49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Synthesis of Stabilizing Model Predictive Controllers via Canonical Piecewise Affine Approximations

Alberto Bemporad†, Alberto Oliveri‡, Tomaso Poggi‡, Marco Storace‡

Abstract— **This paper proposes the use of canonical piecewise affine (PWA) functions for the approximation of linear MPC controllers over a regular simplicial partition of a given set of states of interest. Analysis tools based on the construction of PWA Lyapunov functions are provided for certifying the asymptotic stability of the resulting closed-loop system. The main advantage of the proposed controller synthesis approach is that the resulting stabilizing approximate MPC controller can be implemented on chip with sampling times in the order of tens of nanoseconds.**

I. Introduction

Model predictive control (MPC) is a popular approach in industry for designing feedback controllers for multivariable systems with constraints [1], [2]. In MPC a performance index that depends on the current state vector is repeatedly optimized on line under constraints. Although very successful in industry, especially for control of large multivariable processes, the main drawback of MPC is the need to embed an optimization solver in the controller, which complicates the control code and its verification, and asks for more intensive CPU resources compared to classical control schemes, therefore preventing the application of MPC to fast-sampling processes with cheap control units.

To overcome the aforementioned computational issues, *explicit* MPC techniques were developed during the last ten years to preprocess off line the MPC control law and convert it into a piecewise affine (PWA) law. In this way, on-line operations reduce to the evaluation of a lookup table of linear control gains. We refer the reader to [3] for a recent survey on explicit MPC.

Although successfully applied in several practical applications, especially to automotive systems [4] and power converters [5], 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. The most suitable applications for explicit MPC were shown for fast-sampling problems in the order of 1-50 ms sampling time and relatively small size (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 will of solving the multiparametric programming problem *exactly*. To simplify the complexity of explicit MPC controllers, *approximate* explicit MPC techniques were addressed recently [6]–[9]. By sacrificing the optimality of the controller (which is often not a big loss, as the performance index is usually selected by trial and error by the designer), approximate explicit MPC simplifies the control law, so that sampling frequencies can be pushed up considerably.

An alternative route to synthesize approximate explicit MPC is to treat the MPC control law as a generic nonlinear function and use general purpose off-line *function approximation* techniques. Ideas in this direction were pursued in [10] using artificial neural networks and in [11] 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 PWA functions* [12], to approximate a given linear MPC controller and impose constraints in the approximation procedure that allow analyzing closed-loop stability properties using PWA Lyapunov functions. The choice of this class relies mainly on the availability of direct circuit implementation techniques for PWA functions expressed as linear combinations of canonical PWA functions [13], [14].

This paper is organized as follows. Section II setups the linear MPC problem. In Section III we introduce the class of canonical PWA function used in Section IV for the approximation of the MPC controller. Section V deals with the analysis of stability of the closed-loop system, and numerical and experimental results are shown in Section VI.

II. Model predictive control setup

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

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

of the open-loop process, where $x(t) \in \mathbb{R}^n$ is the state vector at sampling time t, and $u(t) \in \mathbb{R}^m$ is the vector of manipulated variables. The MPC algorithm selects $u(t)$ by solving the finite-time optimal control problem

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

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

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

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

†Department of Mechanical and Structural Engineering, University of Trento, Italy. Email: bemporad@ing.unitn.it

‡Department of Biophysical and Electronic Engineering, University of Genoa, Italy. Email: marco.storace@unige.it

This work was supported by the European Commission under project FP7-INFSO-ICT-248858 "MOBY-DIC – Model-based synthesis of digital electronic circuits for embedded control"

978-1-4244-7744-9/10/$26.00 ©2010 IEEE

where N is the prediction horizon, N_u is the control horizon, $U \triangleq \left[u_0' \; \dots \; u_{N_u-1}' \right]' \in \mathbb{R}^{mN_u+1}$ 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. In (2d) E_u and G_u are matrices of appropriate dimensions defining constraints on input variables. 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. A typical instance of (2d) are saturation constraints $E_u = [I \;\; -I]'$, $G_u = [u_{\max}' \;\; u_{\min}']'$, $u_{\min} < 0 < u_{\max}$. 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 LQR gain associated with matrices Q, R and P is the corresponding Riccati matrix.

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

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

$$\text{s.t.} \quad GU \leq W \tag{3b}$$

where $U^*(x(t)) = \left[u_0'^*(x(t)) \; \dots \; u_{N-1}'^*(x(t)) \right]'$ is the optimal solution, $H = H' \succ 0$ and F, Y, G, W are matrices of appropriate dimensions [15]–[17]. The MPC control law is

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

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 $x(t+1)$.

One of the drawbacks of the MPC law (4) is the need to solve the QP problem (3) on line. An alternative approach to evaluate the MPC law (4) was proposed in [15]. Rather then 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*. 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

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

where n_r is the number of polyhedral regions $\mathcal{X}_i = \{x : H_i x \leq K_i\}$, $i = 0, \dots, n_r - 1$ defining the domain partition. In (5) we labeled as $\mathcal{X}_0$ the region corresponding to the unconstrained solution $F_0 = -[I \;\; 0 \;\; \dots \;\; 0]H^{-1}F$ of problem (3), where $H_0 = GF_0 - D$ and $K_0 = W$, with $0 \in \mathring{\mathcal{X}}_0$.[1]

The evaluation of the MPC controller (4), once put in the form (5), can be carried out by a very simple piece of control code. Experiments on the implementation of explicit MPC on field programmable gate arrays (FPGA) and application specific integrated circuits (ASIC) with sampling times around 1 μs have been recently reported in [18], [19]. In this paper we aim at pushing the sampling time in the tens of nanoseconds range by adopting a special class of PWA approximating functions that have a direct implementation counterpart on electronic circuits.

[1]The symbol $\mathring{A}$ denotes the interior of the set A.

III. PWA SIMPLICIAL FUNCTIONS

A generic PWA function belongs to $L^2[S]$, the space of Lebesgue square integrable functions, but it is of interest for modeling and circuit implementation reasons to consider the subclass of *continuous and regular* PWA functions, i.e., functions defined over a regular partition of the domain S into a set of *simplices* having the same shape. 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 notions of simplex and simplicial partition.

A. Domain partition and basis functions definition

Definition 1: Given a set of $n+1$ points $x_i^0, x_i^1, \dots, x_i^n \in \mathbb{R}^n$, called *vertices*, a *simplex* S_i in $\mathbb{R}^n$ is a convex combination of the vertices, i.e., is the set of points $S_i(x_i^0, \dots, x_i^n) = \left\{ x \in \mathbb{R}^n : \; x = \sum_{j=0}^n \alpha_j x_j^i, \; 0 \leq \alpha_j \leq 1, \; \sum_{j=0}^n \alpha_j = 1 \right\} = \text{conv}\{x_i^0, \dots, x_i^n\}$

A simplex can also be represented by the hyperplanes that define its boundary, i.e., by a set of inequalities: $S_i(x_i^0, \dots, x_i^n) = \{x : \hat{H}_i x \leq \hat{K}_i\}$. As shown in [20], matrices $\hat{H}_i$, $\hat{K}_i$ are defined directly by the inequalities

$$\begin{bmatrix} 1 & \dots & 1 \\ x_i^0 & \dots & x_i^n \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0 \tag{6}$$

that are a minimal system of linear inequalities representing S_i. In this work we exploit both vertex and hyperplane representations of S_i, $i = 0, \dots, L-1$.

The domain 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 $\frac{x_{maxj} - x_{minj}}{p_j}$, collected into the vector p. 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 by applying the algorithm described in [21]; 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 $\mathring{S}_i \cap \mathring{S}_j = \emptyset$, $\forall i, j = 0, \dots, L-1$. The resulting partition is called *simplicial partition* or type-1 triangulation and is univocally identified by the vector p.

The class of continuous functions that are affine over each simplex constitutes an N_v-dimensional linear space $PWAS_p[S] \subset PWAS[S] \subset PWA[S] \cap C^0[S]$ [12]. 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

$$\hat{u}(x) = \sum_{k=1}^{N_v} w_k \, \phi_k(x) = w' \phi(x) \tag{7}$$

The coefficients w_k, collected into the vector w, determine uniquely $\hat{u}$ for each given $x \in S$.

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

$$\begin{aligned} \hat{u}(x) &= [\hat{u}_1(x), \ldots, \hat{u}_m(x)]' = \\ &= \begin{bmatrix} \phi'(x) & 0 & \cdots & 0 \\ 0 & \phi'(x) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \phi'(x) \end{bmatrix} w = \Phi(x) w \end{aligned}$$

Even if different types of basis functions can be defined, in this work we refer to the so called α-basis [22].

Whatever basis one chooses, a PWAS function can be implemented in a digital circuit by using linear interpolators. Indeed, 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 defining the simplex the input vector belongs to. The algorithm usually adopted to locate such simplex is based on the Kuhn lemmas [21] and is optimal with respect to the number of inputs [23]. Some examples of digital circuit solutions for fast piecewise-linear interpolation can be found in [13], [24], [25].

IV. PWAS APPROXIMATIONS OF MPC

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

As shown in [13], 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)). In other words, $\hat{u}(x)$ must be a feasible control. We suppose that the simplicial partition (i.e., the vector p and then the vertex set $\mathcal{V}_s$) is fixed, thus we are looking for a (vector) function $\hat{u} \in PWAS_p[S]$ as close as possible to u^* according to some metrics. To this end, we propose a functional whose minimization leads to an approximation of u^* in $L^2[S]$. The method proposed in this paper does not require the "exact" explicit MPC control law (5) to compute the approximation.

The partition $\bigcup_{i=0}^{L-1} S_i$ of S is required to satisfy the following assumption.

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

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. Since we assume that only input constraints (2d) are present, feasibility of $\hat{u}(x)$ is simply enforced by imposing

$$E_u \Phi(v) w \leq G_u, \; \forall v \in \mathcal{V}_s \tag{8}$$

Note that constraint (8) only imposes a condition on the vertices of each simplex S_i; feasibility of the control law $\hat{u}(x)$ on the entire simplex S_i simply follows by linearity of $\hat{u}(x)$ on S_i.

Local optimality. Because of Assumption 1, the further set of constraints

$$\hat{u}(v) = \Phi(v) w = u^*(v), \; \forall \text{ vertex } v \text{ of } S_0 \tag{9}$$

imposes the optimality of $\hat{u}$ around the origin. In particular, since u^* is linear in $\mathcal{X}_0$, the constraints in (9) impose that

$$\hat{u}(x) = u^*(x), \quad \forall x \in S_0. \tag{10}$$

Invariance. The approximate control law $\hat{u}$ is only defined over the selected simplicial partition 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 further examined in Section V-B:

$$H_{\mathcal{E}} \left(Av + B\Phi(v) w\right) \leq K_{\mathcal{E}}, \; \forall v \in \mathcal{V}_s \; . \tag{11}$$

These constraints impose that for all $x \in S$, the updated state $Ax + B\hat{u}(x) \in \Omega_{\mathcal{E}}$. Roughly speaking, constraints (11) should mainly affect the resulting control law $\hat{u}(x)$ only towards the boundaries of S. Note that imposing the invariance of S (instead of the invariance of $\Omega_{\mathcal{E}}$) could result in a very stringent constraint, especially if S is large and input constraints are tight.

B. Definition of the approximation functional

The functional can be 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

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

where $||\cdot||$ denotes the Euclidean norm. Considering that $\hat{u} \in PWAS_m[S]$, $\mathcal{F}_2$ reduces to a cost function F_2:

$$\mathcal{F}_2(\hat{u}) = F_2(w) = \sum_{k=1}^{m} \int_S \left[u_k^*(x) - (w^k)' \phi(x)\right]^2 dx$$

F_2 can be expressed as

$$F_2(w) = ||Cw - d||_2^2 \tag{12}$$

where

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

$\hat{C} \in \mathbb{R}^{N_v \times N_v}$ is the square matrix of the L^2 inner products between basis functions, $[\hat{C}_{ij}] = \langle \phi_i, \phi_j \rangle$, and d^k is given by: $d_i^k = \langle u_k^*, \phi_i \rangle$.

The approximated control law is calculated by solving the following QP problem

$$\begin{aligned} \min_{w} \quad & F_2(w) \\ \text{s.t.} \quad & \text{constraints (8), (9), (11)} \end{aligned} \tag{13}$$

In this formulation 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 position of the vertices influences the structure of the PWAS function.

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

$$\hat{u}(x) = \begin{cases} w'\phi(x) & \text{if} \quad x \in S \\ F_{\mathcal{E}}x & \text{if} \quad x \notin S \end{cases} \tag{14}$$

with $S_0 \subseteq \mathcal{X}_0$, $0 \in \mathring{S}_0$, where we have added a backup gain $F_{\mathcal{E}}$ outside S, that will be defined in Section V-B.

V. Stability analysis

Because of the approximation involved in constructing the approximate PWAS controller (14), possible stability properties of the original "exact" MPC controller (4) may be lost.

The approach taken in this paper consists of three steps: (i) synthesize a piecewise linear (PWL) Lyapunov function around the origin; (ii) synthesize the backup gain $F_{\mathcal{E}}$ and the corresponding polyhedral invariant set $\Omega_{\mathcal{E}}$ containing S; (iii) synthesize a PWA Lyapunov function on $\Omega_{\mathcal{E}}$ by solving an LP problem, tailored to the special structure of controller (14) and exploiting the double description of simplices S_i which is immediately available (i.e., hyperplane and vertex representations).

A. Inner λ-contractive set

We exploit contractive polyhedral sets and Minkowski functions to synthesize a PWL Lyapunov function, following ideas in the spirit of [26].

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

A λ-contractive and finitely-generated polyhedral set $\Omega = \{x \in \mathbb{R}^n : \; H_\Omega x \leq \mathbb{1}\} \subseteq S_0$, where $\mathbb{1} \in \mathbb{R}^{n_\Omega}$ is a vector of all ones, $G_0 \in \mathbb{R}^{n_\Omega \times n}$ and $0 < \lambda < 1$, can be constructed as in [27]. The following lemma is easy to prove.

Lemma 1: The largest λ-contractive set Ω contained is S_0 is the maximum output admissible set for the linear closed-loop system $x(k+1) = \frac{1}{\lambda} A_0 x(k)$ under the constraint $x \in S_0$

$$\Omega = \{x \in \mathbb{R}^n : \; \hat{H}_0 (\frac{1}{\lambda} A_0)^k x \leq \hat{K}_0, \; \forall k \geq 0\} \tag{15}$$

We denote by $\Omega = \{x \in \mathbb{R}^n : \; G_0 x \leq \mathbb{1}\}$ the minimal representation of Ω, where $\mathbb{1} \in \mathbb{R}^{n_\Omega}$, $n_\Omega \leq \bar{n}_\Omega$, $G_0 \in \mathbb{R}^{n_\Omega \times n}$. Then, the Minkowski function $V_0 : \mathbb{R}^n \to \mathbb{R}$

$$V_0(x) = \max_{j=1,\ldots,n_\Omega} \{G_0^j x\}$$

where $()^j$ denotes the jth row, is a PWL Lyapunov function for the local closed-loop system $x(k+1) = (A + BF_0)x(k)$.

B. Outer λ-contractive polyhedral set

A polytopic λ-contractive set $\Omega_{\mathcal{E}}$ covering the given partition S under a proper feedback gain $F_{\mathcal{E}}$ is used here to define the approximate control law outside S, and therefore the dynamics of the closed-loop system outside S. A method to determine a gain $F_{\mathcal{E}}$, to derive an associated λ-contractive ellipsoid $\mathcal{E} \subset \mathbb{R}^n$ containing S, and to determine a λ-contractive polyhedron $\Omega_{\mathcal{E}}$ for the closed-loop system $x(k+1) = (A + BF_{\mathcal{E}})x(k)$ is reported in [28].

C. PWA Lyapunov function

In this section we construct a PWA function $V(x)$ such that $V(Ax + B\hat{u}(x)) \leq \lambda V(x)$, for all $x \in \Omega_{\mathcal{E}}$. Let the function $V : \Omega_{\mathcal{E}} \to \mathbb{R}$ be defined as

$$V(x) = \max_{i \in I_s(x)} \{z_i' x + y_i\} \tag{16}$$

where $z_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$ are coefficients (to be determined next), and $I_s(x) = \{i \in \{1, \ldots, n_t\} : \; x \in S_i^{\mathcal{E}}\}$, $S_i^{\mathcal{E}}$ is a properly chosen polyhedral partition of $\Omega_{\mathcal{E}}$, see [28] for details.

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. [29]).

Lemma 2: Let z, y, q be obtained by solving the following LP feasibility test

$$\begin{aligned} \min_{z,y,q} \quad & 0 \\ \text{s.t.} \quad & z_i' x_i^h + y_i \geq q, \; \forall i \in \{1, \ldots, n_t - n_\Omega\} \\ & \quad \forall h = 0, \ldots, n && \text{(17a)} \\ & z_j'(A_i x_{ij}^h + b_i) + y_j \leq \lambda_1 (z_i' x_{ij}^h + y_i) \\ & \quad\quad \forall i \in \{1, \ldots, n_t - n_\Omega\}, \; \forall j \in I_r(i) \\ & \quad \forall h = 0, \ldots, n_{ij} && \text{(17b)} \\ & z_i = G_0^{i - n_t + n_\Omega}, \; y_i = 0 \\ & \quad \forall i \in \{n_t - n_\Omega + 1, \ldots, n_t\} && \text{(17c)} \\ & q > 0 && \text{(17d)} \end{aligned}$$

with $0 < \lambda_1 < 1$. Then the function $V : \Omega_{\mathcal{E}} \to \mathbb{R}$ in (16) is such that

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

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

Proof: See [28]. ■

D. Stability result

Theorem 1: If problem (17) admits a feasible solution, then the feedback control law (14) 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: See [28]. ■

VI. An Example

We have applied the proposed approximation method to a MIMO system. The MPC controllers and their explicit representations, as well as Problem (13), have been solved in MATLAB relying on the the Hybrid Toolbox [17] and YALMIP [30]. Both the exact explicit MPC law and the suboptimal PWAS law were implemented on a Xilinx Spartan 3 FPGA (xc3s200). The exact PWA law is implemented using the architecture based on a binary search tree proposed in [19], whereas the PWAS law is realized by resorting to architectures A and B in [13].

The following metrics is used to compare the trajectories obtained by applying different control laws

$$\mathcal{Q}(x_0) = \sum_{k=0}^{T} x_k' Q x_k + u_k' R u_k \tag{18}$$

where $x_0 \in S$ is a given initial state and Q and R are the same as in (2a). Then, the performances of the tested approximations are compared by evaluating the average $\bar{\mathcal{Q}}$ of $\mathcal{Q}(x_0)$ over a number of initial conditions x_0.

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

$$\begin{aligned} x_{k+1} &= \left[\begin{smallmatrix} 1.3 & 1 \\ 0 & 1.1 \end{smallmatrix}\right] x_k + \left[\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right] u_k \\ y_k &= \left[\begin{smallmatrix} 1 & 0 \end{smallmatrix}\right] x_k \end{aligned} \tag{19}$$

while minimizing the quadratic performance measure (2a) with

$$R = \left[\begin{smallmatrix} 0.1 & 0 \\ 0 & 0.1 \end{smallmatrix}\right],\ Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right],\ N = 4,\ N_u = 4,\ \rho = 10^4$$

and P solving the Riccati equation associated with A, B, Q, R, in the presence of the the hard constraints $\left[\begin{smallmatrix} -0.5 \\ -0.7 \end{smallmatrix}\right] \le u(k) \le \left[\begin{smallmatrix} 0.5 \\ 0.7 \end{smallmatrix}\right]$; these correspond to setting

$$E_u = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},\ G_u = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.7 \\ 0.7 \end{bmatrix}$$

in (2).

1) Stability analysis: Problem (17) has been solved with $\lambda = 0.99$ and $\lambda_1 = 0.89$. The Lyapunov function for the L^2 approximation obtained by solving (17) and defined over the invariant polytope Ω_ϵ is shown in Figure 1.

2) PWA and PWAS laws: The resulting exact explicit MPC state feedback u over the domain $[-1.4,\ -1.5] \times [1.5,\ -1.4]$ is a PWA vector function defined partitioned into 50 polytopes. Problem (13) has been solved by setting $m_1 = m_2 = 15$ obtaining a PWAS control in $64s$ (on a 3 GHz Pentium 4 PC with 3.25GB of RAM).

Figure 2 (b) shows the input and output signals of the controlled system, starting from the intial condition $x_0 = [0.7\ 1]'$ under different control laws (exact MPC, L^2-PWAS approximations).

Table I shows the average performance obtained with $T = 30$ starting from a set of 256 intial conditions uniformely distribuited over S. The table reports the values obtained by considering all the trajectories, only the trajectories that never leave S (2nd column), and the trajectories that have at least one point outside S. The main differences are due to the trajectories that leave S, as outside S the backup gain F_ϵ is used to control the system. Note that for trajectories living in S, the loss of performance of the PWAS laws is quite limited, as shown in the 2nd column of Table I.

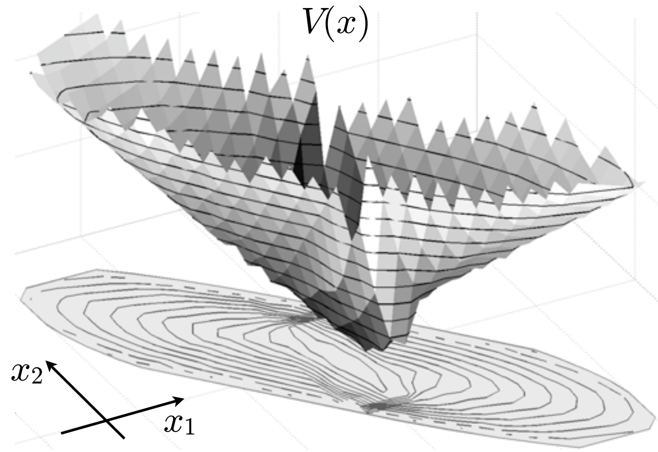

Fig. 1. Lyapunov function for the closed-loop system under the L^2 PWAS control.

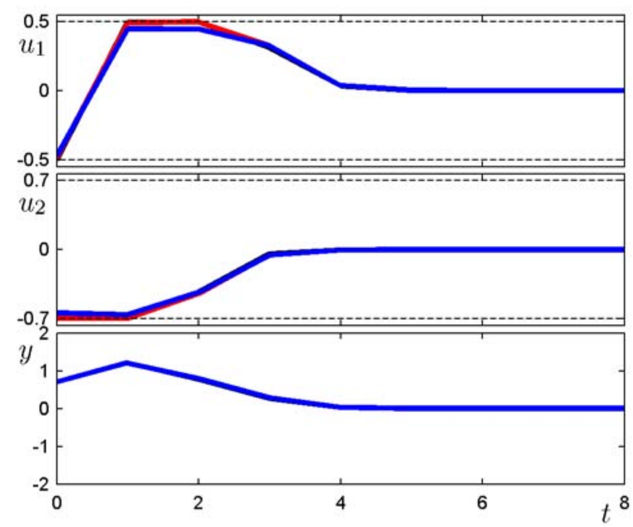

Fig. 2. Input and output signals of the MIMO system under different controls: MPC PWA control (blue line), L^2 PWAS control (red line).

Control	$\bar{\mathcal{Q}}$		
	all	$\in S$	$\notin S$
MPC	4.12	2.09	13.59
L^2	17.29	2.17	94.39

TABLE I

Mean quality factor for the exact MPC and approximated controls. "$\in S$" indicates the trajectories that never leave S, "$\notin S$" the trajectories that leave S.

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 [13]. By using architecture A, we can calculate a control move every 186 ns and 11% of slices is occupied. By using architecture

B, we can calculate a control move every 36 ns and the percentage of occupied slices is 35%. If the optimal control is implemented using the architecture based on a binary search tree, the maximum and mean times needed to evaluate the control are 530 ns and 430 ns, respectively, and the percentage of occupied slices would be 25%.

This example points 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, etc.) like the optimal solution, but their performances in terms of latency and area occupation can be set by the designer changing the number of partitions along each dimension and the numerical accuracy, looking for a trade-off between the approximation accuracy and the circuit specifications.

VII. Conclusions and Future Research

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. In particular, the storage requirements in the circuit implementing the PWA approximation grows with the number N_v of vertices of the simplicial partition (i.e., of coefficients to be stored in the memory), which increases exponentially with the number of dimensions.

Future research will 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.

References

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

[2] J. Maciejowski, *Predictive Control with Constraints*. Harlow, UK: Prentice Hall, 2002.

[3] 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, F. Allgower, Ed., vol. 384. Berlin Heidelberg: Springer-Verlag, 2009, pp. 345–369.

[4] 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. on Decision and Control*, Cancun, Mexico, 2008, pp. 5686–5691.

[5] A. Beccuti, G. Papafotiou, R. Frasca, and M. Morari, "Explicit hybrid model predictive control of the DC-DC boost converter," in *IEEE PESC*, Orlando, Florida, USA, June 2007.

[6] A. Bemporad and C. Filippi, "Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming," *Journal of Optimization Theory and Applications*, vol. 117, no. 1, pp. 9–38, Apr. 2003.

[7] C. Jones, M. Baric, and M. Morari, "Multiparametric linear programming with applications to control," *European Journal of Control*, vol. 13, pp. 152–170, 2007.

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

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

[10] 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.

[11] 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.

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

[13] M. Storace and T. Poggi, "Digital architectures realizing piecewise-linear multi-variate functions: two FPGA implementations," *International Journal of Circuit Theory and Applications*, vol. 37, 2010, in the press (doi: 10.1002/cta.610).

[14] M. Di Federico, T. Poggi, P. Julian, and M. Storace, "Integrated circuit implementation of multi-dimensional piecewise-linear functions," *Digital Signal Processing*, vol. 20, 2010.

[15] 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.

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

[17] A. Bemporad, *Hybrid Toolbox – User's Guide*, Jan. 2004, http://www.dii.unisi.it/hybrid/toolbox.

[18] T. Johansen, W. Jackson, R. Schrieber, and P. Tøndel, "Hardware synthesis of explicit model predictive controllers," *IEEE Trans. Contr. Systems Technology*, vol. 15, no. 1, pp. 191–197, 2007.

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

[20] C. Filippi, "An algorithm for approximate multiparametric linear programming," *Journal of Optimization Theory and Applications*, vol. 120, no. 1, pp. 73–95, 2004.

[21] H. Kuhn, "Some combinatorial lemmas in topology," *IBM J. of Research and Development*, vol. 4, pp. 518–524, 1960.

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

[23] R. Rovatti, M. Borgatti, and R. Guerrieri, "A geometric approach to maximum-speed n-dimensional continuous linear interpolation in rectangular grids," *IEEE Transactions on Computers*, vol. 47, no. 8, pp. 894–899, AUG 1998.

[24] R. Rovatti, C. Fantuzzi, and S. Simani, "High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems," *Signal Processing*, vol. 80, pp. 951–963, 2000.

[25] 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: Springer, 2007, pp. 245–252.

[26] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Trans. Automatic Control*, vol. 39, no. 2, pp. 428–433, Feb. 1994.

[27] 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.

[28] A. Bemporad, A. Oliveri, T. Poggi, and M. Storace, "Ultra-fast stabilizing model predictive control via canonical piecewise affine approximations," 2010, internal report, submitted for publication. Available upon request from the authors.

[29] 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.

[30] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip