} z\} = (1/V_0(x)) \max_{j=1,\dots,n_\Omega}\{G_0^{(j)} x\} = 1$, that is $z \in \Omega$. Since Ω is λ-contractive, $A_0 z \in \lambda\Omega$, that is $A_0 z \in \{\bar{x} \in \mathbb{R}^n : \bar{x} = \lambda y, G_0^{(i)} y \leq 1, \forall i = 1,\dots,n_\Omega\} = \{\bar{x} \in \mathbb{R}^n : G_0^{(i)} \bar{x} \leq \lambda \mathbb{1}, \forall i = 1,\dots,n_\Omega\}$. This in turn implies that $\max_{j=1,\dots,n_\Omega}\{G_0^{(j)} A_0 z\} \leq \lambda$, and hence that $(1/V_0(x)) \max_{j=1,\dots,n_\Omega}\{G_0^{(j)} A_0 x\} \leq \lambda$, i.e., $V_0(A_0 x) \triangleq \max_{j=1,\dots,n_\Omega}\{G_0^{(j)} A_0 x\} \leq \lambda V_0(x)$, or, equivalently, that $V_0(A_0 x) - V_0(x) \leq -(1-\lambda)V_0(x)$. □

From now on we denote the subdomains of $V_0(x)$ by $R_i \triangleq \{x \in \mathbb{R}^n : G_0^{(h)} x \leq G_0^{(i)} x \leq 1, \forall h = 1,\dots,n_\Omega\}$, $i = 1,\dots,n_\Omega$.

B. Outer λ-Contractive Polyhedral Set

We want to determine a polytopic λ-contractive set $\Omega_{\mathcal{E}}$ covering the given partition S under a proper feedback gain $F_{\mathcal{E}}$. For the closed-loop system $x_{k+1} = (A + BF_{\mathcal{E}})x_k$, we first derive an associated λ-contractive ellipsoid $\mathcal{E} \subset \mathbb{R}^n$ containing S, and then determine a λ-contractive polyhedron $\Omega_{\mathcal{E}}$ containing $\mathcal{E}$. The idea for determining $F_{\mathcal{E}}$ is sketched in Fig. 4, and is based on the following Lemma.

Lemma 6: Let $(Q_{\mathcal{E}}, Y_{\mathcal{E}})$ be the optimizer of the following semidefinite program

$$\min_{Q,Y} \quad -\log\det(Q) \tag{31a}$$

$$\text{s.t.} \quad \begin{bmatrix} Q & (AQ+BY)' \\ AQ+BY & \lambda Q \end{bmatrix} \succeq 0 \tag{31b}$$

$$\begin{bmatrix} I_m & B_{\mathcal{E}}^{-1} Y \\ Y' \left(B_{\mathcal{E}}^{-1}\right)' & Q \end{bmatrix} \succeq 0 \tag{31c}$$

$$\begin{bmatrix} 1 & \left(x_i^j\right)' \\ x_i^j & Q \end{bmatrix} \succeq 0$$

$$\forall (i,j) \text{ such that } x_i^j \text{ is vertex of } S$$

$$j = 0,\dots,n,\; i = 0,\dots,L-1 \tag{31d}$$

$$Q = Q' \succ 0 \tag{31e}$$

where $B_{\mathcal{E}}$ solves the semidefinite program

$$\begin{aligned} \min_{B_{\mathcal{E}}} \quad & \log\det B_{\mathcal{E}}^{-1} \\ \text{s.t.} \quad & \left\| B_{\mathcal{E}} E_u^{(i)} \right\|_2 \leq G_{u_i},\; i = 1,\dots,n_u \end{aligned} \tag{32}$$

with $E_u \in \mathbb{R}^{n_u \times m}$, $G_u \in \mathbb{R}^{n_u}$. Then $P_{\mathcal{E}} \triangleq Q_{\mathcal{E}}^{-1}$, $F_{\mathcal{E}} \triangleq Y_{\mathcal{E}} Q_{\mathcal{E}}^{-1}$, and $\mathcal{E} = \{x : x' P_{\mathcal{E}} x \leq 1\}$ enjoy the following properties:

1) $V_{\mathcal{E}} : \mathbb{R}^n \to \mathbb{R}$, $V_{\mathcal{E}}(x) = x' P_{\mathcal{E}} x$, is a Lyapunov function for the feedback system $x_{k+1} = (A + BF_{\mathcal{E}})x_k$ such that

$$V_{\mathcal{E}}\left((A + BF_{\mathcal{E}})x\right) \leq \lambda V_{\mathcal{E}}(x) \tag{33}$$

and $\mathcal{E}$ is a λ-contractive ellipsoid containing S for the closed-loop system $x_{k+1} = (A + BF_{\mathcal{E}})x_k$;

2) the command input $u = F_{\mathcal{E}} x$ fulfils constraints (2d) for all $x \in \mathcal{E}$.

Proof: 1) By taking Schur complements, condition (31b) implies $\lambda P_{\mathcal{E}} - (A + BF_{\mathcal{E}})' P_{\mathcal{E}} (A + BF_{\mathcal{E}}) \succeq 0$, and hence implies the stability condition (33) and the λ-contractiveness of $\mathcal{E}$. Similarly, (31d) implies that $(x_i^j)' P (x_i^j) \leq 1$, and hence that $S \subset \mathcal{E}$. 2) Similarly to the method in [48], by setting $U_{\mathcal{E}} = (B_{\mathcal{E}}' B_{\mathcal{E}})^{-1}$, the semidefinite program (32) determines the largest ellipsoid $\mathcal{E}_u = \{u \in \mathbb{R}^m : u' U_{\mathcal{E}} u \leq 1\}$ centered in the origin and contained in the set of admissible inputs $\{u \in \mathbb{R}^m : F_u u \leq G_u\}$ [49, Ch. 8.4.2]. As constraint (31c) enforces $\|B_{\mathcal{E}} F_{\mathcal{E}} x\|_2 \leq 1$ for all $x \in \mathcal{E}$ [48], it follows that $u = F_{\mathcal{E}} x \in \mathcal{E}_u \subset \{u \in \mathbb{R}^m : E_u u \leq G_u\}, \forall x \in \mathcal{E}$. □

The cost function (31c) aims at maximizing the volume of the ellipsoid $\mathcal{E}$. Hard constraints (2e), (2f) for $V_u = 0$, $V_x = 0$ are not imposed in (31) to also maximize the size of $\mathcal{E}$.

Note that constraint (31d) may cause infeasibility of problem (31) because the selected set S is too large for the achievable domain of attraction of the posed constrained regulation problem. In this case, one should revise the set of states S for which the simplicial PWA approximation is obtained by making S smaller.

Lemma 7: Let $\Omega_{\mathcal{E}}$ be the largest λ-contractive set contained in the polytope

$$\mathcal{P}_{\mathcal{E}} = \{x \in \mathbb{R}^n : E_u F_{\mathcal{E}} x \leq G_u, x \in \varphi S_{\mathcal{E}}\} \tag{34}$$

for the closed-loop system $x_{k+1} = (A + BF_{\mathcal{E}})x_k$ under input constraints (2d), where $S_{\mathcal{E}}$ is the bounding box of $\mathcal{E}$, and $\varphi \geq 1$ is a fixed scalar. Then $\Omega_{\mathcal{E}} = \{x \in \mathbb{R}^n : H_{\mathcal{E}} x \leq K_{\mathcal{E}}\}$ is a polytopic λ-contractive set containing S.

Proof: By applying Lemma 3 and Lemma 4 with F_0 replaced by $F_{\mathcal{E}}$, and $\mathcal{P}$ replaced by the polytope $\mathcal{P}_{\mathcal{E}}$, it follows that $\Omega_{\mathcal{E}}$ is the largest λ-contractive polyhedral set contained in $\mathcal{P}_{\mathcal{E}}$

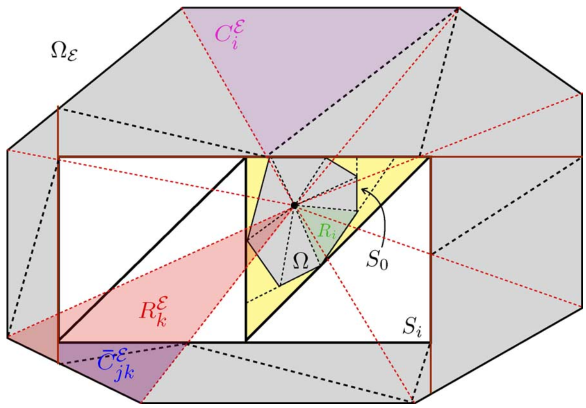

Fig. 5. Construction of the PWA decreasing function $V(x)$.

under the control law $u = F_{\mathcal{E}}x$, with $E_u u \leq G_u$. By Lemma 6, $\mathcal{E}$ is also λ-contractive under the same control law. Hence, $\Omega_{\mathcal{E}} \cup \mathcal{E}$ is a λ-contractive set. Since $\Omega_{\mathcal{E}}$ is the *largest* λ-contractive set, it must contain $\Omega_{\mathcal{E}} \cup \mathcal{E}$, and therefore $\mathcal{E}$. As $S_{\mathcal{E}}$ is a bounded set, $\Omega_{\mathcal{E}} \subseteq \varphi S_{\mathcal{E}}$ is also bounded. □

Note that the scalar φ can be chosen arbitrarily large to maximize the size of $\Omega_{\mathcal{E}}$. However, removing $x \in \varphi S_{\mathcal{E}}$ from (34) (i.e., setting $\varphi = +\infty$) may result in a possibly unbounded set $\Omega_{\mathcal{E}}$. Although the following arguments may be extended to an unbounded set $\Omega_{\mathcal{E}}$ by introducing unbounded simplices (using extreme rays besides vertices), we prefer to enforce boundedness of $\Omega_{\mathcal{E}}$ to simplify the exposition, and since anyway one is interested in states inside S, where the PWA approximation of the original MPC controller is performed.

C. PWA Lyapunov Function

Given a simplicial partition $S = \bigcup_{i=0}^{L-1} S_i$, $0 \in \mathring{S}_0$, we construct next a PWA function $V : \Omega_{\mathcal{E}} \to \mathbb{R}$ such that $V(Ax + B\hat{u}(x)) \leq \lambda V(x)$, for all $x \in \Omega_{\mathcal{E}}$, with $1 > \lambda > \mu$. Let $\{x \in \mathbb{R}^n : G_{\mathcal{E}}x \leq \mathbb{1}\}$ be the normalized representation of $\Omega_{\mathcal{E}}$ with $G_{\mathcal{E}} \in \mathbb{R}^{m_{\mathcal{E}} \times n}$, and let $R_k^{\mathcal{E}} \triangleq \{x \in \mathbb{R}^n : G_{\mathcal{E}}^{(h)}x \leq G_{\mathcal{E}}^{(k)}x \leq 1, \forall h = 1, \ldots, m_{\mathcal{E}}\}$, $k = 1, \ldots, m_{\mathcal{E}}$ (see Fig. 5). Consider a polyhedral partition of the nonconvex set $S_0 \setminus \Omega$ into convex polyhedral sets $\bar{C}_{jk}$, $j = 1, \ldots, n_\Omega$, $k = 1, \ldots, m_{\mathcal{E}}$, in accordance with [21, Theorem 3]:

$$\bar{C}_{jk} = \Big\{x \in \mathbb{R}^n : \hat{H}_0 x \leq \hat{K}_0,\ G_0^{(j)}x \geq 1,\ G_0^{(h)}x \leq 1, \\ \forall h = 1, \ldots, j-1\Big\} \cap R_k^{\mathcal{E}} \quad (35)$$

for all $j = 1, \ldots, n_\Omega$, $k = 1, \ldots, m_{\mathcal{E}}$. Let $\bar{L}_0$ be the number of polytopes $\bar{C}_{jk}$ defined in (35) having a nonempty interior, and split each $\bar{C}_{jk}$ further into simplices by Delaunay triangulation [50]. Denote by C_j, $j = 1, \ldots, L_0$, $L_0 \geq \bar{L}_0$, the resulting simplices, $\bigcup_{j=1}^{L_0} C_j = S_0 \setminus \Omega$.

Consider a polyhedral partition of $S \setminus S_0$ given by the intersection of the simplicial partition and the regions $R_k^{\mathcal{E}}$, $S_{ik} = S_i \cap R_k^{\mathcal{E}}$, $i = 1, \ldots, L$, $k = 1, \ldots, m_{\mathcal{E}}$. Let $\bar{L}_{\mathcal{E}}$ be the number of polytopes S_{ik} having a nonempty interior, and split each S_{ik} further into simplices by Delaunay triangulation. Denote by $S_j^{\mathcal{E}}$, $j = 1, \ldots, L_{\mathcal{E}}$, $L_{\mathcal{E}} \geq \bar{L}_{\mathcal{E}} \geq L$, the resulting simplices. Clearly, $S = (\bigcup_{i=1}^{L_{\mathcal{E}}} S_i^{\mathcal{E}}) \bigcup (\bigcup_{j=1}^{L_0} C_j) \bigcup (\bigcup_{h=1}^{n_\Omega} R_h)$. Denote by $\{x : H_S x \leq K_S\}$ the polyhedral representation of the box S, $H_S \in \mathbb{R}^{2n \times n}$, $K_S \in \mathbb{R}^{2n}$.

Similarly, consider a partition of $\Omega_{\mathcal{E}} \setminus S$ into the polyhedral cells

$$\bar{C}_{jk}^{\mathcal{E}} = \Big\{x \in \mathbb{R}^n : H_{\mathcal{E}}x \leq K_{\mathcal{E}},\ H_S^{(j)}x \geq K_{S_j}, \\ H_S^{(h)}x \leq K_{S_h}, \forall h = 1, \ldots, j-1\Big\} \cap R_k^{\mathcal{E}}$$

$j = 1, \ldots, 2n$, $k = 1, \ldots, m_{\mathcal{E}}$. Then, split each $\bar{C}_{jk}^{\mathcal{E}}$ further into simplices by Delaunay triangulation and denote by $C_1^{\mathcal{E}}, \ldots, C_{n_{\mathcal{E}}}^{\mathcal{E}}$ the collection of the resulting simplices (see Fig. 5). Let $n_t \triangleq L_{\mathcal{E}} + L_0 + n_\Omega + n_{\mathcal{E}}$ be the total number of generated simplices, $S_{L_{\mathcal{E}}+h}^{\mathcal{E}} \triangleq C_h^{\mathcal{E}}$, $h = 1, \ldots, n_{\mathcal{E}}$, $S_{L_{\mathcal{E}}+n_{\mathcal{E}}+j}^{\mathcal{E}} \triangleq C_j$, $j = 1, \ldots, L_0$, $S_{L_{\mathcal{E}}+n_{\mathcal{E}}+L_0+h}^{\mathcal{E}} \triangleq R_h$, $h = 1, \ldots, n_\Omega$, so that $\Omega_{\mathcal{E}} = \bigcup_{i=1}^{n_t} S_i^{\mathcal{E}}$. Let the function $V : \Omega_{\mathcal{E}} \to \mathbb{R}$ be defined as

$$V(x) = \max_{i \in I_s(x)} \{z_i' x + y_i\} \quad (36)$$

where $z_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, $i = 1, \ldots, n_t$, are coefficients to be determined, and $I_s(x) = \{i \in \{1, \ldots, n_t\} : x \in S_i^{\mathcal{E}}\}$. Note that $I_s(x)$ is a singleton for all x belonging to the interior of one of the simplices $S_i^{\mathcal{E}}$; the max operation in (36) allows a unique definition of $V(x)$ in case x is on the boundary of multiple simplices and $z_i'x + y_i \neq z_j'x + y_j$ for some $i, j \in I_s(x)$, $i \neq j$, that is a discontinuity of the overall piecewise affine mapping V. Let $\{x_i^h\}_{h=0}^n$ be the vertex representation of $S_i^{\mathcal{E}}$, $i = 1, \ldots, n_t$. Denote by $I_r(i)$ the set of indices $j = 1, \ldots, n_t$ such that $S_j^{\mathcal{E}}$ is reachable from $S_i^{\mathcal{E}}$ (cf. [40]): an index $j \in \{1, \ldots, n_t\}$ belongs to $I_r(i)$ if the polytope $P_{ij} \triangleq \{x \in \mathbb{R}^n : x \in S_i^{\mathcal{E}}, A_i x + b_i \in S_j^{\mathcal{E}}\}$, is nonempty. Note that P_{ij} is the subset of $S_i^{\mathcal{E}}$ containing the states x whose update $A_i x + b_i$ maps completely into $S_j^{\mathcal{E}}$, and obviously $S_i^{\mathcal{E}} = \bigcup_{j \in I_r(i)} P_{ij}$ because of the invariance condition (16). Let x_{ij}^h, $h = 1, \ldots, n_{ij}$ denote the vertices of P_{ij}.

The closed-loop state-update function under the approximate MPC control law $\hat{u}(x)$ will be denoted as the following PWA function $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^n$

$$\mathcal{A}(x) \triangleq A_i x + b_i \\ \triangleq \begin{cases} (A + B\hat{F}_j)x + B\hat{g}_j, & \text{if } x \in S_i^{\mathcal{E}},\ i = 1, \ldots, L_{\mathcal{E}},\ j \text{ such that } S_i^{\mathcal{E}} \subseteq S_j \\ (A + BF_{\mathcal{E}})x, & \text{if } x \in S_i^{\mathcal{E}},\ i = L_{\mathcal{E}} + 1, \ldots, L_{\mathcal{E}} + n_{\mathcal{E}} \\ (A + BF_0)x, & \text{if } x \in S_i^{\mathcal{E}},\ i = L_{\mathcal{E}} + n_{\mathcal{E}} + 1, \ldots, n_t. \end{cases} \quad (37)$$

Note that $b_i = 0$ for $i = L_{\mathcal{E}} + 1, \ldots, n_t$ since $\hat{u}(x)$ is linear for $x \in S_0$, i.e., for $x \in S_i^{\mathcal{E}}$, $i = L_{\mathcal{E}} + n_{\mathcal{E}} + 1, \ldots, n_t$, and for $x \in \Omega_{\mathcal{E}} \setminus S$, i.e., for $x \in S_i^{\mathcal{E}}$, $i = L_{\mathcal{E}} + 1, \ldots, n_{\mathcal{E}}$.

Lemma 8: Let z, y, q be obtained by solving the following LP feasibility test

$$\min_{z,y,q} \quad 0$$

$$\text{s.t.} \quad z_i' x_i^h + y_i \geq q, \; \forall i \in \{1, \ldots, n_t - n_\Omega\}, \; \forall h = 0, \ldots, n \tag{38a}$$

$$z_j' \left(A_i x_{ij}^h + b_i\right) + y_j \leq \lambda_1 \left(z_i' x_{ij}^h + y_i\right), \; \forall i \in \{1, \ldots, n_t - n_\Omega\}, \; \forall j \in I_r(i), \; \forall h = 0, \ldots, n_{ij} \tag{38b}$$

$$z_i = G_0^{(i-n_t+n_\Omega)}, \; y_i = 0, \; \forall i \in \{n_t - n_\Omega + 1, \ldots, n_t\} \tag{38c}$$

$$q > 0 \tag{38d}$$

for a fixed $0 < \lambda_1 < 1$. Then the function $V : \Omega_\mathcal{E} \to \mathbb{R}$ in (36) is such that

$$V\left(Ax + B\hat{u}(x)\right) \leq \lambda_2 V(x), \; \forall x \in \Omega_\mathcal{E}$$
$$V(x) \geq q, \; \forall x \in \Omega_\mathcal{E} \setminus \Omega, \; V(x) \geq 0, \; \forall x \in \Omega$$

where $\lambda_2 = \max\{\lambda, \lambda_1\} < 1$.

Proof: Consider the set $\Omega^\sharp$ of the states x such that $V(x) = V_0(x)$

$$\begin{aligned}\Omega^\sharp &= \{x \in \Omega : I_s(x) \subseteq \{n_t - n_\Omega + 1, \ldots, n_t\}\} \\ &= \mathring{\Omega} \cup \left\{x \in \partial\Omega : x \notin S_i^\mathcal{E}, \; \forall i \in \{1, \ldots, n_t - n_\Omega\}\right\}\end{aligned}$$

Let $x \in \Omega^\sharp$. By Lemma 5, we get

$$V_0(A_0 x) = \max_{i=1,\ldots,n_\Omega} \left\{G_0^{(i)} A_0 x\right\} \leq \lambda V_0(x) \tag{39}$$

and therefore that $V(Ax + B\hat{u}(x)) = V_0(A_0 x) \leq \lambda V_0(x) = \lambda V(x) \leq \lambda_2 V(x)$.

Let $x \in S \setminus \Omega^\sharp$. Then $x \in S_i^\mathcal{E}$ for all $i \in I_s(x)$, where $I_s(x) \bigcap \{1, \ldots, n_t - n_\Omega\} \neq \emptyset$. Also, for all $i \in I_s(x)$ there exists $j \in I_r(i)$ such that $x \in P_{ij}$, i.e., $x = \sum_{h=0}^{n_{ij}} \mu_h x_{ij}^h$, $\mu_h \geq 0$, $\sum_{h=0}^{n_{ij}} \mu_h = 1$. Condition (38a) implies that

$$\begin{aligned}V(x) &= \max_{i \in I_s(x)} \{z_i' x + y_i\} \\ &= \max_{i \in I_s(x)} \left\{z_i' \left(\sum_{h=0}^{n} \mu_h x_i^h\right) + y_i \sum_{h=0}^{n} \mu_h\right\} \\ &= \max_{i \in I_s(x)} \left\{\sum_{h=0}^{n} \mu_h \left(z_i' x_i^h + y_i\right)\right\} \\ &\geq q \sum_{h=0}^{n} \mu_h = q.\end{aligned}$$

Similarly, condition (38b) implies that

$$\begin{aligned}z_j' \left(Ax + B\hat{u}(x)\right) + y_j &= z_j' \left(A_i \left(\sum_{h=0}^{n_{ij}} \mu_h x_{ij}^h\right) + b_i\right) + y_j \\ &= \sum_{h=0}^{n_{ij}} \mu_h \left(z_j' \left(A_i x_{ij}^h + b_i\right) + y_j\right) \\ &\leq \sum_{h=0}^{n_{ij}} \mu_h \left(\lambda_1 \left(z_i' x_{ij}^h + y_i\right)\right) \\ &= \lambda_1 \left(z_i' \left(\sum_{h=0}^{n_{ij}} \mu_h x_{ij}^h\right) + y_i\right) \\ &= \lambda_1 \left(z_i' x + y_i\right)\end{aligned}$$

and by (38a), it also follows that

$$z_j' \left(Ax + B\hat{u}(x)\right) + y_j \leq \lambda_2 \left(z_i' x + y_i\right)$$

Because of invariance of Ω, for all $i \in I_s(x) \cap \{n_t - n_\Omega + 1, \ldots, n_t\}$, by (39) it also follows that $z_j'(Ax + B\hat{u}(x)) + y_j \leq \lambda(z_i' x + y_i) \leq \lambda_2(z_i' x + y_i)$, $\forall j \in I_r(i) \subseteq \{n_t - n_\Omega + 1, \ldots, n_t\}$. Hence, $V(Ax + B\hat{u}(x)) \leq \max_{i \in I_s(x)}\{\lambda_2(z_i' x + y_i)\} = \lambda_2 V(x)$. □

Note that (38b) is not explicitly enforced for $i \in \{n_t - n_\Omega + 1, \ldots, n_t\}$ as $S_i^\mathcal{E} = R_{i-n_t+n_\Omega} \subseteq \Omega$ and Ω is λ-contractive. Note also that the max in (36) could be eliminated (and the above proof simplified) by imposing in (38) also the continuity of $V(x)$ on the common vertices of neighboring simplices; this however would reduce the number of degrees of freedom in determining function V by (38).

D. Stability Result

Theorem 1: If problem (38) admits a feasible solution, then the feedback control law (20) asymptotically stabilizes (1) with domain of attraction $\Omega_\mathcal{E}$, and $E_u \hat{u}(x_k) \leq G_u$, for all $k \geq 0$, for all $x(0) \in \Omega_\mathcal{E}$.

Proof: Invariance of $\Omega_\mathcal{E}$ is proved by λ-contractiveness of $\Omega_\mathcal{E}$ under the backup control law $u = F_\mathcal{E} x$, and by (16) when $x \in S$ under the control law $\hat{u}(x)$. Hence, $V(x_k)$ is defined at all time instant $k \geq 0$, for any initial condition $x_0 \in \Omega_\mathcal{E}$. By Lemma 8, $V(x_k)$ contracts by a factor $\lambda_2 < 1$ at each step $k \in \mathbb{N}$. Then, there exists a time k_1 such that $V_s(x_{k_1})) \leq \lambda_2^{k_1} V_s(x_0) < q$, which implies $x_{k_1} \in \Omega$. By λ-contractiveness of Ω, $x_k \in \Omega$, $\forall k \geq k_1$, and $x_k \to 0$ for $k \to \infty$ under the control law $u = F_0 x$. Note that $x = 0$ is an equilibrium point, as $\hat{u}(0) = F_0 0 = 0$. As the origin is locally asymptotically stable and all trajectories starting in $\Omega_\mathcal{E}$ converge to the origin, the origin is also asymptotically stable in large, with domain of attraction $\Omega_\mathcal{E}$. □

VI. Numerical and Implementation Results

The proposed approximation method was tested on two different examples. The MPC controllers, their explicit representations, and (18) and (19), were solved in MATLAB using the Hybrid Toolbox [23], YALMIP [51], CVX [52], GLPK [53], and the Optimization Toolbox of MATLAB. All computations were performed on a 3-GHz Pentium 4 PC with 3.25 GB of RAM.

In the examples both the optimal MPC PWA laws and the suboptimal PWAS laws were implemented on a Xilinx Spartan 3 FPGA (xc3s200) board. The PWA laws are implemented using the architecture based on a binary search tree proposed in [25], whereas the PWAS laws are realized by resorting to architectures A and B in [19].

The performance measure

$$\mathcal{Q}(x_0) = \sum_{k=0}^{T} x_k' Q x_k + u_k' R u_k \tag{40}$$

is used to compare the trajectories obtained by applying different control laws, where $x_0 \in S$ is a given initial condition and Q and R are the same as in (2a). Performance is compared by evaluating the average $\bar{\mathcal{Q}}$ of $\mathcal{Q}$ on a set of initial conditions x_0.

A. MIMO System

Consider the problem of regulating the discrete-time unstable multivariable system

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1.2 & 1 \\ 0 & 1.1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} u_k \\ y_k &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} x_k \end{aligned} \tag{41}$$

to the origin while minimizing (2a) with

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix},\ Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ N = 5,\ N_u = 5,\ \rho = 10^4$$

and P solving the Riccati equation associated with A, B, Q, R, in the presence of the hard constraint $[-0.5\ \ -0.6]' \leq u_k \leq [0.5\ 0.6]'$ on the inputs and the soft constraint $-2 - \epsilon \leq y_{k,1}, y_{k,2} \leq 2 + \epsilon$ on both outputs; these correspond to setting

$$E_u = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},\ G_u = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.6 \\ 0.6 \end{bmatrix},\ E_x = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$F_x = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},\ G_x = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix},\ V_x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

in (2).

1) PWA and PWAS Laws: The resulting explicit MPC state feedback u is a PWA vector function defined over 52 partitions. In problems (18) and (19) we set $p_1 = p_2 = 15$ and $\tau = 1000$, obtaining two PWAS approximations in 21 s and 14 s, respectively. The maximum error M in (22) between the explicit MPC control and its L^2 and L^∞ approximations is 0.095 in both cases.

Fig. 6(a) shows the trajectories of the state variables of the MIMO system (41), starting from the initial condition $x_0 = [-1.3\ -1]'$ under different control laws (exact MPC, L^2- and L^∞-PWAS approximations). Fig. 6(b) shows the input and output signals of the controlled system. Starting from x_0, the

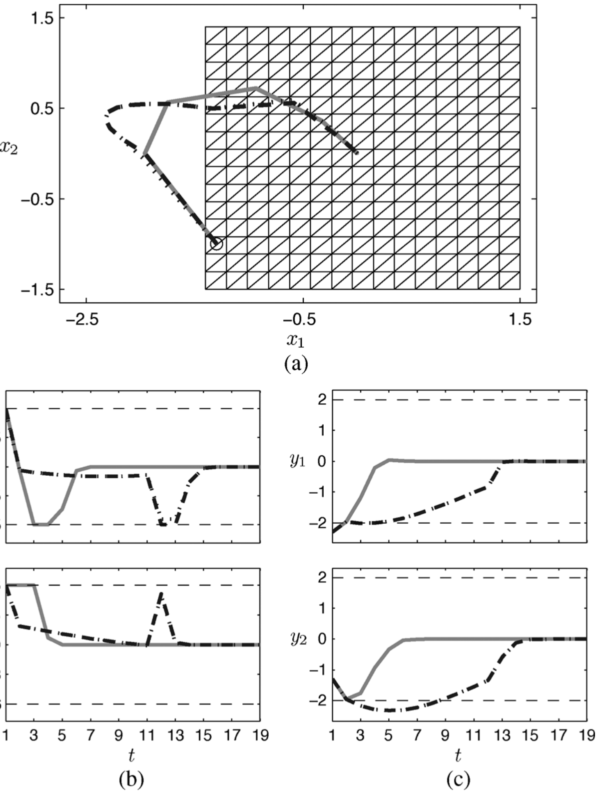

Fig. 6. Trajectories of MIMO system (41) under different controls: MPC PWA control (gray solid line), L^2 PWAS control (dashed line), and L^∞ PWAS control (dotted line). (a) State trajectories. (b) Input signals u_1 and u_2. (c) Output signals y_1 and y_2.

trajectories of the closed-loop system under the approximated control step outside S and are brought back inside S by the back-up control law $F_{\mathcal{E}}$. Since the back-up control law is not derived as an approximation of the optimal control u^*, the trajectories follow very different paths outside S. Note also that because of (14), the exact and PWAS approximate control laws coincide near the origin.

Table I shows the average performance measure $\bar{\mathcal{Q}}$ of exact MPC, obtained with $T = 20$ starting from a set of 256 initial conditions uniformly distributed over S, and its percentage degradation when approximating the control law. The table reports the values obtained by considering all the trajectories, only the trajectories that never leave S and the trajectories that have at least one point outside S. It is apparent that the main differences are due to the trajectories that leave S, since outside S the backup gain F_ϵ is used to control the system.

2) Stability Analysis: Problem (38) was solved for the L^2 and L^∞ approximations with $\lambda = 0.85$ and $\lambda_1 = 0.95$. The resulting Lyapunov functions, defined over the invariant polytope Ω_ϵ are shown in Fig. 7.

3) Circuit Implementations of the Control Laws: The state variables (circuit inputs) are coded with 12 bits words and the PWAS controls are implemented with the digital architectures A (smaller) and B (faster) proposed in [19]. For $p = p_1 = p_2 = 15$ by using architecture A the control move is calculated in 210 ns and 13% of slices is occupied. By using architecture B, it takes 43 ns and 21% of occupied slices. If the optimal control is implemented using the architecture based on a binary search tree, the maximum and average times needed to evaluate the

TABLE I
DEGRADATION OF AVERAGE PERFORMANCE MEASURE FOR SYSTEM (41) WITH RESPECT TO EXACT MPC PERFORMANCE $\bar{\mathcal{Q}}$ AND FOR DIFFERENT COARSENESS OF THE SIMPLICIAL PARTITION. THE SYMBOL "$\in S$" INDICATES THE TRAJECTORIES THAT NEVER LEAVE S, "$\notin S$" THE TRAJECTORIES THAT INSTEAD EXIT S, AND "$\% \in S$" THE PERCENTAGE OF TRAJECTORIES FULLY CONTAINED IN S ($p = p_1 = p_2$)

Control	p	all	$\in S$	$\notin S$	$\% \in S$
MPC ($\bar{\mathcal{Q}}$)		3.757	2.104	12.618	83.6%
	7	+219%	+1.24%	+403%	83.6%
L^2 / MPC	15	+217%	+0.11%	+399%	83.6%
	31	+217%	+0.00%	+400%	83.6%
	63	+217%	+0.00%	+399%	83.6%
	7	+247%	+9.33%	+418%	80.8%
L^∞ / MPC	15	+236%	+5.93%	+416%	82.4%
	31	+230%	+3.29%	+407%	82.4%
	63	+221%	+1.66%	+401%	83.2%

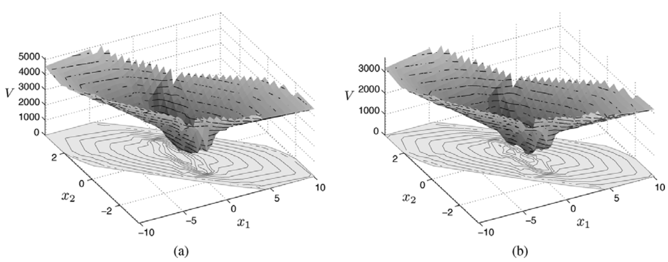

Fig. 7. Lyapunov functions for system (41) under approximate MPC control. Although not evident from the plot, continuity of V is not enforced. (a) L^2 PWAS control. (b) L^∞ PWAS control.

TABLE II
CIRCUIT PERFORMANCE FOR SYSTEM (41) WITH RESPECT TO THE COARSENESS OF THE SIMPLICIAL PARTITION ($p = p_1 = p_2$)

Control	p	CPU time	M	%occupied slices	Latency [ns]
	7	12 s	0.1818	A:10%, B:8%	A:170, B:31
L^2	15	21 s	0.095	A:13%, B:21%	A:210, B:43
	31	36 s	0.0461	A:24%, B:55%	A:238, B:45
	63	170 s	0.0229	A:61%, B:177%	A:272, B:46
	7	11 s	0.1679	A:10%, B:8%	A:170, B:31
L^∞	15	14 s	0.095	A:13%, B:21%	A:210, B:43
	31	34 s	0.0461	A:24%, B:55%	A:238, B:45
	63	160 s	0.0227	A:61%, B:177%	A:272, B:46

control are 486 ns and 383 ns, respectively, and the percentage of occupied slices is 25%.

Table II shows the circuit performance with respect to the coarseness of the simplicial partition, where the symbols A and B refer to the chosen architectures. The data reported in Tables I and II highlight that a finer partition provides a better approximation, at the cost of a higher computational effort and of a more complex circuit implementation.

B. Three-Dimensional System

Consider the system

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k \\
&= \begin{bmatrix} 1.1 & 0 & 1.5 \\ 0 & 0.5 & 0.1 \\ 0 & 0 & 1.2 \end{bmatrix} x_k + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u_k \\
y_k &= [1 \quad -1 \quad -2] x_k
\end{aligned} \tag{42}
$$

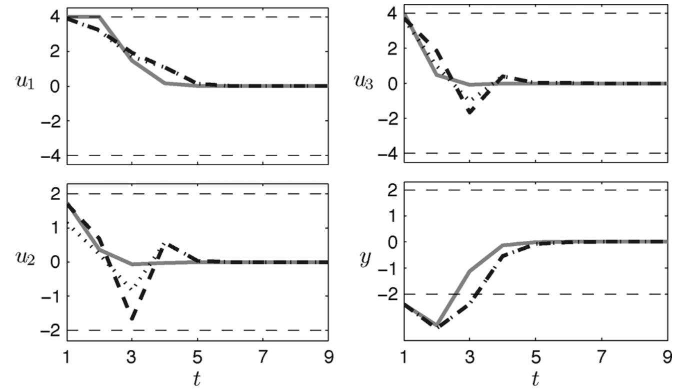

Fig. 8. Input and output signals of system (42) under different controls: MPC PWA control (gray solid lines), L^2 PWAS control (dashed lines), and L^∞ PWAS control (dotted lines).

We want to regulate the system to the origin while minimizing the quadratic performance measure (2a) with

$$
R = 0.1,\ Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ N = 7,\ N_u = 7,\ \rho = 10^4
$$

and P equal to the solution of the Riccati equation associated with A, B, Q, R, while fulfilling the hard constraints $[-1.3\ -0.65\ -1.3]' \leq u(k) \leq [1.3\ 0.65\ 1.3]'$ and the soft constraints $-3 - \epsilon \leq y(k) \leq 3 + \epsilon$, that map into the linear constraints (2d), (2e), (2f) with

$$
E_u = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix},\ G_u = \begin{bmatrix} 1.3 \\ 1.3 \\ 0.65 \\ 0.65 \\ 1.3 \\ 1.3 \end{bmatrix},\ E_x = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},
$$

$$
F_x = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix},\ G_x = \begin{bmatrix} 3 \\ 3 \end{bmatrix},\ V_x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}
$$

1) PWA and PWAS Laws: The resulting optimal explicit control $u^* : \mathbb{R}^3 \to \mathbb{R}^3$ is a PWA vector function defined over 454 partitions. By solving problems (18) and (19) with $p_1 = p_2 = p_3 = 7$ and $\tau = 1000$ we obtained two PWAS control functions in 485 s and 273 s, respectively. The maximum error M in this case is 1.09 for the L^2 approximation, and 1.74 for the L^∞ approximation. Fig. 8 shows the input and output signals of system (42) obtained by applying the optimal MPC law, its PWAS L^2-approximation, and its PWAS L^∞-approximation, starting from the initial condition $x_0 = [-1.4\ 1.4\ -1.6]'$. Table III shows the average performance measure of exact MPC, obtained with $T = 15$ starting from a set of 4096 initial conditions uniformly distributed over S, and its percentage degradation under approximated MPC. Also in this case the main differences are due to the trajectories that leave S.

2) Stability Analysis: The linear programming problem (38) was solved successfully for the L^2 and the L^∞ approximations with $\lambda = 0.86$ and $\lambda_1 = 0.99$, therefore certifying closed-loop stability in both cases.

3) Circuit Implementations of the Control Laws: The state variables (circuit inputs) are coded with 12 bits words and the PWAS controls are implemented with the digital architectures A

TABLE III
DEGRADATION OF AVERAGE PERFORMANCE MEASURE FOR SYSTEM (42) WITH RESPECT TO EXACT MPC. "$\in S$" INDICATES THE TRAJECTORIES THAT NEVER LEAVE S, "$\notin S$" THE TRAJECTORIES THAT EXIT S, AND "$\% \in S$" THE PERCENTAGE OF TRAJECTORIES FULLY CONTAINED IN S

Control	all	$\in S$	$\notin S$	$\% \in S$
MPC ($\bar{Q}$)	108.3	11.3	384.6	74%
L^2 / MPC	+134%	+137%	+102%	74%
L^∞ / MPC	+142%	+141%	+151%	67%

and B proposed in [19]. By using architecture A, a control move is calculated in 288 ns and 20% of slices is occupied. By using architecture B, it takes 76 ns and 46% of slices are occupied. In this case, the optimal control cannot be implemented on the selected FPGA due to the complexity of the resulting circuit.

The above examples point out a fundamental aspect of PWAS approximations: their circuit implementation does not depend directly on the parameters of the MPC problem (Q, R, N, N_u, G_u, G_x, etc.) like the optimal solution, but the designer can set their performances in terms of latency and area occupation by changing the number of partitions along each dimension and the numerical accuracy, looking for a tradeoff between accuracy and circuit specifications.

VII. CONCLUSION

This paper has proposed a novel approach to the approximation of MPC controllers using canonical PWA functions, and provided techniques to prove their stabilization properties. Compared to other function approximation methods, the resulting approximation can be implemented on chip in an extremely fast way. The main limitation of the approach is the "curse of dimensionality" due to the simplicial partitioning of the set of states where the control law is approximated. A way to mitigate this effect is currently under investigation by looking at the construction of decentralized approximate MPC solutions, each one defined in small dimensional spaces. Future research will also address the extension of the ideas of this paper to nonlinear and hybrid MPC settings, and to provide robust stability certifications in the presence of uncertainties affecting the system.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their very useful comments on the original manuscript.

REFERENCES

[1] E. Camacho and C. Bordons, *Model Predictive Control*, ser. Advanced Textbooks in Control and Signal Processing, 2nd ed. London, U.K.: Springer-Verlag, 2004.

[2] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear Model Predictive Control: Towards New Challenging Applications*, ser. Lecture Notes in Control and Information Sciences, D. R. L. Magni and F. Allgower, Eds. Berlin, Heidelberg, Germany: Springer-Verlag, 2009, vol. 384, pp. 345–369.

[3] S. Di Cairano, D. Yanakiev, A. Bemporad, I. Kolmanovsky, and D. Hrovat, "An MPC design flow for automotive control and applications to idle speed regulation," in *Proc. 47th IEEE Conf. Decision Control*, Cancun, Mexico, 2008, pp. 5686–5691.

[4] A. Beccuti, G. Papafotiou, R. Frasca, and M. Morari, "Explicit hybrid model predictive control of the DC-DC boost converter," in *Proc. IEEE PESC*, Orlando, FL, Jun. 2007, pp. 2503–2509.

[5] A. Bemporad and C. Filippi, "Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming," *J. Optim. Theory Applicat.*, vol. 117, no. 1, pp. 9–38, Apr. 2003.

[6] C. Jones, M. Baric, and M. Morari, "Multiparametric linear programming with applications to control," *Eur. J. Control*, vol. 13, pp. 152–170, 2007.

[7] T. A. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal search tree," *IEEE Trans. Autom. Control*, vol. 58, no. 5, pp. 810–815, May 2003.

[8] T. A. Johansen, "Approximate explicit receding horizon control of constrained nonlinear systems," *Automatica*, vol. 40, pp. 293–300, 2004.

[9] D. Muñoz de la Peña, A. Bemporad, and C. Filippi, "Robust explicit MPC based on approximate multi-parametric convex programming," *IEEE Trans. Autom. Control*, vol. 51, no. 8, pp. 1399–1403, Aug. 2006.

[10] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," in *Proc. 42th IEEE Conf. Decision Control*, Maui, HI, 2003, pp. 3179–3184.

[11] F. Christophersen, M. Zeilinger, C. Jones, and M. Morari, "Controller complexity reduction for piecewise affine systems through safe region elimination," in *Proc. IEEE Conf. Decision Control*, New Orleans, LA, 2007, pp. 4773–4778.

[12] T. A. Johansen, W. Jackson, R. Schrieber, and P. Tøndel, "Hardware synthesis of explicit model predictive controllers," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 1, pp. 191–197, Jan. 2007.

[13] S. Mariéthoz, U. Mäder, and M. Morari, "High-speed FPGA implementation of observers and explicit model predictive controllers," in *Proc. IEEE IECON, Ind. Electron. Conf.*, Porto, Portugal, Nov. 2009.

[14] T. Parisini and R. Zoppoli, "A receding-horizon regulator for nonlinear systems and a neural approximation," *Automatica*, vol. 31, no. 10, pp. 1443–1451, 1995.

[15] M. Canale, L. Fagiano, and M. Milanese, "Set membership approximation theory for fast implementation of model predictive control," *Automatica*, vol. 45, no. 1, pp. 45–54, 2009.

[16] P. Julián, A. Desages, and O. Agamennoni, "High-level canonical piecewise linear representation using a simplicial partition," *IEEE Trans. Circuits Syst. I*, vol. 46, no. 4, pp. 463–480, Apr. 1999.

[17] P. Julián, A. Desages, and B. D'Amico, "Orthonormal high level canonical PWL functions with applications to model reduction," *IEEE Trans. Circuits Syst. I*, vol. 47, no. 5, pp. 702–712, May 2000.

[18] M. Storace, L. Repetto, and M. Parodi, "A method for the approximate synthesis of cellular non-linear networks—Part 1: Circuit definition," *Int. J. Circuit Theory Applicat.*, vol. 31, no. 3, pp. 277–297, May–Jun. 2003.

[19] M. Storace and T. Poggi, "Digital architectures realizing piecewise-linear multi-variate functions: Two FPGA implementations," *Int. J. Circuit Theory Applicat.*, vol. 39, pp. 1–15, 2011.

[20] M. Di Federico, T. Poggi, P. Julian, and M. Storace, "Integrated circuit implementation of multi-dimensional piecewise-linear functions," *Digital Signal Process.*, vol. 20, pp. 1723–1732, 2010, 10.1016/j.dsp.2010.02.007.

[21] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[22] A. Bemporad, M. Morari, and N. Ricker, "Model Predictive Control Toolbox for Matlab—User's Guide," The Mathworks, Inc., 2004 [Online]. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/

[23] A. Bemporad, "Hybrid Toolbox—User's Guide," Jan. 2004 [Online]. Available: http://www.dii.unisi.it/hybrid/toolbox

[24] A. Bemporad, "Model-based predictive control design: New trends and tools," in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, 2006, pp. 6678–6683.

[25] A. Oliveri, A. Oliveri, T. Poggi, and M. Storace, "Circuit implementation of piecewise-affine functions based on a binary search tree," in *Proc. ECCTD'09*, Antalya, Turkey, Aug. 2009, pp. 145–148.

[26] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, May 2003.

[27] C. Filippi, "An algorithm for approximate multiparametric linear programming," *J. Optim. Theory Applicat.*, vol. 120, no. 1, pp. 73–95, 2004.

[28] L. Repetto, M. Storace, and M. Parodi, "A method for the approximate synthesis of cellular non-linear networks—Part 2: Circuit reduction," *Int. J. Circuit Theory Applicat.*, vol. 31, no. 3, pp. 299–313, May–Jun. 2003.

[29] H. Kuhn, "Some combinatorial lemmas in topology," *IBM J. Res. Develop.*, vol. 4, pp. 518–524, 1960.
[30] H. Kuhn, "Simplicial approximation of fixed points," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 61, pp. 1238–1242, 1968.
[31] M. Chien and E. Kuh, "Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision," *IEEE Trans. Circuits Syst.*, vol. CAS-24, no. 6, pp. 305–317, Jun. 1977.
[32] R. Rovatti, M. Borgatti, and R. Guerrieri, "A geometric approach to maximum-speed n-dimensional continuous linear interpolation in rectangular grids," *IEEE Trans. Comput.*, vol. 47, no. 8, pp. 894–899, Aug. 1998.
[33] R. Rovatti, C. Fantuzzi, and S. Simani, "High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems," *Signal Process.*, vol. 80, pp. 951–963, 2000.
[34] A. Boggiano, S. Delfitto, T. Poggi, and M. Storace, "FPGA implementation of a new scheme for the circuit realization of PWL functions," in *Proc. Eur. Conf. Circuit Theory Design (ECCTD'07)*, Aug. 2007, pp. 874–877.
[35] P. Echevarria, M. Martínez, J. Echanobe, I. del Campo, and J. Tarela, "Digital hardware implementation of high dimensional fuzzy systems," in *Applications of Fuzzy Sets Theory*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2007, pp. 245–252.
[36] S. Di Cairano and A. Bemporad, "Model predictive control tuning by controller matching," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 185–190, Jan. 2010.
[37] T. Poggi, F. Comaschi, and M. Storace, "Digital circuit realization of piecewise affine functions with non-uniform resolution: Theory and FPGA implementation," *IEEE Trans. Circuits Syst. II*, vol. 52, no. 2, pp. 131–135, Feb. 2010.
[38] A. Bemporad, W. Heemels, and B. D. Schutter, "On hybrid systems and closed-loop MPC systems," *IEEE Trans. Autom. Control*, vol. 47, no. 5, pp. 863–869, May 2002.
[39] M. Johannson and A. Rantzer, "Computation of piece-wise quadratic Lyapunov functions for hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 555–559, Apr. 1998.
[40] G. Ferrari-Trecate, F. Cuzzola, D. Mignone, and M. Morari, "Analysis of discrete-time piecewise affine and hybrid systems," *Automatica*, vol. 38, pp. 2139–2146, 2002.
[41] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari, "Stabilizing low complexity feedback control of constrained piecewise affine systems," *Automatica*, vol. 41, no. 10, pp. 1683–1694, 2005.
[42] M. Lazar and A. Jokić, "On infinity norms as Lyapunov functions for piecewise affine systems," in *Proc. Hybrid Syst.: Comput. Control*, Stockholm, Sweden, 2010, pp. 131–139.
[43] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Trans. Autom. Control*, vol. 39, no. 2, pp. 428–433, Feb. 1994.
[44] H. Lin and P. Antsaklis, "Synthesis of uniformly ultimate boundedness switching laws for discrete-time uncertain switched linear systems," in *Proc. 42th IEEE Conf. Decision Control*, Maui, HI, 2003, pp. 4806–4811.
[45] A. Alessio, M. Lazar, A. Bemporad, and W. Heemels, "Squaring the circle: An algorithm for generating polyhedral invariant sets from ellipsoidal ones," *Automatica*, vol. 43, pp. 2096–2103, Dec. 2007.
[46] E. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and applications of maximal output admissible sets," *IEEE Trans. Autom. Control*, vol. 36, no. 9, pp. 1008–1020, Sep. 1991.
[47] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
[48] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
[49] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge Univ. Press, 2004 [Online]. Available: http://www.stanford.edu/boyd/cvxbook.html
[50] L. Yepremyan and J. Falk, "Delaunay partitions in $\mathbb{R}^n$ applied to non-convex programs and vertex/facet enumeration problems," *Comput. Oper. Res.*, vol. 32, no. 4, pp. 793–812, 2005.
[51] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD Conf.*, Taipei, Taiwan, 2004 [Online]. Available: http://users.isy.liu.se/johanl/yalmip
[52] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," Stanford Univ., Jun. 2009 [Online]. Available: http://stanford.edu/boyd/cvx, Tech. Rep.
[53] A. Makhorin, "GLPK (GNU Linear Programming Kit) User's Guide," 2004 [Online]. Available: http://www.gnu.org/software/glpk/glpk.html

Alberto Bemporad (M'99–SM'06–F'10) received the M.S. degree in electrical engineering and the Ph.D. degree in control engineering from the University of Florence, Florence, Italy, in 1993 and 1997, respectively.

He spent the academic year 1996/1997 at the Center for Robotics and Automation, Department of Systems Science and Mathematics, Washington University, St. Louis, MO, as a visiting Researcher. From 1997 to 1999, he held a postdoctoral position at the Automatic Control Laboratory, ETH, Zurich, Switzerland, where he collaborated as a Senior Researcher from 2000 to 2002. From 1999 to 2009, he was with the Department of Information Engineering, University of Siena, Italy, becoming an Associate Professor in 2005. In 2010-2011, he was with the Department of Mechanical and Structural Engineering, University of Trento, Trento, Italy. Since 2011 he is a Full Professor at the IMT Institute for Advanced Studies Lucca, Italy. He has published more than 220 papers in the areas of model predictive control, hybrid systems, automotive control, multiparametric optimization, computational geometry, robotics, and finance. He is coauthor of the Model Predictive Control Toolbox (The Mathworks, Inc.) and author of the Hybrid Toolbox for MATLAB.

Dr. Bemporad was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 2001 to 2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society from 2002 to 2010.

Alberto Oliveri received the M.Sc. degree in electronic engineering from the University of Genoa, Genova, Italy, in September 2009. He is currently pursuing the Ph.D. degree in electrical engineering in the Department of Biophysical and Electronic Engineering, University of Genoa.

His main research interests are piecewise-affine approximation of nonlinear functions and circuit implementation of explicit model predictive control. He is also interested in bifurcation analysis and detection of invariant manifolds in nonlinear dynamical systems.

Tomaso Poggi received the M.Sc. degree in electronic engineering and the Ph.D. degree in mathematical and simulation engineering from the University of Genoa, Genova, Italy, in November 2006 and April 2010, respectively.

He has been a Research Assistant in the Department of Biophysical and Electronic Engineering, University of Genoa. His main research interests are piecewise-affine approximation/identification of nonlinear functions and systems and the circuit implementation of nonlinear circuits. He is also interested in the circuit implementation of explicit model predictive control.

Marco Storace (M'01) received the "Laurea" (M.Sc.) five-year degree (*summa cum laude*) in electronic engineering and the Ph.D. degree in electrical engineering from the University of Genoa, Genova, Italy, in March 1994 and April 1998, respectively.

Since 2004, he has been an Associate Professor in the Department of Biophysical and Electronic Engineering, University of Genoa. He was a visitor to EPFL, Lausanne, Switzerland, in 1998 and 2002. His main research interests are in the area of nonlinear circuit theory and applications, with emphasis on circuit models of nonlinear systems (e.g., hysteresis, biological neurons), methods for the piecewise-linear approximation of nonlinear systems and for the consequent circuit synthesis, and nonlinear dynamics. He is the author or coauthor of more than ninety scientific papers, more than an half of which have been published in international journals.

Dr. Storace served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II (2008–2009) and is member of the IEEE Technical Committee on Nonlinear Circuits and Systems (TC-NCAS). He is coordinator of the MOBY-DIC European project (FP7-IST-248858).