PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Approximate Explicit MPC and Closed-loop Stability Analysis based on PWA Lyapunov Functions

Sergio Trimboli

Advisor:

Prof. Alberto Bemporad

IMT Institute for Advanced Studies Lucca

XXV Cycle - November 2012

Abstract

Model Predictive Control (MPC) is the de facto standard in advanced industrial automation systems. There are two main formulations of the MPC algorithm: an implicit one and an explicit MPC one. The first requires an optimization problem to be solved on-line, which is the main limitation when dealing with hard real-time applications. As the implicit MPC algorithm cannot be guaranteed in terms of execution time, in many applications the explicit MPC solution is preferable. In order to deal with systems integrating mixed logic and dynamics, the class of the hybrid and piecewise affine models (PWA) were introduced and tackled by the explicit MPC strategy. However, the resulting controller complexity leads to a requirement on the CPU/memory combination which is as strict as the number of states, inputs and outputs increases. To reduce drastically the complexity of the explicit controller while preserving the controller's performance, a strategy combining switched MPC with discontinuous simplicial PWA models is introduced in this thesis. The latter is proven to be circuit implementable, e.g., in FPGA. To ensure that closed-loop stability properties are guaranteed, a stability analysis tool is proposed which exploits suitable and possibly discontinuous PWA Lyapunov-like functions. The tool requires solving offline a linear programming problem. Moreover, the tool is able to compute an invariant set for the closed-loop system, as well as ultimate boundedness and input-to-state stability properties.

Acknowledgements

This thesis is the result of three years of intense research partially supported by the European Commission through project MOBY-DIC "Model-based synthesis of digital electronic circuits for embedded control" (FP7-INFSO-ICT-248858).
I would like to express here the pleasure I had in working together with Prof. Alberto Bemporad. He gave me the possibility to gain a lot of experience in advanced control topics. Alberto, I would like to thank you for sharing your knowledge with me. This thesis, but more important the work possibilities I had, would never happen without your support.
Also, I am grateful to all my co-authors for collaborating in a very productive way. Stefano, Ilya, Matteo, Daniele, Marco, Tomaso, thank you for all the time and knowledge we shared together.
I would particularly like to extend warm thanks to the Committee of this thesis, Marco Storace, Mircea Lazar, Francesco Biral and Thorsten Hestermeyer.
Special thanks are for Matteo. Without his guidance and experience, most of this thesis would be never carried out. Matteo, thank you for sharing your methodology and professional abilities. I would also thank you, my friend, for supporting me during my blackout together with Antonio and Laura.
Antonio, your infinite patience and your infinite respect are the abilities that I recognize the most in you. Thank you for all your support, my friend.

Mattia, Giuseppe, Claudio, Laura, Daniele, Carlo, Sara, Irena, thank you for valuable friendship during the years in Trento. I would not spend too many words to explain that I found friends in you. I would only say thank you guys.

Davide, Simone, Daniele, Emiliano, Alessandro, Sara, Giulio, Ilaria, Rudy, Fabio, Alfio, Stefano thank you for the time we spent together during my first studies in Siena. Thank you for introducing me in the real Ph.D. lifestyle.

Finally, I would like to thank my family for the immense support I received in those years (and still now). Antonio and Tina, I am very lucky and I am glad to be your son. This thesis and this Ph.D. is the result of the freedom I had in choosing my way, since of your sacrifices and my effort. I wish to thank Giuliarosa and Walter for their advices and insight. Thank you, with love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

During the last decades Model Predictive Control (MPC) has become the most used technology in highly automated industry [4]. Right now MPC is a standard control approach for both large and medium/small scale processes, even if they are highly complex and multivariable. The word *model* in the acronym MPC stands for *model-based*, while the word *predictive* means that there is a so-called *prediction* of the future behavior of the process when selecting the control action.
MPC exploits the knowledge of a dynamical model that describes the plant behavior in the time domain (usually a state-space model), and is based on an optimization problem that minimizes a cost function, usually leading to a quadratic programming problem. Due to physical reasons, the plant actuators are often constrained to operate in a certain range and this implies the imposition of constraints in the optimization algorithm associated with the MPC control law. It is clear that the constraints on the state, input and output of a dynamical system are embedded in the MPC algorithm and are not imposed *a posteriori*, as occurs in classical control theory (e.g., PID controllers).
One basic block of MPC strategy is the state-space mathematical model. The

interactions between the state variables are described by a set of differential or difference equations, for continuous-time and discrete-time models, respectively. In general, the simpler is the model in terms of number of states, the better it is for MPC design.

There exist two main formulations for MPC, the *implicit* MPC and the *explicit* MPC. If the system is linear with linear constraints (or can be approximated by such a system), one can map the *implicit* MPC formulation to an equivalent *explicit* MPC formulation. By solving an *off-line multi-parametric* linear or quadratic optimization problem (mpLP, mpQP), the *implicit* controller can be mapped to a piecewise affine (PWA) explicit control law, that can be evaluated *on-line* rather easily [1].

The on-line computational complexity of the explicit MPC algorithm can be estimated exactly. The complexity of the off-line multi-parametric optimization problem grows very quickly with respect to the number of constraints in the MPC problem. As a result, for a very large scale problem, the implicit MPC strategy may be the only viable solution.

Technological innovation drives the attention to a class of models that can deal with continuous and discrete components of systems. The dynamical model exploited in the MPC technology can integrate logical rules, switching physical laws and different kind of operating constraints. As a matter of fact, the *hybrid system* class of dynamical models integrating logical behavior has been developed [7].

The term *hybrid system* was introduced to describe the combination of continuous dynamical and discrete event systems. The *hybrid* models can deal with implicit and explicit MPC strategy, as the logical rules for that model can be translated in constraints for the optimization problem, at the expense of its

complexity. The closed-loop stability and optimality analysis are only two of the currently open problems in the MPC research community. Many are the milestone papers, for instance [30], in which the researchers undertake the stability and the optimality issues. Also the MPC robustness problem is addressed in [8], in which the authors survey the possible presence of uncertainty in the dynamical model description together with the stability and performance issues.
The aim of this thesis is to develop low complexity solutions for the control problem based on an MPC strategy, even if the model is hybrid, with particular attention to real-time implementation issues. This involves the stability problem, that must be solved. The stability results will provide the theoretical background for low complexity MPC solutions in the hybrid explicit MPC framework.

1.1 The problem

For time-critical applications the implicit MPC solution is not suitable, hence the explicit solution is more compatible. In the explicit MPC strategy, the feasible set of the multi-parametric optimization problem is bounded and the dynamics are continuous function in that set. The assumption of a bounded set is not restrictive when it is coupled with real life applications issues.
If the system under control is not only described by differential or difference equations, but integrates a switching dynamics logic, then the model can be stated as hybrid model. A particular class of hybrid models is the PWA description, that is suitable for representing a switching system or a region-wise linearized system. An explicit MPC of a PWA system results typically in a very

complex multi-parametric optimization problem. The implementation of the resulting explicit PWA control law, assuming one is able to determine it offline, in the closed-loop can be very hard, due to the large number of regions (i.e. the lookup table is hard to store and to search).

The main problem is that the explicit MPC solution is employed when the sample time must be small, but for large controller (i.e. too many regions) this leads to an inapplicable solution. As a result of the analysis of the state of the art in the hard real time control framework, the target of this thesis is the study of low complexity solution for explicit MPC of PWA systems (e.g., switching MPC), by also addressing the closed-loop stability problem for this class of systems. Moreover, the synthesized low complexity controller can be circuit implementable, thus leading to a complete bundle that fulfills the requirements on sampling time, circuit architecture, controller performances and closed loop-stability.

Right now, the literature results on explicit MPC stability are not directly applicable to low complexity solutions. For instance, if the low complexity controller is the result of a union of several PWA laws (i.e. switching MPC), right now there are few and patchy results giving a stability region in which the controller will be stabilizing for the closed loop, even if the switching MPC can be represented as a PWA control law (i.e. union of PWA). In this case, the *stability analysis* problem is addressed in this thesis.

1.2 Contribution

First of all, in this thesis we introduce a digital architecture implementing the explicit solution of a switched MPC problem. Given a mixed-logic dynamical

system, we derive an explicit controller in the form of a possibly discontinuous piecewise-affine function. This function is then approximated by resorting to piecewise-affine simplicial functions, which can be implemented on a circuit by extending the representation capabilities of a previously proposed architecture to evaluate the control action. The architecture has been implemented on FPGA and validated on a benchmark example related to an air conditioning system.
Secondarily, this thesis proposes a method to analyze uniform asymptotic stability, uniform ultimate boundedness, and input-to-state stability of uncertain piecewise affine systems whose dynamics are only defined in a bounded and possibly non-invariant set $\mathcal{X}$ of states. The approach relies on introducing fake dynamics outside $\mathcal{X}$ and on synthesizing a piecewise affine and possibly discontinuous Lyapunov function via linear programming. The existence of such a function proves stability properties of the original system and allows the determination of a region of attraction contained in $\mathcal{X}$. The procedure is particularly useful in practical applications for analyzing the stability of piecewise affine control systems that are only defined over a bounded subset $\mathcal{X}$ of the state space, and to determine whether for a given set of initial conditions the trajectories of the state vector remain within the domain $\mathcal{X}$.

1.3 Summary of publications

This thesis is based on the following publications.

- S. Trimboli, M. Rubagotti and A. Bemporad, Stability and invariance analysis of uncertain PWA systems based on linear programming, in 50th IEEE CDC-ECC, Orlando, FL, USA, 2011.

- M. Rubagotti, S. Trimboli, D. Bernardini and A. Bemporad, Stability and invariance analysis of approximate explicit MPC based on PWA Lyapunov functions, in 18th IFAC World Congress, Milano, Italy, 2011.

- T. Poggi, S. Trimboli, A. Bemporad and M. Storace, Explicit hybrid model predictive control: discontinuous piecewise-affine approximation and FPGA implementation, in 18th IFAC World Congress, Milano, Italy, 2011.

1.4 Structure of the Thesis

This thesis is organized as follows. After a first introduction of the problem and the contribution, given in Chapter 1, some preliminary concepts are introduced in Section 1.5. In Chapter 2 is given the general MPC framework in which this thesis is contextualized, as well as the state of the art of circuit implementation of PWA controllers and the related stability analysis problem. In Section 3, the architecture for the electronic implementation of discontinuous PWAS functions is introduced. Then the architecture is used to implement explicit switched MPC controllers. The procedure is tested on a hybrid temperature control system. Both the maximum working frequency and the power consumption of the control FPGA implementation are estimated in the range of tens of MHz and tens of mW, respectively. The closed-loop stability analysis problem is addressed in Chapter 4. 4.1 introduces the class of considered uncertain PWA systems. 4.2 is devoted to the one-step reachability analysis of the system, while the main results on the analysis of the extended system are formulated in 4.3. The region of attraction is obtained in Section 4.4, where the analysis of the original system is performed. The input-to-state stability of the system is analyzed in 4.5. 4.6 shows simulation examples. Conclusions are gathered in Chapter 5.

1.5 Basic notations and definitions

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the sets of reals, non-negative reals, integers and non-negative integers, respectively. The floor function $\lfloor \cdot \rfloor$ of $a \in \mathbb{R}$ is defined as the largest $b \in \mathbb{Z}$ such that $a \geq b$. The symbol $\| \cdot \|$ represent any p-norm of a vector, while $\| \cdot \|_\infty$ represents the infinity norm. With respect to the infinity norm, we define the norm ball of radius $\chi > 0$ as $\mathfrak{B}_\chi^\infty \triangleq \{a \in \mathbb{R}^n : \|a\|_\infty \leq \chi\}$. Given a discrete-time signal $v : \mathbb{Z}_+ \to \mathbb{R}^{n_v}$, the sequence of the values of v from the zero instant to the k-th instant is denoted by $\mathbf{v}_{[k]}$. The norm of a sequence is defined as $\|\mathbf{v}_{[k]}\| \triangleq \sup\{\|v(i)\|\}$, for $i = 1, ..., k$. Given a set $\mathcal{A} \subseteq \mathbb{R}^n$, its interior is denoted by $int(\mathcal{A})$, its closure by $\bar{\mathcal{A}}$, and its convex hull by $conv(\mathcal{A})$. If $\mathcal{A}$ is a *polyhedron*, the set of the vertices of $\bar{\mathcal{A}}$ is denoted by *vert*$(\bar{\mathcal{A}})$. A bounded polyhedron is called *polytope*. Given two sets $\mathcal{A} \in \mathbb{R}^n$ and $\mathcal{B} \in \mathbb{R}^n$, the Minkowski sum is $\mathcal{A} \oplus \mathcal{B} \triangleq \{a + b : a \in \mathcal{A}, b \in \mathcal{B}\}$. A function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ is called *$\mathcal{K}$-function* if it is continuous, positive definite, and strictly increasing. A function $\phi : \mathbb{R}_+ \times \mathbb{Z}_+ \to \mathbb{R}_+$ is a *$\mathcal{KL}$-function* if, for each fixed $k \geq 0$, $\phi(\cdot, k)$ is a $\mathcal{K}$ function, for each fixed $c \geq 0$, $\phi(c, \cdot)$ is decreasing, and $\phi(c, k) \to 0$ as $t \to \infty$.

Consider a generic discrete-time nonlinear system

$$x(k+1) = f(x(k), v(k)) \tag{1.1}$$

where $x \in \mathbb{R}^n$ is the state vector, the input $v \in \mathcal{V} \subset \mathbb{R}^{n_v}$ collects the model uncertainties and external disturbances, and $\mathcal{V}$ is a compact set.

Definition 1 (One-step reachable set). *Given a set $\mathcal{X} \subset \mathbb{R}^n$ and system dynamics* (1.1)*, the one step reachable set from $\mathcal{X}$ is*

$$\mathcal{R}(\mathcal{X}) \triangleq \{y \in \mathbb{R}^n : y = f(x, v),\ v \in \mathcal{V},\ x \in \mathcal{X}\}$$

$\square$

Definition 2 (RPI set). *A set $\mathcal{F} \subset \mathbb{R}^n$ is called* robustly positively invariant *(RPI) with respect to dynamics* (1.1) *if, for all $x \in \mathcal{F}$ and all $v \in \mathcal{V}$, $f(x, v) \in \mathcal{F}$.* $\square$

Definition 3 (Uniform asymptotic stability). *Given a set $\mathcal{X} \subseteq \mathbb{R}^n$ with $0 \in \mathcal{X}$, system* (1.1) *is uniformly asymptotically stable in $\mathcal{X}$ ($\mathrm{UAS}(\mathcal{X})$) if there exists a $\mathcal{KL}$-function ϕ such that, for all the initial conditions $x(0) \in \mathcal{X}$ and for all the sequences $\mathbf{v}_{[k]}$ with $v(i) \in \mathcal{V}$, $i = 0, ..., k$, $\|x(k)\| \leq \phi(\|x(0)\|, k)$, for all $k \in \mathbb{Z}_+$.* $\square$

Definition 4 (Uniform ultimate boundedness). *Given a set $\mathcal{X} \subseteq \mathbb{R}^n$ and an RPI set $\mathcal{F} \subseteq \mathcal{X}$, with $0 \in int(\mathcal{F})$, system* (1.1) *is uniformly ultimately bounded from $\mathcal{X}$ to $\mathcal{F}$ ($\mathrm{UUB}(\mathcal{X}, \mathcal{F})$) if, for all $a > 0$ there exist $T(a) > 0$ such that, for every $x(0) \in \mathcal{X}$ with $\|x(0)\| \leq a$, $x(T) \in \mathcal{F}$ for all the sequences $\mathbf{v}_{[k]}$ with $v(i) \in \mathcal{V}$, $i = 0, ..., k$.* $\square$

Definition 5 (Input-to-state stability). *Given a set $\mathcal{X} \subseteq \mathbb{R}^n$ with $0 \in \mathcal{X}$, system* (1.1) *is input-to-state stable in $\mathcal{X}$ ($\mathrm{ISS}(\mathcal{X})$) if there exist a $\mathcal{KL}$-function ϕ and a $\mathcal{K}$-function γ such that, for all the initial conditions $x(0) \in \mathcal{X}$ and for all the sequences $\mathbf{v}_{[k]}$ with $v(i) \in \mathcal{V}$, $i = 0, ..., k$, $\|x(k)\| \leq \phi(\|x(0)\|, k) + \gamma(\|\mathbf{v}_{[k-1]}\|)$, for all $k \in \mathbb{Z}_+$.* $\square$

Chapter 2

State of the Art

2.1 Framework

2.1.1 MPC algorithm

The use of a prediction model, the use of an optimization problem subject to constraints and the *receding horizon* strategy are the three key principles for stating the MPC algorithm. The control algorithm involves the evaluation of an on-line and open-loop optimization subject to state, input and output constraints, in which the prediction model is incorporated. Since only the first input of the optimal sequence is applied, the MPC methodology is also called *receding horizon*.

Referring to Figure 2.1, an instant detailed description of the implicit MPC algorithm is given. At each discrete-time t, the measurements y_t acquired from the sensors and the dynamical model are used in order to predict the plant behavior. The state prediction is computed among a N-step time window (N is usually called *prediction horizon*). The optimal input sequence u^*_{t+k} is computed by minimizing the related cost function. According to the receding horizon control paradigm, only the first element of the optimal sequence u^* is applied as the

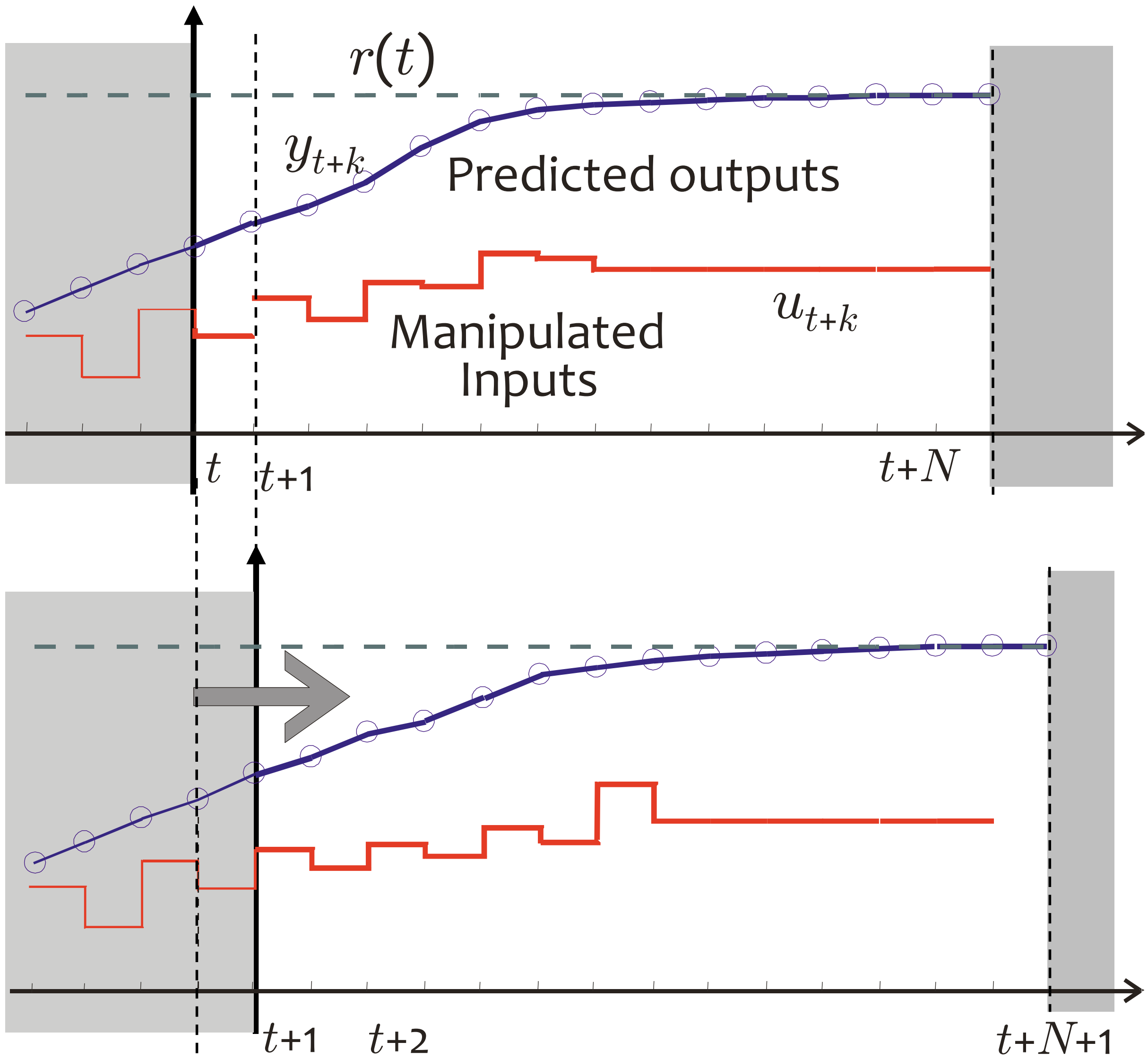

Figure 2.1: Implicit MPC algorithm

input u_{t+1} of the real plant. At the next time step $t+1$, a new optimization problem is solved by exploiting the new measurements, and only the first element of the resulting optimal control sequence is applied to the plant. As a result of the receding horizon strategy, it follows that the control action is applied in a closed-loop fashion.
In general the dynamical model can be nonlinear, the cost function can be a 2-norm, 1-norm or ∞-norm and the constraints can be time varying. A nonlinear dynamical system implies a more complicated optimization algorithm. For a 1-norm or a ∞-norm cost function is sufficient a linear programming algorithm, while for a 2-norm cost function is required a quadratic programming algorithm. Moreover, time varying constraints imply a high computational burden. In the following, a general formulation of MPC problem is stated.

Definition 6 (Problem). *Let $N \geq 1$ be given, let $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^m$ be sets which represent the state and the input constraints, respectively, and contain the origin in their interior. The prediction model is $x_{k+1} = g(x_k, u_k)$, $k \geq 0$, with $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ a nonlinear, possibly discontinuous function with $g(0,0) = 0$. Let $F : \mathbb{R}^n \to \mathbb{R}_+$ with $F(0) = 0$ and $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ with $L(0,0) = 0$ be known mappings. For each discrete time instant $k \geq 0$ let x_k the measured state, let $x_{0|k} \triangleq x_k$ and minimize the cost function*

$$J(x_k, \mathbf{u}_k) \triangleq F(x_{N|k}) + \sum_{i=0}^{N-1} L(x_{i|k}, u_{i|k}) \tag{2.1}$$

over all input sequences $\mathbf{u}_k \triangleq (u_{0|k}, \dots, u_{N-1|k})$ subject to the constraints

$$x_{i+1|k} \triangleq g(x_{i+1|k}, u_{i|k}) \quad , \quad i = 0, \dots, N-1, \tag{2.2}$$

$$x_{i+1|k} \in \mathbb{X} \quad , \quad \forall i = 1, \dots, N, \tag{2.3}$$

$$u_{i|k} \in \mathbb{U} \quad , \quad \forall i = 1, \dots, N-1 \tag{2.4}$$

In Problem 6, the function F denotes the terminal cost, while L denotes the stage cost and N is the prediction horizon. The term $x_{i|k}$ denotes the predicted state at future instant i, while k is the actual instant. The state preview $x_{i|k}$ is obtained by applying the input sequence $\{u_{i|k}\}_{i=0,\dots,N-1}$ to the dynamical model, with the measured state $x_k = x_{0|k}$ as initial condition. The optimization variable in the minimization of the cost function $J(x_k, \mathbf{u}_k)$ is the input sequence $\{u_{i|k}\}_{i=0,\dots,N-1}$. Suppose that the problem of minimizing (6), subject to (2.2), (2.3), (2.4), is feasible and let $\{u^*_{i|k}\}_{i=0,\dots,N-1}$ denote an optimal solution. According to the receding horizon strategy, the MPC control action $u^{MPC}(x_k)$ is obtained as the first element of the optimizer $u^* = \{u^*_{i|k}\}_{i=0,\dots,N-1}$

$$u^{MPC}(x_k) \triangleq u^*_{0|k}, k \geq 0 \tag{2.5}$$

2.1.2 Hybrid models

A hybrid system is a system whose behavior is characterized by several modes of operation. For each mode there is a differential or difference equation set describing the continuous time or discrete time dynamics. The switch between the models occurs when a particular event happens. These events can be caused by variables crossing specific thresholds (state event), by the elapsing of certain time periods (time events), or by external inputs (input events) and, for example, they can be modeled as a *finite state machine*. Due to switch acting in the hybrid system, the dynamics can be discontinuous. Also, the stability analysis is much harder than in the case of a single operating mode system.
The hybrid model class is large, but this research focus on a subset of this class, for which there is a trade-off between the modeling power and the complexity

of analysis. The trade-off results in a simpler model representation, but sufficiently structured in order to represent a industrially relevant process. Without loss of generality, in this paper the author considers the *piecewise affine* (PWA) functions as the class of hybrid models, defined as follows

Definition 7 (PWA Function)**.** *A function $z(x) : X \to \mathbb{R}^s$, where $X \subseteq \mathbb{R}^n$ is a polyhedral set, is PWA if it is possible to partition X into convex polyhedral regions, CR_i, and $z(x) = H^i x + k^i$, $\forall x \in CR_i$.*

In particular, a PWA model is described by the following definition

Definition 8 (PWA Model)**.** *A PWA dynamical model is the set of discrete time linear systems*

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \\ i &: H_i x(k) + L_i u(k) \leq K_i, \; i = 1, \ldots, s \end{aligned} \tag{2.6}$$

where $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times m}$, $C_i \in \mathbb{R}^{p\times n}$, $D_i \in \mathbb{R}^{p\times m}$, $f_i \in \mathbb{R}^n$, $g_i \in \mathbb{R}^p$ are the state space affine models, i is the active mode at time k and H_i, L_i, K_i are matrices of appropriate dimension.

In the Definition 8, for each i, the polytopic set depends on the state $x(k)$ and the input $u(k)$ of the system. All the sets are polytopic, hence bounded by definition.

2.1.3 Explicit MPC

The implicit MPC strategy described in Section 2.1.1 requires to solve an online optimization problem. The corresponding code implementation of the algorithm is *time unpredictable*, hence it is not suitable for hard real-time implementation, as it does not provide guarantees on the execution time. There are a lot of processes for which implicit MPC is suitable, for instance a low frequency chemical plant, for which the execution time is not really important. A hard real-time control loop must be time predictable, certifiable (i.e. provides guarantee) and relatively fast. However, the effort of the researchers in the field of the optimization algorithms has produced solutions to speed-up the optimization algorithm, for example the very fast active set strategies [21]. One of the most important result in the hard real-time control is addressed in the explicit solution of the implicit MPC strategy [5].

The idea behind the explicit MPC is to solve off-line several implicit MPC instances for all x_k within a given set $X = \{x \in \mathbb{R}^n : Hx \leq K\} \subset \mathbb{R}^n$, and to make the dependence of the input on the state explicit. The set X is assumed to be polytopic (i.e. bounded set and described by linear inequalities). As a result, by solving a multi-parametric program, the equivalent explicit solution of the implicit ones is a piecewise affine function of the state

$$\begin{aligned} u(k) &= F^i x(k) + G^i \\ if \;\; H^i x(k) &\leq K^i \end{aligned} \tag{2.7}$$

where i indexes the i-th region $\mathcal{R}$ in the explicit linear MPC.

Also, for a hybrid model, under the assumption of bounded state and linear cost function, the explicit MPC solution is a PWA function of the state.

It is clear that formulation (2.6) is more suitable for real-time controllers, rather than the solution of implicit MPC. In order to evaluate the optimal input at time step k, one has to evaluate a finite number of sets of inequalities and has to compute one matrix product and one sum. The actual microprocessor technology is sufficiently mature to achieve that specifications. As the resulting few lines of code can run in high speed, the explicit MPC is suitable for time-critical applications.

2.1.4 Switched MPC

The architecture described in the previous sections can be used to implement an explicit SwMPC controller in approximate form. In this section we summarize the main elements of the SwMPC control strategy. A MLD system subject to constraints can be controlled through an implicit HMPC strategy. The explicit HMPC strategy can be applied to a MLD model, after recasting it to an equivalent PWA form. A suitable strategy to control (2.6) in state feedback, subject to state and input constraints, is the explicit HMPC [2]. This approach requires enumerating all the feasible switch sequences between the dynamics i and solving a multi-parametric quadratic problem for each sequence. Storing all the control gains leads to a large use of memory blocks in the FPGA implementation with respect to a simpler controller such as the SwMPC. Considering only the sequences for which the region i is the same during the prediction steps, since the constraints that define the PWA regions are ignored after the first prediction step, the number of multi-parametric quadratic problems to be solved is equal to the number of PWA regions, leading to a suboptimal solution to the control problem. A formal definition of the SwMPC will be given in Sec. 3.3.

2.2 Controller circuit implementation

The real-time implementation of linear model predictive control strategies in embedded architectures was deeply analyzed in [13,29], and an automated code generation strategy was developed in [36]. Starting from a model, through the definition of a model-based control strategy suitable for the implementation in an embedded architecture, the problem of implementing the control strategy was also investigated for parallel architectures in [28]. Often a system that integrates continuous dynamics and logical structures can be described as a mixed-logic dynamical system (MLD). A suitable strategy to control a MLD system subject to constraints is hybrid model predictive control (HMPC). In order to obtain a HMPC, one has to solve on-line a mixed-integer quadratic or a mixed-integer linear programming problem. For a high-dimensional model, solving this kind of problems may be computationally too expensive for fast real-time application [14].

Explicit reformulations of HMPC can be carried out by solving off-line a sequence of multi-parametric quadratic or linear problems [14]. The resulting solution is a possibly discontinuous piecewise affine (PWA) function of the state. In other words, the control modes are linear affine over polytopes partitioning the state domain, thus making this approach more suitable for the embedded control implementations. Storing the gains of the explicit HMPC requires larger and larger memory blocks in the electronic implementation as the number of partitions grows. Moreover, in order to evaluate the control action, the pre-computed gains should be selected, according to the state value, from a look-up table associated to the explicit controller. As a result, since the gains selection from the look-up table can be made by a binary-tree search [41], or by other more sophisticated algorithms, determining the correct mode can be a hard problem if the number of regions is too large.

A suitable strategy, alternative to HMPC, to reduce drastically the number of partitions is the switched MPC (SwMPC) approach, successfully applied for instance in [18], where a PWA system is controlled by a set of linear MPC controllers, each one defined over a different polytope of the domain. In explicit form, SwMPC is basically a set of patched PWA controllers. For each i-th *region* $\mathcal{R}_i$ of the domain, a linear MPC problem is solved, whose solution is a continuous PWA function defined over a polytopic partition of the region. Note that a MLD model can be converted in an equivalent PWA formulation [2].

The resulting PWA control function may be discontinuous only at the boundaries of the regions. The overall number of polytopes obtained with the explicit SwMPC approach is typically much lower than the one obtained with the explicit HMPC, especially when the number of optimization variables grows. However, the SwMPC complexity reduction with respect to HMPC is not costless, since optimal switching sequences are restricted to constant mode sequences, possibly breaking *a-priori* stability properties. However, *a-posteriori* stability analysis of the SwMPC can be performed exploiting the results in [20] and in [38].

2.3 Stability analysis

In the last decade the interest in studying the dynamical properties of piecewise affine (PWA) systems has increased considerably, due to their powerful modeling capabilities. Discrete-time PWA models are a special class of hybrid systems that can represent combinations of finite automata and linear dynamics, are a good approximation of nonlinear systems [39], and are equivalent to hybrid systems in mixed logical dynamical form [2, 7].

Analyzing the stability of PWA systems is fundamental to describe the properties of an autonomous hybrid system, or to check a-posteriori the stability of a given closed-loop system [6, 17]. In particular, stability analysis becomes fundamental when a PWA control law is synthesized without a-priori guarantees of closed-loop stability, for example when explicit model predictive control (MPC) laws [9] are approximated in order to reduce their complexity [1].

The most widely used methods for stability analysis of discrete-time PWA systems are based on piecewise quadratic (PWQ) Lyapunov functions [20]. Such methods rely on the solution of a semi-definite program to get a stability certificate. As highlighted in [22], the search for a PWQ Lyapunov function can be overly conservative, even with the use of the so-called S-procedure (see e.g. [15]). A valid alternative are PWA Lyapunov functions, that are computed by solving a linear program (LP) [11]. Other types of Lyapunov functions can be used for the same purpose, such as piecewise polynomial Lyapunov functions [34]. For an overview of such methods, the interested reader is referred to [11].

Most of the existing literature on stability analysis of PWA systems assumes that the set $\mathcal{X}$ of states in which the PWA dynamics are defined is invariant, as the notion of stability has no practical relevance if the state trajectory exits the domain of definition of the dynamics [11]. However, often the PWA system to be analyzed is defined in a set $\mathcal{X}$ that may not be invariant. A possible approach is to perform a reachability analysis to find the maximum positively invariant set to establish, using a recursive procedure, an invariant subset of the given set $\mathcal{X}$ (see [35], [12, Chap. 4-5] and the references therein). Unfortunately this procedure often leads to very involved solutions, due to the exponential complexity of reachability analysis of PWA systems, and in many cases searching the maximum invariant set is an undecidable problem.

Most of the existing literature on stability analysis of PWA systems also deals with *nominal* stability analysis, in spite of the practical relevance in applications of certifying stability properties *in the presence of disturbances*. The problem consists of determining if the state will converge to the origin (or to a terminal set) despite parametric uncertainties and/or external disturbances affecting the process. The reason why this problem was almost ignored in the literature is mostly due to the complexity of uncertain switching systems. Notably, interesting results were introduced for the case of additive disturbances in [24], where an optimal control strategy is synthesized to steer the state in finite time to a terminal set, and in [35], where the authors determine how to drive the state into the maximal robust invariant set in minimum time, using set-theoretic techniques. Some classical results appeared for parametric uncertainties in case of linear parameter varying systems [12, Chap. 7]. Results for linear switched systems can be found in [45], where quadratic stabilizability is analyzed in case of two discrete states and polytopic parametric uncertainties, and in [27], where the synthesis of switching control laws is tackled, assuring that the state is ultimately bounded within a given set, in case of both parametric uncertainties and external disturbances. To the authors' knowledge, general results on the analysis of uncertain PWA systems are not available in the literature.

Chapter 3

Explicit HMPC approximation and FPGA implementation

In this thesis we extend to discontinuous PWA functions the results of [32, 40], related to the circuit implementation of continuous PWA functions. Recalling Sec. 2.2, where the state of the art regarding the implementation of possibly discontinuous PWA functions in approximate form on fast digital circuits, we restrict our attention to PWA control functions for which each mode is defined over a hyper-rectangular region. This limits the approach to hybrid dynamical systems where threshold conditions only depend on single components of the state vector.

In order to circuit implement the SwMPC solution in an approximate but fast way, we resort to a modified version of the method proposed in [10]. Accordingly, each explicit solution (valid over the i-th hyper-rectangular region) is first approximated by using a PWA continuous function, defined over a regular simplicial partition of the i-th region (called *PWAS* function). Then, the obtained approximations can be merged into one PWAS discontinuous function, which can be directly mapped on programmable hardware such as a field programmable gate array (FPGA).

The architectures able to implement PWAS functions proposed so far in [19, 37,40] perform a linear interpolation of the values of the function at the vertices of the simplex the input belongs to. The main limit of such an approach is that the implementable functions are continuous. If functions with discontinuities that are not perpendicular to an axis were to be implemented, more complex and power-hungry architectures would be necessary [23, 31].

3.1 Circuit implementation of continuous PWAS functions

In this section, we briefly summarize the mathematical theory the proposed architecture is based on and we introduce some basic definitions. We deal with a continuous PWAS function $f_{PWAS} : S \to \mathbb{R}$, defined over a properly scaled n-dimensional compact domain $S = \{z \in \mathbb{R}^n : 0 \leq z_h \leq m_h,\ h = 1, \ldots, n,\ m_h \in \mathbb{N}\}$. Function f_{PWAS} can be easily implemented by introducing a regular partition of the domain S [32]: each dimensional component z_h of the domain S is divided into m_h subintervals of unitary length. As a consequence, the domain S is partitioned into $\prod_{h=1}^{n} m_h$ hyper-squares and contains $N = \prod_{h=1}^{n}(m_h + 1)$ vertices v_k collected in a set V. Each hyper-square can be further partitioned (*simplicial partition*) into $n!$ non-overlapping regular simplices. The coordinates of the corner of the hyper-square closest to the origin that contains a given input z can be found by extracting the integer part of z. The exact position of z within the related simplex is coded by the decimal part of z (denoted as δz) [32]. The PWAS function f_{PWAS} is linear over each simplex of the partitioned domain S and can be expressed as a linear combination of N α-basis functions

$$f_{PWAS}(z) = \sum_{k=0}^{N-1} c_k \alpha_k(z). \tag{3.1}$$

Once the scaled simplicial domain is defined, the basis functions (belonging to the α-basis) are directly defined as well. The k-th α-function is PWAS, holds the value 1 at the vertex corresponding to v_k and the value 0 at all the other vertices.

The shape of a given PWAS function f_{PWAS} is coded by the N coefficients c_k in Eq. (3.1), which are the values of f_{PWAS} at the vertices v_k of its simplicial partitions. Henceforth, we assume that the coefficients are already determined by a function approximation procedure (see, e.g. [10]).

The coefficients c_k ($k = 1, \dots, N$) are stored by assigning a proper memory address to each vertex v_k of the simplicial partition. Define $\beta_p : \mathbb{N}^n \to \mathbb{N}_b$ as the binarizing operator that, given a column vector of n integer values and a precision p, returns a np-long string of bits, concatenating the binary values of the elements of the vector. For instance, if $v_k = [2,\, 0,\, 5]^T$ and $p = 3$, then $\beta_p(v_k) = 010\,000\,101$. Then, $\beta_p(v_k)$ is an unambiguous address for the vertex v_k. The value of $f_{PWAS}(z)$ can be calculated as a linear interpolation of the f_{PWAS} values at the vertices of the simplex containing z, i.e., as a linear interpolation of a subset of $n+1$ coefficients c_k:

$$f_{PWAS}(z) = \sum_{j=0}^{n} \mu_j c_{\Omega_j} \tag{3.2}$$

where the μ_j's are the weights that give z as a convex combination of the vertices of the simplex that contains it (i.e., $z = \sum_{j=0}^{n} \mu_j v_{\Omega_j}$, with $\sum_{j=0}^{n} \mu_j = 1$) and Ω_j is a function that maps the index j of the weight μ_j into the corresponding index k of one of the vertices surrounding z [32]. Ω_j, as well as the interpolation weights μ_j, depends on z. This dependence is omitted here for ease of notation.

As a consequence, the circuit realization of a PWAS function proposed in [40] requires three functional elements:

1. a memory where the N c_k coefficients are stored;
2. a block that finds, for any given input z, the indices Ω_j and the coefficients μ_j;
3. a block performing the weighted sum (3.2).

Since the $\{c_k\}$'s are stored in a memory, Ω_j corresponds uniquely to the address Ω_j^b of the j-th coefficient in Eq. (3.2), through the binarizing operator β_p

$$\Omega_j^b = \beta_p(\lfloor z \rfloor + a_j), \qquad j = 0, \ldots, n \tag{3.3}$$

where a_j's are vectors whose components are calculated from the decimal parts of the input z [32].

3.2 Generalization to a class of discontinuous functions

The algorithm presented in Sec. 3.1 can be generalized to include a particular class of discontinuous functions, namely the functions composed of continuous PWAS functions separated by discontinuities that lie perpendicular to a coordinate axis. In this case, we can define an index labeling the subregion a continuous PWAS function is defined over and use this index to solve the point location problem, i.e. to address correctly the memory containing the coefficients.

Let us suppose that there are D_h discontinuities orthogonal to each axis z_h, with $h = 1, \ldots, n$. The discontinuities are hyperplanes (straight lines for $n = 2$)

in the form $z_h = d_{h,t}$ ($h = 1, \ldots, n$; $t = 1, \ldots, D_h$; $d_{h,t}$ constant) that further partition the domain S into $P = \prod_{h=1}^{n} (D_h + 1)$ hyper-rectangular regions $\mathcal{R}_i$ (discontinuity partition, $i = 1, \ldots, P$). Figure 3.1 shows an example of a two-dimensional domain of a discontinuous function with $m_1 = 4$, $m_2 = 3$, $D_1 = 2$ and $D_2 = 1$. Both the regular simplicial partition and the six regions $\mathcal{R}_i$ are highlighted.

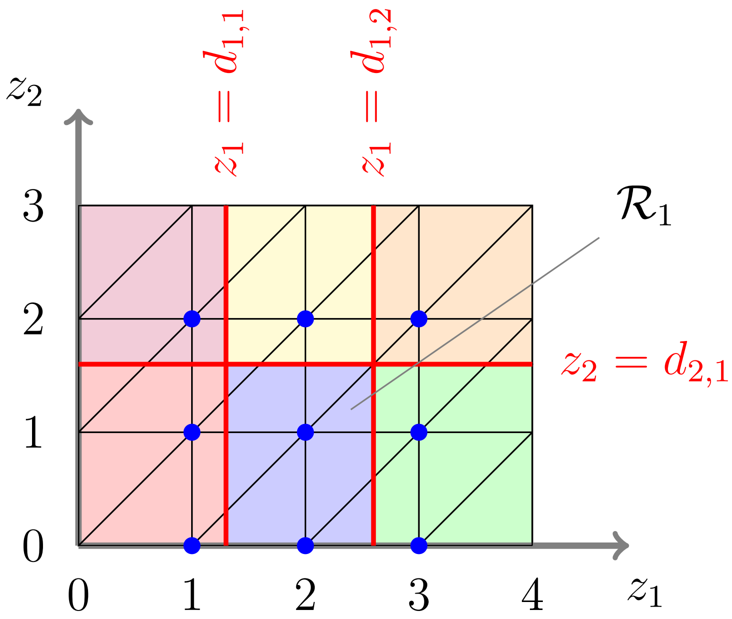

Figure 3.1: Two-dimensional domain with discontinuities.

The discontinuous function f_{PWAS} can be defined as follows:

$$f_{PWAS}(z) = f_{PWAS_i}(z) = \sum_{k=0}^{N-1} c_k^i \alpha_k(z), \quad \forall z \in \mathcal{R}_i \tag{3.4}$$

where f_{PWAS_i} are continuous functions that can be implemented using the technique proposed in Sec. 3.1. They are defined over the whole domain S, since the set of α-functions is unique for both continuous and discontinuous PWAS functions (see Eqs. (3.1) and (3.4)). The shape of a particular function f_{PWASi} is coded by the coefficients related to the vertices that lie inside $\mathcal{R}_i$ and immediately outside of the boundary of $\mathcal{R}_i$. Thus, most of the coefficients c_k^i related to vertices that fall outside $\mathcal{R}_i$ can be discarded. Indeed, we need to consider only the set V_i of the vertices that lie inside the smallest hyper-rectangle containing $\mathcal{R}_i$ defined over the vertices of the simplicial partition. For instance, in Fig. 3.1 the vertices V_1 that define the shape of f_{PWAS1} over $\mathcal{R}_1$ are marked by blue dots. They are all contained inside the rectangle $[1,\ 3] \times [0,\ 2]$. [1] Then, f_{PWASi} is completely characterized by the coefficients corresponding to V_i:

$$f_{PWASi}(z) = \sum_{k \in \mathcal{K}_i} c_k^i \alpha_k(z), \qquad z \in \mathcal{R}_i,$$

where $\mathcal{K}_i = \{k : v_k \in V_i\}$.

An example of discontinuous PWAS function is shown in Fig. 3.2. In this case, f_{PWAS} is defined over a one-dimensional domain $S = [1, 5]$, with one discontinuity ($z = d_{1,1} = 2.7$) and

$$f_{PWAS}(z) = \begin{cases} f_{PWAS0}(z), & z \in \mathcal{R}_0 = [1, 2.7) \\ f_{PWAS1}(z), & z \in \mathcal{R}_1 = [2.7, 5] \end{cases}$$

To calculate f_{PWAS0}, it is necessary to know the value of its coefficients at the vertices v_0, v_1, v_2 (all $\in \mathcal{R}_0$) and $v_3 (\notin \mathcal{R}_0)$, then $V_0 = \{v_0, v_1, v_2, v_3\}$ and $\mathcal{K}_0 = \{0, 1, 2, 3\}$. On the other hand, the set of vertices needed to evaluate f_{PWAS1} is v_3, v_4, v_5 (all $\in \mathcal{R}_1$) and $v_2 (\notin \mathcal{R}_1)$, then $V_1 = \{v_2, v_3, v_4, v_5\}$ and

[1] A formal definition of V_i is

$$V_h = \left\{ v_k \in V : v_i \in \operatorname*{argmin}_{\mathcal{R}(v_k) \supseteq \mathcal{R}_i} \{|\mathcal{R}(v_k)|\} \right\}$$

where $\mathcal{R}(v_k)$ is a hyper-rectangle and $|\mathcal{R}(v_k)|$ is its volume.

$\mathcal{K}_1 = \{2, 3, 4, 5\}$. We notice that the correspondence between vertices of the simplicial partition and coefficients is no longer one-to-one, as both c_2^0 and c_2^1 are related to v_2 and both c_3^0 and c_3^1 are related to v_3.

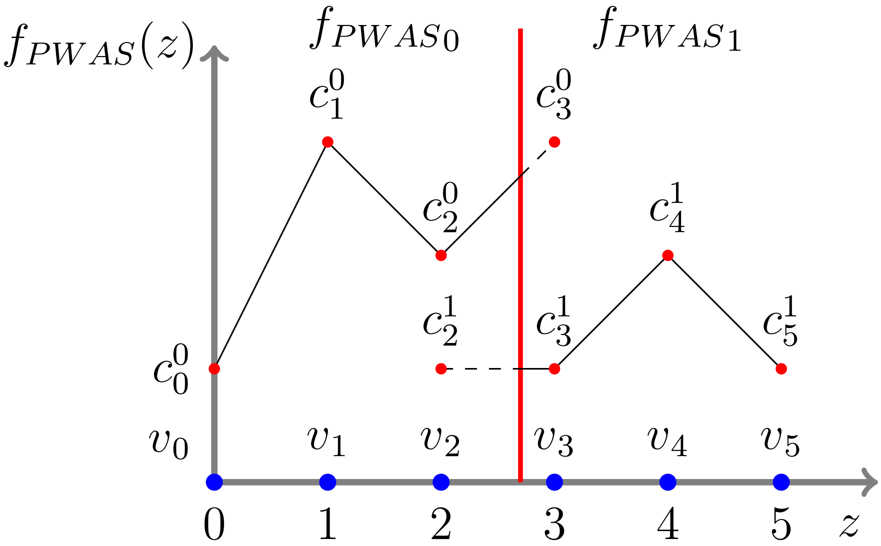

Figure 3.2: One-dimensional discontinuous PWAS function

Since some coefficients c_k^i of each function f_{PWASi} are in a relation many-to-one with the vertices of the simplicial partition and since they are stored in the same memory, we need to refine the way they are addressed. For any given $z = [z_1, z_2, \ldots, z_n]^T \in \mathcal{R}_i$, we can define

$$r(z) = \begin{bmatrix} \sum_{t=1}^{D_1} u(z_1 - d_{1,t}) \\ \sum_{t=1}^{D_2} u(z_2 - d_{2,t}) \\ \vdots \\ \sum_{t=n}^{D_1} u(z_n - d_{n,t}) \end{bmatrix} \tag{3.5}$$

where $u(\cdot)$ denotes the unitary step function. Then each rectangle can be uniquely identified by the binary string (with $n \times (p-1)$ bits) $\beta_{p-1}(r(z))$, $z \in \mathcal{R}_i$. We choose p as the lowest integer such that $D_h \leq 2^{p-1} - 1$ ($h = 1, \ldots, n$).

Given a point z, we need to find the rectangle $\mathcal{R}_i$ such that $z \in \mathcal{R}_i$ and the related set of coefficients c_k^i. Thus, the index map Ω_j is redefined so that it corresponds uniquely (for any z) to the memory address

$$\Omega_j^b = \beta_{p+1}(2r(z) + \lfloor z \rfloor + a_j), \qquad j = 0, \ldots, n \tag{3.6}$$

Finally, a discontinuous PWAS function can be evaluated using the method provided in Sec. 3.1 by substituting Eq. (3.3) with Eq. (3.6). The binary vector $\beta_{p-1}(r(z))$ can be easily obtained by using comparators to process the input z and find the region it belongs to.

To evaluate each function f_{PWASi} it is possible to use the architecture A proposed in [40], that provides a correct output every $p + q + n + 5$ clock cycles, where p and q are the number of bits used to code the integer part and the decimal part, respectively, and n is the input dimension. As stated before, the coefficients c_k^i defining the shape of f_{PWAS} are stored in a memory and they can be addressed by calculating the strings Ω_j^b. Then, we need to modify the way the address is calculated in [40], according to Eq. (3.6). Equation (3.5) is evaluated asynchronously with respect to the system clock through comparators and 1-bit adders when the input vector z is fed into the circuit.

The number of elementary devices (comparators, adders, multipliers, etc.) required to evaluate a discontinuous PWAS function is reported in Tab. 3.1. The items of the part added to evaluate discontinuous functions are kept separated and described in italic text.

3.2.1 An example

Given the three-dimensional domain $S = [0, 15]^3$, we want to calculate the value of a certain function f_{PWAS} at the point $z = [2.6,\ 4.3,\ 1.4]^T$. f_{PWAS} has

Item	Bits	# Devices
Comparator	q	n
Multiplexer	n	n
ROM	$2^{np} \times b$	1
Adder/Subtractor	$n+1$	n
Adder/Subtractor	q	n
Multiplier	$b \times q$	1
Comparator	$p+q$	$\sum_{i=1}^{n} D_i$
Adder	1	n
Adder	$p+1$	n
Shift Register	p	1

Table 3.1: Number of elementary devices to evaluate a discontinuous PWAS

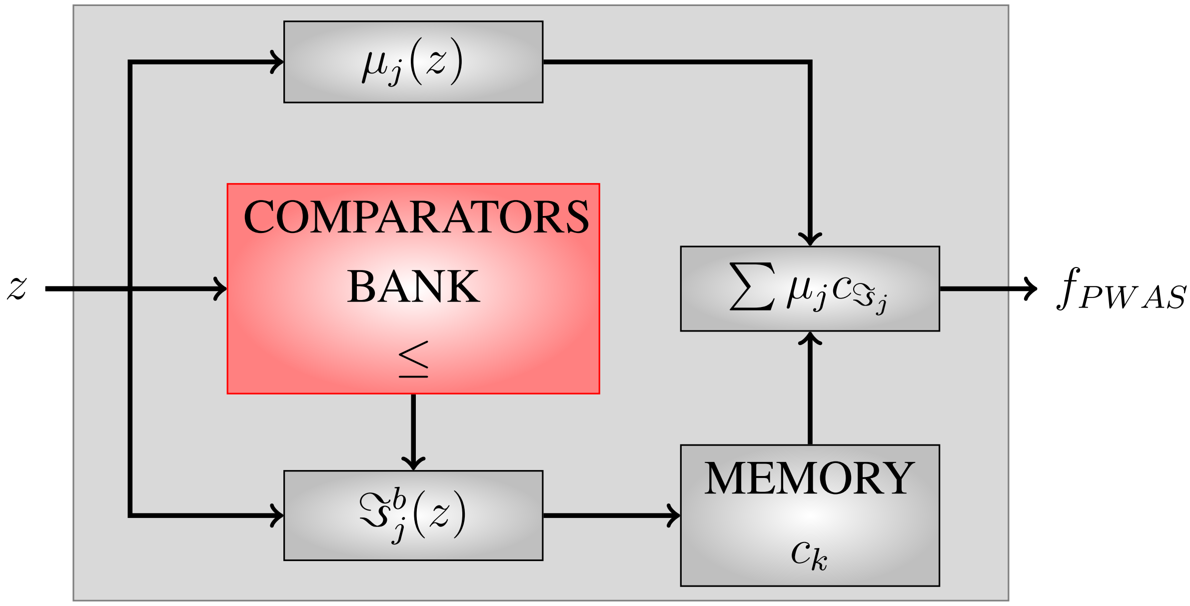

Figure 3.3: Architecture to implement discontinuous functions

discontinuities in $d_{1,1} = 2.1$, $d_{1,2} = 8.5$, $d_{2,1} = 0.9$ and $d_{3,1} = 9.7$. We need to retrieve from the memory $n + 1 = 4$ coefficients $c^h_{\Omega_j}$ by evaluating Eq. (3.6).

The integer part of z is easily obtained $\lfloor z \rfloor = [2,\ 4,\ 1]^T$, while the vectors

a_j depend on the decimal part of z and assume the values (the reader is referred to [32] for details):

$$a_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad a_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

By applying Eq. (3.5) we obtain

$$r(z) = \begin{bmatrix} u(z_1 - d_{1,1}) + u(z_1 - d_{1,2}) \\ u(z_2 - d_{2,1}) \\ u(z_3 - d_{3,1}) \end{bmatrix} = \begin{bmatrix} 1+0 \\ 1 \\ 0 \end{bmatrix}$$

Fixing $p = 4$, the first address is given by

$$\begin{aligned} \Omega_0^b &= \beta_{p+1}(2r(z) + \lfloor z \rfloor + a_0) \\ &= \beta_5([2,\ 2,\ 0]^T + [2,\ 4,\ 1]^T + [0,\ 0,\ 0]^T) \\ &= \beta_5([4,\ 6,\ 1]^T) \\ &= [00100\ 00110\ 00001] \end{aligned}$$

and the others take the values

$$\begin{aligned} \Omega_1^b &= [00101\ 00110\ 00001], \\ \Omega_2^b &= [00101\ 00110\ 00010], \\ \Omega_3^b &= [00101\ 00111\ 00010]. \end{aligned}$$

3.3 Switched MPC

The architecture described in the previous sections can be used to implement an explicit SwMPC controller in approximate form. In this section we summarize the main elements of the SwMPC control strategy. As explained in Sec. 2.2, a

MLD system subject to constraints can be controlled through an implicit HMPC strategy. The explicit HMPC strategy can be applied to a MLD model, after recasting it to an equivalent PWA form. A time-invariant PWA discrete-time model is defined as follows

$$x(k+1) = A_i x(k) + B_i u(k) + f_i \tag{3.7}$$

$$i \; : \; H_i x(k) \leq K_i \; , \; i \in \mathcal{I} \tag{3.8}$$

where $x \in \mathbb{R}^{n\times 1}$, $u \in \mathbb{R}^{m\times 1}$, $A_i \in \mathbb{R}^{n\times n},\ B_i \in \mathbb{R}^{n\times m},\ f_i \in \mathbb{R}^{n\times 1}$ characterizes the i-th mode, $H_i,\ K_i$ are matrices of suitable dimensions defining the i-th region $\mathcal{R}_i$, $\mathcal{I} = \{1,\ldots,P\}$ and P is the number of regions. A suitable strategy to control (3.7), (3.8) in state feedback, subject to state and input constraints, is the explicit HMPC [2]. This approach requires enumerating all the feasible switch sequences between the dynamics i and solving a multi-parametric quadratic problem for each sequence. Storing all the control gains leads to a large use of memory blocks in the FPGA implementation with respect to a simpler controller such as the SwMPC. Considering only the sequences for which the region i is the same during the prediction steps, since the constraints that define the PWA regions are ignored after the first prediction step, the number of multi-parametric quadratic problems to be solved is equal to the number of PWA regions, leading to a suboptimal solution to the control problem.

In order to formulate (3.7), (3.8) as standard linear system, fixing the mode i, we merge the affine term f_i in the input matrix B_i. The resulting set of linear systems is a suitable formulation for a set of linear MPCs. Let $v(k)$ be a measured input disturbance such that $v(k) = 1, \forall k \geq 0$, then (3.7) can be rewritten as follows

$$x(k+1) = A_i x(k) + B_i u(k) + f_i v(k) \tag{3.9}$$

or, in a more compact way,

$$x(k+1) = A_i x(k) + \bar{B}_i \bar{u}(k) \tag{3.10}$$

where $\bar{B}_i = \left[\begin{smallmatrix} B_i & f_i \end{smallmatrix}\right]$ and $\bar{u} = \left[\begin{smallmatrix} u'(k) & v(k) \end{smallmatrix}\right]'$. Exploiting model (3.10), (3.8) we define a set of linear MPCs based on the following quadratic problem:

$$\begin{aligned} \min_{U=[u_0,\ \dots,\ u_M]} J(x,U) = \sum_{k=0}^{M-1} & x(k)' Q x(k) + \\ & + u'(k) R u(k) + \rho \epsilon^2 \\ \text{s.t.} \quad x_{min} - \epsilon \leq & x(k) \leq x_{max} + \epsilon, \\ u_{min} \leq & u(k) \leq u_{max}, \\ x(k+1) = & A_i x(k) + \bar{B}_i \bar{u}(k) \end{aligned} \tag{3.11}$$

where M is the prediction horizon; the quantities x_{min}, x_{max}, u_{min}, u_{max} are state and input bounds, respectively; ϵ is a slack variable, weighted by ρ; R, Q are weight matrices of suitable dimensions; A_i, B_i are the i-th model matrices. At time k, only the first component u_0 of the optimal sequence is applied, in a receding horizon fashion. A SwMPC is a set of linear MPCs based on (3.11) each one defined over its corresponding region $\mathcal{X}_i$. For each control step, one has to evaluate the active mode i and compute the i-th control action.
Problem (3.11) is stated as a regulation of the states to the origin. A reference tracking problem can be recast as partial state regulation problem by extending the state vector and exploiting the same formulation, as follows. Let $y(k) = Cx(k)$ be the output of model (3.10), (3.8), where $C \in \mathbb{R}^{o \times n}$ is the output matrix, then consider the extended state vector $x_e = \left[\begin{smallmatrix} x' & r_y' \end{smallmatrix}\right]$, where $r_y \in \mathbb{R}$. The reference tracking formulation is obtained by substituting in (3.11) $Q = \left[\begin{smallmatrix} C & -\mathbf{I} \end{smallmatrix}\right]' Q_y \left[\begin{smallmatrix} C & -\mathbf{I} \end{smallmatrix}\right]$, where Q_y is a weight matrix of suitable dimension and $\mathbf{I}$ is the

identity matrix of order o. This leads to a reference tracking problem with cost function $J(x,U) = \sum_{k=0}^{M-1}(Cx(k) - r_y)'Q_y(Cx(k) - r_y) + u'(k)Ru(k) + \rho\epsilon^2$.

Exploiting the results in [9], each linear MPC of the SwMPC formulation could be explicitly solved through a multi-parametric quadratic problem, leading to a set of linear explicit MPCs. Moreover, in each region the explicit controller is a continuous PWA function of the state. The overall explicit SwMPC controller is defined as follows.

$$u(k) = F_j^i x(k) + G_j^i \tag{3.12}$$

$$\text{if } H_j^i x(k) \leq K_j^i \tag{3.13}$$

where j indexes the polytopes of the i-th region $\mathcal{R}_i$ in the explicit linear MPC. In the framework described in the previous sections, these polytopes reduce to identical simplexes and the regions are hyper-rectangles.

In the next section, a benchmark for the SwMPC implemented with PWAS in a FPGA reveals the capabilities of the proposed approach, suggesting that the SwMPC performances can get very close to the HMPC ones, at least for functions belonging to the class described in Sec. 3.2.

3.4 Experimental validation

In order to test the circuit implementation on FPGA, we propose a revised case study of the hybrid temperature control problem described in the Hybrid Toolbox [3]. The model is a MLD description of an air conditioning system and in closed loop with a HMPC. A discontinuous PWA control of the MLD system is found by using the SwMPC approach described in Sec. 3.3 using the same HMPC tuning parameters for each linear MPC. Then, each PWA continuous function defining the controller is approximated by a PWAS function. We obtain a discontinuous PWAS controller, which is implemented on a FPGA by using the architecture introduced in Sec. 3.2.

3.4.1 Model and control description

The state vector x represents two different temperatures, while the input u is the ambient temperature to be regulated:

$$x(k) \triangleq \begin{bmatrix} T_1(k) \\ T_2(k) \end{bmatrix} \;,\; u \triangleq T_{amb}$$

The auxiliary variables associated with threshold events u_{hot}, u_{cold} are such that

$$\begin{aligned} &\text{IF } x_1 \leq T_{c1} \text{ OR } (x_2 \leq T_{c2} \text{ AND } x_1 < T_{h1}) \\ &\qquad \text{THEN } u_{hot} = U_h \text{ , ELSE } u_{hot} = 0 \\ &\text{IF } x_1 \leq T_{h1} \text{ OR } (x_2 \leq T_{h2} \text{ AND } x_1 < T_{c1}) \\ &\qquad \text{THEN } u_{cold} = U_c \text{ , ELSE } u_{cold} = 0 \end{aligned} \tag{3.14}$$

where $T_{\{c1,c2,h1,h2\}}$ are constant temperatures, U_c represents the air conditioning power flow, U_h represents the heater flow.

The hybrid model is stated as follows.

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s[-\alpha_1(x_1(k) - u(k)) + \\ &\qquad + K_1(u_{hot}(k) - u_{cold}(k))] \\ x_2(k+1) &= x_2(k) + T_s[-\alpha_2(x_2(k) - u(k)) + \\ &\qquad + K_2(u_{hot}(k) - u_{cold}(k))] \end{aligned} \tag{3.15}$$

where $T_s = 0.5\ s$ is the sampling time and $\alpha_{\{1,2\}}$, $K_{\{1,2\}}$ $[s^{-1}]$ are constant coefficients. The output of the system is the state x.

The state and input constraints for the hybrid model are the following.

$$
\begin{aligned}
-10 \leq & u(k) \leq 50 \\
-10 \leq & x(k) \leq 50
\end{aligned} \tag{3.16}
$$

By exploiting the results of [2], the hybrid model (3.14),(3.15) is translated into an equivalent PWA model, which is defined over a three-dimensional domain ($n = 3$, with 2 state dimensions and 1 input dimension) partitioned into 5 polyhedral regions. As shown in Fig. 3.4, where a section of the PWA model for $T_{amb} = 25°C$ is shown, the polyhedral partition has boundaries parallel with respect to the state axes T_1 and T_2. Since the open-loop dynamics in regions 2, 3 and in 1, 4 are the same, the MPC calculated over the partition $\mathcal{P}$ associated to region 2 is equivalent to the one calculated in region 3, as well as region 1 shares the same MPC with region 4, although on different sets of states.

As described in Sec. 3.3, the affine terms in the PWA formulation are considered as constant measured input disturbances in the SwMPC formulation. The target of the controller is to track a reference r_y for $y = x_2$, while enforcing the constraint $x_1 \geq 25$ in addition to the constraints in (3.14), (3.16). The controllers (HMPC and SwMPC) share the same tuning parameters: $M = 4$, $Q_y = 1$, $R = 0$, $\rho = +\infty$ (corresponding to hard constraints). For a constant reference tracking $r_y = 30$, Figure 3.5 shows that the difference Δ_u between the manipulated variables in HMPC and SwMPC is negligible in most of the considered points. The controllers characteristics are summarized in Table 3.2.

3.4.2 FPGA implementation

The PWAS approximation of the SwMPC controller has been implemented on a Xilinx Spartan 3 FPGA, using the VHDL language to define the circuit.

The first step towards the FPGA implementation is the PWAS approximation of each explicit linear MPC by applying the method proposed in [10]. The result of the approximations are five continuous PWAS functions defined all over the domain, partitioned into simplices using $m_h = 7$ divisions along each dimensional component.

The second step is to merge the five continuous PWAS functions into one discontinuous PWAS function f_{PWAS}. Since there are two discontinuities along

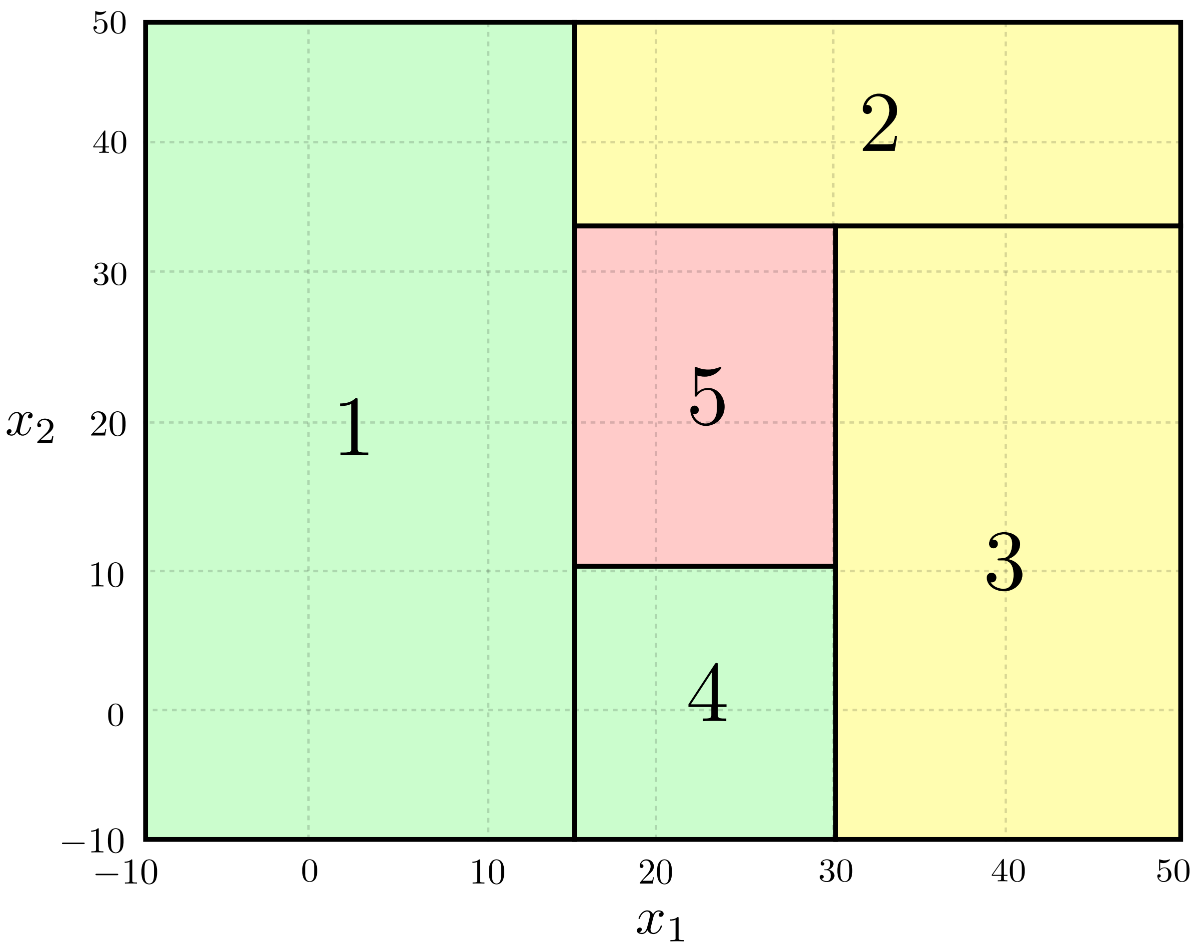

Figure 3.4: PWA state partitions for $u = 25°C$.

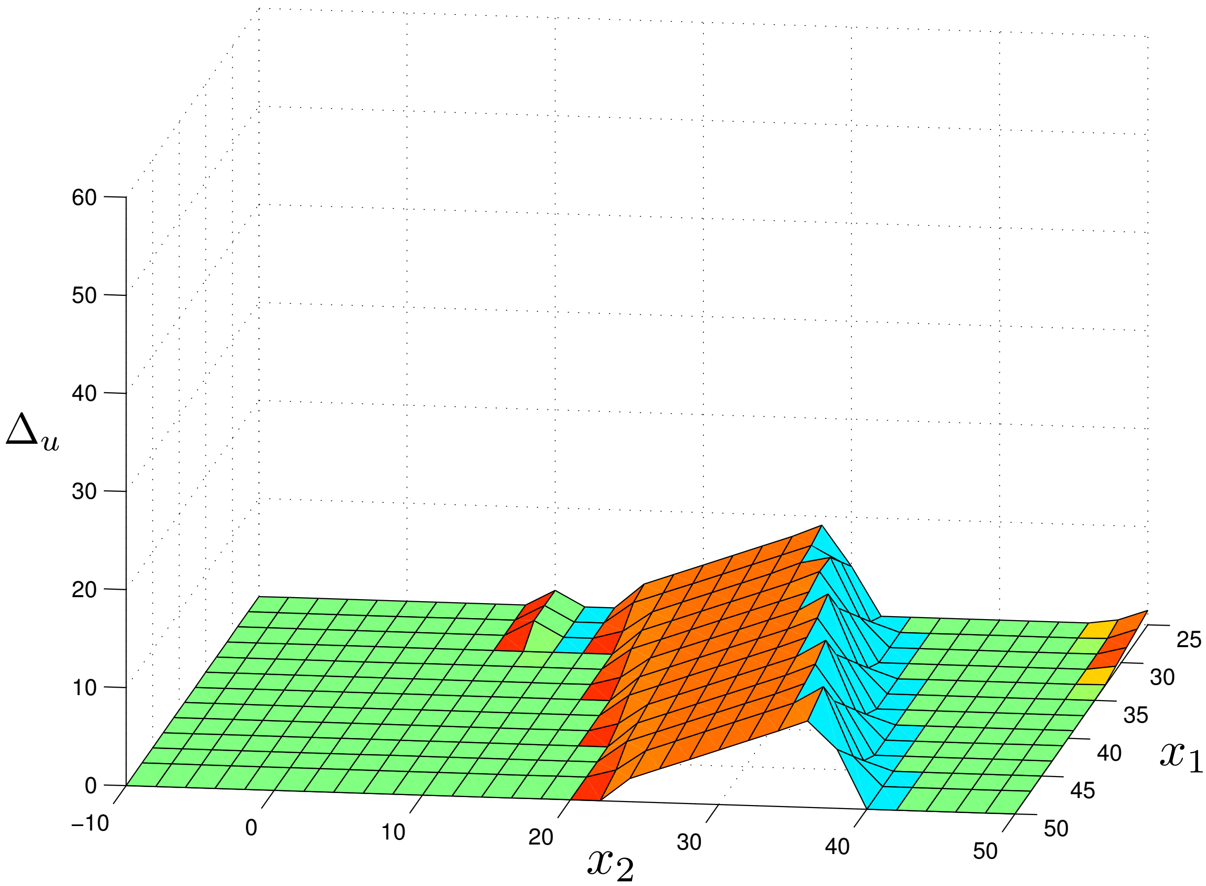

Figure 3.5: Differences between manipulated variables of HMPC and SwMPC for $r_y = 30$.

Item	HMPC	SwMPC (for each controller)
# params	3 (2 states, 1 reference)	3
# variables	36 (12 cont., 24 binary)	6
# inequalities	96 (mixed-integer)	26
# regions	N/A	5
# subregions explicit	1385[a]	10 (maximum), (35 total)

[a]Possible overlapping regions that are never optimal are not removed

Table 3.2: Controllers dimensions

the first and the second dimensions, the discontinuity partition is composed by nine hyper-rectangular subregions $\mathcal{R}_i$, $i = 1, \dots, 9$.

Fig. 3.6 shows the input and state signals obtained using the PWAS approximated control and the HMPC approach, where a sinusoidal reference r_y for x_2 is imposed.

Function f_{PWAS} is implemented by resorting to the proposed architecture. To implement the circuit architecture on FPGA we have used a fully generic VHDL description, so that it is possible to change any parameter before programming the FPGA without editing the VHDL code; these parameters are the input dimension n, the number of bits used to code the integer part (p) and decimal part (q) of the input and the number of bits (b) used to code the coefficients c_k^i. The position of the discontinuities $d_{h,t}$ inside the domain and the value of the coefficients c_k^i is set through a Matlab routine that allows the configuration of the whole VHDL code.

The estimated maximum working frequency is $40\ MHz$, that corresponds to a throughput of one sample every $550\ ns$, with a power consumption of $85\ mW$.

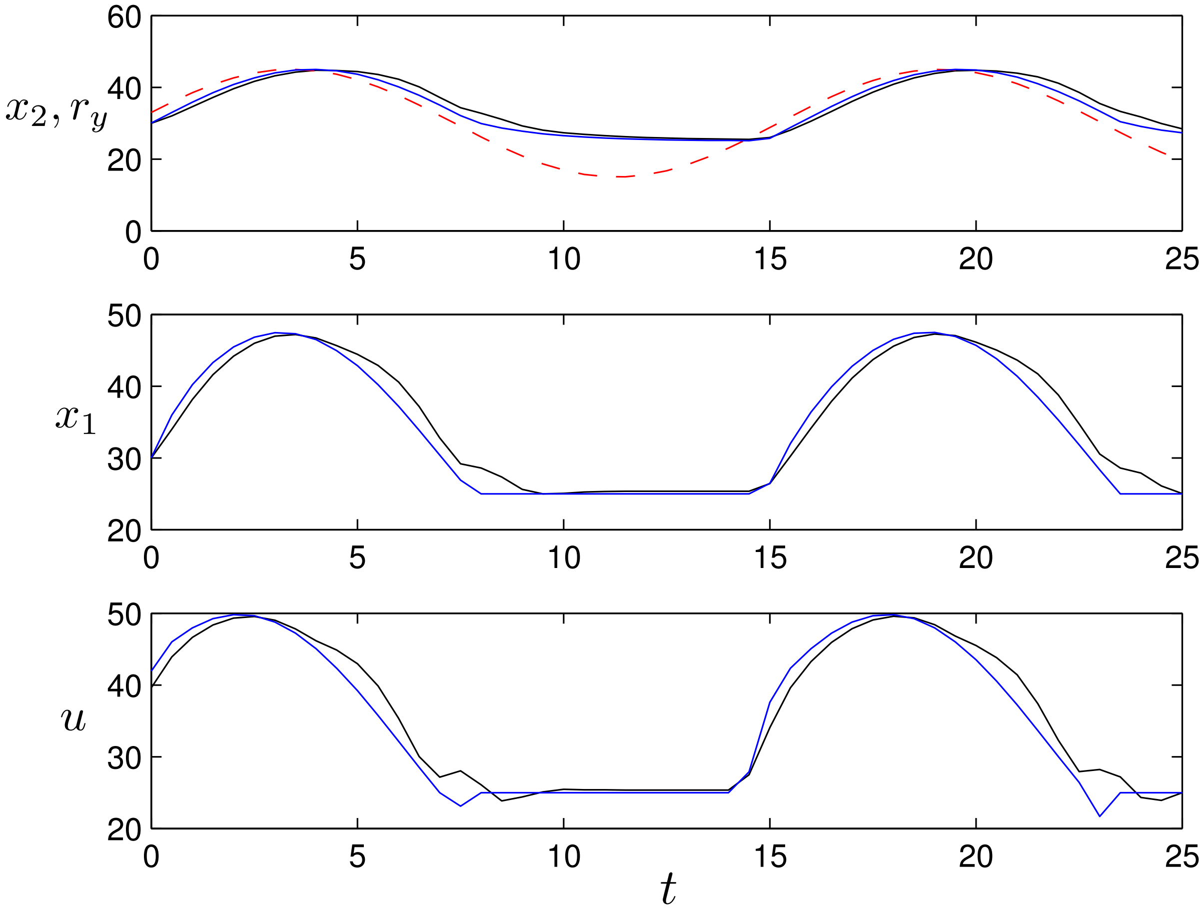

Figure 3.6: HMPC (blue) vs. PWAS (black).

The approximated discontinuous PWAS control occupies 69% of the hardware resources of the chosen FPGA.

Chapter 4

Closed-loop stability analysis

The contribution of this Chapter is the definition of a stability analysis framework for discrete-time PWA systems subject to both parametric uncertainties and additive disturbances which are unknown but bounded and defined in polytopic sets. The proposed method is based on the use of PWA Lyapunov functions synthesized via linear programming, and permits to determine if the state converges to the origin (or to a terminal set including the origin). The system dynamics are defined only in a closed polytopic region $\mathcal{X}$, which is not necessarily required to be invariant. By artificially extending the systems dynamics outside $\mathcal{X}$, the proposed method can determine an invariant subset of $\mathcal{X}$, in which the dynamics of the original PWA system of interest are defined. The attractiveness of the origin (or that of the terminal set) is determined with respect to such a region of attraction. Finally, discontinuities on the boundaries of the partitions are tackled for both the system dynamics and the PWA Lyapunov function, in order to broaden the range of applicability of the proposed approach and to reduce the conservativeness due to the imposition of continuity. The presence of discontinuities, however, requires additional attention on technical conditions [26].

Preliminary results of this thesis focusing only on the asymptotic stability analysis of systems without disturbances or with parametric disturbances are reported in [38] and in [42], respectively.

4.1 Problem formulation

Consider the autonomous discrete-time uncertain PWA system

$$x(k+1) = A_i(w(k))x(k) + a_i(w(k)) + E_i(w(k))d(k) \text{ if } x(k) \in \mathcal{X}_i \quad (4.1)$$

where $x(k) \in \mathbb{R}^n$, $w(k) \in \mathcal{W} \subset \mathbb{R}^q$, $d(k) \in \mathcal{D} \subset \mathbb{R}^p$,

$$A_i(w) \triangleq A_{i,0} + \sum_{r=1}^{q} A_{i,r} w_r \quad (4.2a)$$

$$a_i(w) \triangleq a_{i,0} + \sum_{r=1}^{q} a_{i,r} w_r \quad (4.2b)$$

$$E_i(w) \triangleq E_{i,0} + \sum_{r=1}^{q} E_{i,r} w_r \quad (4.2c)$$

$$\mathcal{W} \triangleq \left\{ w \in \mathbb{R}^q : \sum_{r=1}^{q} w_r = 1,\ w_r \geq 0 \right\} \quad (4.2d)$$

$$\mathcal{D} \triangleq \left\{ d \in \mathbb{R}^p : \tilde{H} d \leq \tilde{h} \right\} \quad (4.2e)$$

$A_{i,r} \in \mathbb{R}^{n\times n}$, $a_{i,r} \in \mathbb{R}^n$, $E_{i,r} \in \mathbb{R}^{n\times p}$, with $r = 0, ..., q$, and $k \in \mathbb{Z}_+$, $\tilde{H} \in \mathbb{R}^{p\times\eta}$, and $\tilde{h} \in \mathbb{R}^\eta$. Denote by $d_1, \ldots, d_\eta \in \mathbb{R}^p$ the vertices of $\mathcal{D}$, $\mathcal{D} = conv(d_1, \ldots, d_\eta)$. The sets $\mathcal{X}_i$, $i \in \mathcal{I} \triangleq \{1, ..., s\}$, are (possibly non-closed) polytopes such that $int(\mathcal{X}_i) \neq \emptyset$, $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$, $\forall i, j \in \mathcal{I}$ with $i \neq j$, and such that $\mathcal{X} \triangleq \bigcup_{i=1}^{s} \mathcal{X}_i$ is a closed polytope. The subset of indices $\mathcal{I}_0$ is defined as $\mathcal{I}_0 \triangleq \{i \in \mathcal{I} \ : \ 0 \in \bar{\mathcal{X}}_i\}$. The interior of each partition $\mathcal{X}_i$ is defined as

$$int(\mathcal{X}_i) \triangleq \{x : \ H_i x < h_i\},\ i \in \mathcal{I} \quad (4.3)$$

where H_i and h_i are a constant matrix and a constant vector, respectively, of suitable dimensions, and let $\bar{\mathcal{X}}_i$ the closure of $\mathcal{X}_i$, $\bar{\mathcal{X}}_i \triangleq \{x : H_i x \leq h_i\}$, $i \in \mathcal{I}$.

Denote by

$$x(k+1) = A_{i,0}x(k) + a_{i,0} \text{ if } x(k) \in \mathcal{X}_i \tag{4.4}$$

the *nominal* model of (4.1). Note that dynamics (4.1) may not be continuous with respect to x on the boundaries of the partitions $\mathcal{X}_i$, while it is continuous with respect to w and d.

Assumption 1. *There exists an index $i \in \mathcal{I}$ such that $0 \in$ vert$(\bar{\mathcal{X}}_i)$, $0 \in int(\mathcal{X})$.* □

Note that Assumption 1 can be always satisfied. In fact, if the origin is not on a vertex of any polyhedron $\mathcal{X}_i$, it is always possible to further partition $\mathcal{X}$ to obtain a new set of partitions $\mathcal{X}_i$ which fulfills Assumption 1. Note also that the state trajectories may not be persistent in time, since $\mathcal{X}$ is not necessarily an RPI set, and the dynamics are not defined outside $\mathcal{X}$.

This Chapter addresses the following problem: Given the uncertain PWA system (4.1), for which $\mathcal{X}$ is not necessarily an RPI set, prove the properties of stability and convergence to the origin (asymptotic stability, ultimate boundedness, and input-to-state stability) with respect to an RPI subset $\mathcal{P}$ of $\mathcal{X}$. In case of ultimate boundedness, find another RPI set $\mathcal{F}$ where the state is driven in finite time.

4.2 Reachability analysis

4.2.1 One-step reachability analysis

Since the set $\mathcal{X}$ is not assumed to be RPI with respect to dynamics (4.1), we must take into account that the trajectories may possibly leave $\mathcal{X}$, and be therefore defined only on a finite time interval $[0, k_{max}]$. Define the one-step reachable set from $\mathcal{X}$

$$\begin{aligned}\mathcal{R}(\mathcal{X}) \triangleq \{A_i(w)x + a_i(w) + E_i(w)d : w \in \mathcal{W},\ d \in \mathcal{D}, \\ x \in \mathcal{X}_i,\ i \in \mathcal{I}\}\end{aligned}$$

and let

$$\mathcal{R}_{\cup}(\mathcal{X}) \triangleq \mathcal{R}(\mathcal{X}) \cup \mathcal{X} \tag{4.5}$$

The set $\mathcal{R}(\mathcal{X})$ can be computed as the union of the one-step reachable sets from all the $\mathcal{X}_i$, defined as

$$\mathcal{R}(\mathcal{X}_i) \triangleq \{A_i(w)x + a_i(w) + E_i(w)d,\ w \in \mathcal{W}, d \in \mathcal{D}, x \in \mathcal{X}_i\}$$

Note that $\mathcal{R}(\mathcal{X}_i)$ is not a convex set in general. Note that the terms in (4.2a) and (4.2b) can be equivalently expressed as

$$\begin{aligned}A_i(w) &= \sum_{r=1}^{q} (A_{i,0} + A_{i,r})\, w_r \triangleq \sum_{r=1}^{q} \tilde{A}_{i,r} w_r \\ a_i(w) &= \sum_{r=1}^{q} (a_{i,0} + a_{i,r})\, w_r \triangleq \sum_{r=1}^{q} \tilde{a}_{i,r} w_r \\ E_i(w) &= \sum_{r=1}^{q} (E_{i,0} + E_{i,r})\, w_r \triangleq \sum_{r=1}^{q} \tilde{E}_{i,r} w_r\end{aligned}$$

By relying on the results in [12, Chap. 6], we can compute the convex hulls of the sets $\mathcal{R}(\bar{\mathcal{X}}_i)$ as

$$\begin{aligned}&conv\left(\mathcal{R}(\bar{\mathcal{X}}_i)\right) = \\ &\quad conv\left(\tilde{A}_{i,r}v_{i,h} + \tilde{a}_{i,r} + \tilde{E}_{i,r}d_{i,\mu},\ r = 1,...,q,\ \mu = 1,...,\eta,\ h = 1,...,m_i\right)\end{aligned}$$

where $v_{i,h}$ represents each of the m_i vertices of $\bar{\mathcal{X}}_i$.

Therefore, an over-approximation of $\mathcal{R}_\cup(\mathcal{X})$ in (4.5) is

$$\tilde{\mathcal{R}}_\cup(\mathcal{X}) \triangleq \bigcup_{i=1}^{s} \left(conv\left(\mathcal{R}(\bar{\mathcal{X}}_i)\right)\right) \cup \mathcal{X} \supseteq \mathcal{R}_\cup(\mathcal{X}) \tag{4.6}$$

4.2.2 Fake dynamics and extended system

As dynamics (4.1) is not defined outside $\mathcal{X}$, the proposed strategy consists in defining a "fake" dynamics on $\tilde{\mathcal{R}}_\cup(\mathcal{X}) \setminus \mathcal{X}$. Let $\mathcal{X}_H \supseteq \tilde{\mathcal{R}}_\cup(\mathcal{X})$ be the bounding box of $\tilde{\mathcal{R}}_\cup(\mathcal{X})$, i.e., the smallest closed hyper-rectangle containing $\tilde{\mathcal{R}}_\cup(\mathcal{X})$, and consider the dynamics

$$x(k+1) = \rho x(k),\ \text{if } x(k) \in \mathcal{X}_E \triangleq \mathcal{X}_H \setminus \mathcal{X} \tag{4.7}$$

where $\rho \in [0, 1)$ is an adjustable parameter of the approach proposed in this thesis. The region $\mathcal{X}_E$ can be divided into convex polyhedral regions as in [9, Th. 3]. As a result, new regions $\mathcal{X}_i$, $i = s+1,...,\tilde{s}$, are created. Let $\tilde{\mathcal{I}} \triangleq \{1,...,\tilde{s}\}$. The dynamics of the extended system on $\mathcal{X}_H$ is

$$x(k+1) = \begin{cases} A_i(w(k))x(k) + a_i(w(k)) + E_i(w(k))d(k) & \text{if } x(k) \in \mathcal{X}_i,\ i \in \mathcal{I} \\ \rho x(k) & \text{if } x(k) \in \mathcal{X}_E \end{cases} \tag{4.8}$$

Lemma 1. *The set $\mathcal{X}_H$ is an RPI set with respect to the extended dynamics* (4.8).

Proof. If $x \in \mathcal{X}_H$, then either $x \in \mathcal{X}$ or $x \in \mathcal{X}_E$. If $x \in \mathcal{X}$ then the successor state $A_i(w)x + a_i(w) + E_i(w)d \in \tilde{\mathcal{R}}_\cup(\mathcal{X}) \subseteq \mathcal{X}_H$ by definition of $\mathcal{X}_H$. If $x \in \mathcal{X}_E$, the successor state is $\rho x \in \mathcal{X}_H$, because $\mathcal{X}_H$ is a convex set including the origin. □

Defining $\mathcal{X}_H$ as a bounding box and the dynamics in $\mathcal{X}_E$ as in (4.7) is a simplistic choice, yet we will prove its effectiveness. Other choices of $\mathcal{X}_H$ and of the dynamics (4.7) are possible, provided that Lemma 1 holds.

Let $x(k) \in \mathcal{X}_i$ and $x(k+1) \in \mathcal{X}_j$, $(i,j) \in \tilde{\mathcal{I}} \times \tilde{\mathcal{I}}$. To characterize the transitions we define the *region transition map* $\mathcal{S}$

$$\mathcal{S}_{i,j} \triangleq \begin{cases} 1 & \text{if } conv\left(\mathcal{R}(\bar{\mathcal{X}}_i)\right) \cap \bar{\mathcal{X}}_j \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

which states (in a conservative way) whether there exists a state $x \in \bar{\mathcal{X}}_i$ and two uncertain vectors $w \in \mathcal{W}$, $d \in \mathcal{D}$ such that $A_i(w)x + a_i(w) + E_i(w)d \in \bar{\mathcal{X}}_j$. For any pair $(i,j) \in \tilde{\mathcal{I}} \times \tilde{\mathcal{I}}$, we define

$$\mathcal{X}_{i,j} \triangleq \begin{cases} \bar{\mathcal{X}}_i & \text{if } \mathcal{S}_{(i,j)} = 1 \\ \emptyset & \text{if } \mathcal{S}_{(i,j)} = 0 \end{cases} \tag{4.10}$$

that we refer to as *transition set*, representing an overestimate of all the states that can possibly end up in $\mathcal{X}_j$ in one step under dynamics i. In some particular cases it is possible to give less conservative estimates of such a set of states by using controllability analysis, as described in the following section.

4.2.3 Case of additive disturbances only

When only additive disturbances affect the system, (4.1) can be written as

$$x(k+1) = A_{i,0}x(k) + a_{i,0} + E_{i,0}d(k) \tag{4.11}$$

For system (4.11), $conv\left(\mathcal{R}(\bar{\mathcal{X}}_i)\right) = \mathcal{R}(\bar{\mathcal{X}}_i)$ (see e.g. [12, Chap. 6]) and then $\tilde{\mathcal{R}}_{\cup}(\mathcal{X}) = \mathcal{R}_{\cup}(\bar{\mathcal{X}})$. As for the definition of the transition sets, it is possible to determine the subset $\mathcal{X}_{i,j}$ of $\bar{\mathcal{X}}_i$ of states that reach $\bar{\mathcal{X}}_j$ in one step

$$\mathcal{X}_{i,j} \triangleq \left\{x \in \bar{\mathcal{X}}_i :\ \exists d \in \mathcal{D}_i : A_{i,0}x + a_{i,0} + E_{i,0}d \in \bar{\mathcal{X}}_j\right\} \tag{4.12}$$

We exploit here controllability analysis [12, Chap. 5], and consider the disturbance vector d as an external input, with respect to which the controllability analysis is carried out. In the augmented state space (x, d) let

$$\mathcal{M}_i(\bar{\mathcal{X}}_i, \bar{\mathcal{X}}_j) = \left\{(x, d) \in \mathbb{R}^{n+p} : x \in \bar{\mathcal{X}}_i,\ A_{i,0}x + a_{i,0} + E_{i,0}d \in \bar{\mathcal{X}}_j,\ d \in \mathcal{D}\right\}$$

that is computed as

$$\mathcal{M}_i(\bar{\mathcal{X}}_i, \bar{\mathcal{X}}_j) = \left\{(x, d) \in \mathbb{R}^{n+p} : \begin{bmatrix} H_i & 0 \\ H_j A_{i,0} & H_j E_{i,0} \\ 0 & \tilde{H} \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} 0 \\ H_j a_{i,0} \\ 0 \end{bmatrix} \leq \begin{bmatrix} h_i \\ h_j \\ \tilde{h} \end{bmatrix}\right\}$$

The intersection of $\bar{\mathcal{X}}_i$ with the so-called pre-image set of $\bar{\mathcal{X}}_j$, representing all the states that in one step reach $\bar{\mathcal{X}}_j$ under the dynamics and disturbances defined in $\mathcal{X}_i$ [12], is calculated as the projection of the set $\mathcal{M}_i(\bar{\mathcal{X}}_i, \bar{\mathcal{X}}_j)$ onto the state subspace:

$$\mathcal{X}_{i,j} = \left\{x :\ \exists d : (x, d) \in \mathcal{M}_i(\bar{\mathcal{X}}_i, \bar{\mathcal{X}}_j)\right\} \tag{4.13}$$

In case of no additive disturbances we have the nominal case, i.e., we have to prove asymptotic stability of the PWA system (4.4). For such a form it is possible to determine exactly which is the partition of $\bar{\mathcal{X}}_i$ which is mapped in $\bar{\mathcal{X}}_j$ in one step

$$\mathcal{X}_{i,j} = \left\{x \in \mathbb{R}^n : \begin{bmatrix} H_i \\ H_j A_{i,0} \end{bmatrix} x + \begin{bmatrix} 0 \\ H_j a_{i,0} \end{bmatrix} \leq \begin{bmatrix} h_i \\ h_j \end{bmatrix}\right\} \tag{4.14}$$

In both the considered subcases, the transition map (4.9) can be redefined as

$$\mathcal{S}_{i,j} \triangleq \begin{cases} 1 & \text{if } \mathcal{X}_{i,j} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

Note that a similar analysis in presence of a disturbance w would lead to a set of bilinear equations (x and w are multiplied to each other), leading to a non-polytopic shape for the regions $\mathcal{X}_{i,j}$.

4.3 PWA Lyapunov analysis for the extended system

In this section we analyze the asymptotic stability and the ultimate boundedness of the extended system (4.8).

4.3.1 Asymptotic stability

By recalling classical results of stability of nonlinear discrete-time systems (see e.g. [26], [43] and [25, Chap.2]), assume that the origin is an equilibrium point for (4.8). Lyapunov stability is guaranteed by the existence of a function $V : \mathcal{X}_H \to \mathbb{R}$ satisfying the conditions

$$V(x) \geq \alpha_1 \|x\|_\infty \tag{4.16a}$$

$$V(f(x, w, d)) - \lambda V(x) \leq 0 \tag{4.16b}$$

$\forall x \in \mathcal{X}_i$ and $\forall w \in \mathcal{W}$ ($i \in \tilde{\mathcal{I}}$), where $f : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^\eta \to \mathbb{R}^n$ is the PWA state update function defined by (4.8), $\alpha_1 > 0$, $\lambda \in (0, 1)$. Note that (4.16) imply the condition $V(0) = 0$. Considering (4.16b) in $x = 0$ and recalling that 0 is an equilibrium point, $f(0, w, d) = 0, \forall w \in \mathcal{W}$, $\forall d \in \mathcal{D}$, we get $V(0) - \lambda V(0) = (1 - \lambda)V(0) \leq 0$, which implies $V(0) \leq 0$, and by (4.16a), it follows that $V(0) = 0$.

Remark 1. *Condition* (4.16b) *could be replaced by*

$$V(f(x,w,d)) - V(x) \leq -\alpha_3\|x\| \tag{4.17}$$

where $\alpha_3 = (1-\lambda)\alpha_1 > 0$. *In fact, by* (4.16)*, it follows that* $V(f(x,w,d)) - V(x) \leq -(1-\lambda)V(x) \leq -(1-\lambda)\alpha_1\|x\|$. *Also, note that the imposition of an upperbound* $\alpha_2\|x\| \geq V(x)$, $\alpha_2 > 0$, *usually found in the literature is not necessary here, as V is defined over the bounded set* $\mathcal{X}_H$. *As a consequence, it is always possible to find a-posteriori* $\alpha_2 > 0$ *such that* $V(x) \leq \alpha_2\|x\|$, $\forall x \in \mathcal{X}_H$, *once function* V *has been determined.* □

Note that imposing the simpler decreasing condition $V(f(x,w,d)) - V(x) < 0$ for all $x \in \mathcal{X} \setminus 0$ (instead of (4.16b) or (4.17)) does not guarantee the asymptotic stability of the origin, since the system dynamics are allowed to be discontinuous on the boundaries of the partitions $\mathcal{X}_i$. Instead, the fulfillment of (4.16a) and (4.16b) (or (4.17)), even if the resulting Lyapunov function is discontinuous, according to [26], is a sufficient condition for the asymptotic stability of (4.8).

The goal is to synthesize a PWA Lyapunov function for system (4.8) satisfying (4.16). Consider the candidate function $V : \mathcal{X}_H \to \mathbb{R}$

$$V(x) = \max_{i \in \mathcal{N}(x)} V_i(x) \tag{4.18a}$$

where

$$\mathcal{N}(x) \triangleq \{i \in \tilde{\mathcal{I}} : \ x \in \bar{\mathcal{X}}_i\} \tag{4.18b}$$

and let $V_i : \bar{\mathcal{X}}_i \to \mathbb{R}$ be defined as

$$V_i(x) \triangleq F_i x + g_i \tag{4.18c}$$

for $i \in \tilde{\mathcal{I}}$, where in (4.18c) $F_i \in \mathbb{R}^{1\times n}$ and $g_i \in \mathbb{R}$ are coefficients to be determined. Note that simply $V(x) = F_i x + g_i$ for $x \in int(\mathcal{X}_i)$. The rationale for using the $\max$ in (4.18a) is that for numerical reasons we want to consider closed sets $\bar{\mathcal{X}}_i$ and $V_i(x)$, $V_j(x)$ may not coincide on common boundaries $\bar{\mathcal{X}}_i \cap \bar{\mathcal{X}}_j$ unless very conservative continuity conditions are imposed.

Since $\bar{\mathcal{X}}_i$ is a convex set and V_i is affine on the corresponding set $\bar{\mathcal{X}}_i$, it will be shown that it is enough to impose the Lyapunov conditions (4.16a) only at *vert*$(\bar{\mathcal{X}}_i)$, and (4.16b) only at *vert*$(\mathcal{X}_{i,j})$:

$$F_i v_{i,h} + g_i \geq \alpha_1 \|v_{i,h}\| \tag{4.19a}$$

for all m_i vertices $v_{i,h} \in$ *vert*$(\bar{\mathcal{X}}_i)$, $i \in \tilde{\mathcal{I}}$, $h = 1, \ldots, m_i$, together with

$$\alpha_1 > 0 \tag{4.19b}$$

and

$$F_j(\tilde{A}_{i,r} v_{ij,h} + \tilde{a}_{i,r}) + \tilde{E}_{i,r} d_\mu) + g_j - \lambda(F_i v_{ij,h} + g_i) \leq 0 \tag{4.19c}$$

for all $v_{ij,h} \in$ *vert*$(\mathcal{X}_{i,j})$, with $h = 1, \ldots, m_i$, for all $A_{i,r},\ a_{i,r},\ E_{i,r}$ with $r = 1, ..., q$, and all d_μ with $\mu = 1, ..., \eta$. Note that the set generated by the convex combination of the points $A_{i,r} v_{ij,h} + a_{i,r} + E_{i,r} d_\mu$ with respect to the vertices of $\bar{\mathcal{X}}_i$ coincides with $conv\,\{\mathcal{R}\,(\mathcal{X}_{i,j})\}$. As a consequence, to impose the decreasing condition (4.19c) one can consider only the vertices of $\mathcal{X}_{i,j}$ generating the vertices of $conv\,\{\mathcal{R}\,(\mathcal{X}_{i,j})\}$. The resulting constraints (4.19) define a linear feasibility problem in the unknowns F_i, g_i, α_1, for a fixed decay rate λ, and a feasible solution can be determined by linear programming (LP).

Lemma 2. *Let Assumption 1 hold, and let the LP* (4.19) *associated with the autonomous uncertain PWA dynamics* (4.8) *and the candidate Lyapunov function* (4.18) *be feasible. Then system* (4.8) *is* $\mathrm{UAS}(\mathcal{X}_H)$.

Proof. As for the positive definiteness of the Lyapunov function, since functions V_i are affine functions defined on convex partitions $\mathcal{X}_i$, the satisfaction of (4.19a) for all $v_{i,h} \in$ *vert*$(\bar{\mathcal{X}}_i)$, with $i \in \tilde{\mathcal{I}}$, $h = 1, \ldots, m_i$, for $x \in \bar{\mathcal{X}}_i$ leads to

$$\begin{aligned}\alpha_1\|x\| &= \alpha_1 \left\| \sum_{h=1}^{m_i} \beta_{i,h} v_{i,h} \right\| \leq \sum_{h=1}^{m_i} \beta_{i,h} \alpha_1 \|v_{i,h}\| \\ &\leq \sum_{h=1}^{m_i} \beta_{i,h}(F_i v_{i,h} + g_i) = F_i(\sum_{h=1}^{m_i} \beta_{i,h} v_{i,h}) + g_i \sum_{h=1}^{m_i} \beta_{i,h} = F_i x + g_i \end{aligned} \tag{4.20}$$

where $\beta_{i,h} \geq 0$, $\sum_{h=1}^{m_i} \beta_{i,h} = 1$, are a set of coefficients defining x as a convex combination of the vertices of $\bar{\mathcal{X}}_i$. For this reason, for $x \in int(\mathcal{X}_i)$, since $V_i(x) = F_i x + g_i$, (4.16a) holds. Moreover, on the boundaries of $\bar{\mathcal{X}}_i$, according to (4.18a), one has $\alpha_1\|x\| \leq F_i x + g_i$ for all $i \in \mathcal{N}(x)$, and therefore $\alpha_1\|x\| \leq \max_{i \in \mathcal{N}(x)}\{F_i x + g_i\} = V(x)$. This implies that (4.16a) holds for all $x \in \mathcal{X}_H$, since $\mathcal{X}_H = \bigcup_{i \in \tilde{\mathcal{I}}} \bar{\mathcal{X}}_i$.

As for the decay of the Lyapunov function, we have that

$$\begin{aligned}V(f(x,w,d)) = F_j \Bigg[&\left(\sum_{r=1}^{q} \tilde{A}_{i,r} w_r\right)\left(\sum_{h=1}^{m_i} \beta_{i,h} v_{ij,h}\right) + \sum_{r=1}^{q} \tilde{a}_{i,r} w_r \\ &+ \left(\sum_{r=1}^{q} \tilde{E}_{i,r} w_r\right)\left(\sum_{\mu=1}^{\eta} c_\mu d_\mu\right)\Bigg] + g_j \end{aligned} \tag{4.21}$$

with $c_\mu \geq 0$, $\sum_{\mu=1}^{\eta} c_\mu = 1$. Recalling that (4.19c) holds for all the vertices of

$\mathcal{X}_{ij}$, and that $\sum_{h=1}^{m_i} \beta_{i,h} = \sum_{\mu=1}^{\eta} c_h = \sum_{r=1}^{q} w_r = 1$, from (4.21) we get

$$\begin{aligned}
V(f(x,w,d)) =& F_j \left[\sum_{h=1}^{m_i} \beta_{i,h} \left(\sum_{r=1}^{q} \tilde{A}_{i,r} w_r v_{ij,h} \right) + \sum_{r=1}^{q} \tilde{a}_{i,r} w_r \right. \\
& \left. + \left(\sum_{r=1}^{q} \tilde{E}_{i,r} w_r \right) \left(\sum_{\mu=1}^{\eta} c_\mu d_\mu \right) \right] + g_j \\
=& F_j \left[\sum_{h=1}^{m_i} \beta_{i,h} \left(\sum_{r=1}^{q} \tilde{A}_{i,r} w_r v_{ij,h} \right) + \sum_{h=1}^{m_i} \beta_{i,h} \left(\sum_{r=1}^{q} \tilde{a}_{i,r} w_r \right) \right. \\
& \left. + \sum_{h=1}^{m_i} \beta_{i,h} \left(\sum_{\mu=1}^{\eta} c_\mu \left(\sum_{r=1}^{q} \tilde{E}_{i,r} w_r d_\mu \right) \right) \right] + g_j \\
=& F_j \sum_{h=1}^{m_i} \beta_{i,h} \left[\sum_{r=1}^{q} w_r \left(\tilde{A}_{i,r} v_{ij,h} + \tilde{a}_{i,r} + \sum_{\mu=1}^{\eta} c_\mu \tilde{E}_{i,r} d_\mu \right) \right] + g_j \\
=& F_j \sum_{h=1}^{m_i} \beta_{i,h} \left[\sum_{r=1}^{q} w_r \left(\sum_{\mu=1}^{\eta} c_\mu \tilde{A}_{i,r} v_{ij,h} + \sum_{\mu=1}^{\eta} c_\mu \tilde{a}_{i,r} + \sum_{\mu=1}^{\eta} c_\mu \tilde{E}_{i,r} d_\mu \right) \right] + g_j \\
=& \sum_{h=1}^{m_i} \beta_{i,h} \left[\sum_{r=1}^{q} w_r \sum_{\mu=1}^{\eta} c_\mu \left(F_j \left(\tilde{A}_{i,r} v_{ij,h} + \tilde{a}_{i,r} + \tilde{E}_{i,r} d_\mu \right) \right) \right] + g_j \\
\leq& \sum_{h=1}^{m_i} \beta_{i,h} \left[\sum_{r=1}^{q} w_r \sum_{\mu=1}^{\eta} c_\mu \left(\lambda \left(F_i v_{ij,h} + g_i \right) - g_j \right) \right] + g_j \\
=& \lambda \sum_{h=1}^{m_i} \beta_{i,h} \left(F_i v_{ij,h} + g_i \right) - \sum_{h=1}^{m_i} \beta_{i,h} g_j + g_j \\
=& \lambda \left(F_i \sum_{h=1}^{m_i} \beta_{i,h} v_{ij,h} + \sum_{h=1}^{m_i} \beta_{i,h} g_i \right) = \lambda \left(F_i x + g_i \right) = \lambda V(x)
\end{aligned}$$

which proves that (4.16b) holds for all $x \in int\left(\mathcal{X}_{i,j}\right)$. Also, on the boundaries of $\bar{\mathcal{X}}_i$, the decreasing condition (4.16b) is imposed for all $(i,j) \in \mathcal{N}(x) \times$

$\mathcal{N}(f(x,w,d))$, and therefore

$$\max_{j\in\mathcal{N}(f(x,w,d))}\Big(F_j(f(x,w,d))+g_j\Big)\le\lambda\max_{i\in\mathcal{N}(x)}(F_ix+g_i)$$

Since by definition of $\mathcal{X}_{i,j}$ we have that $\mathcal{X}_H=\bigcup_{i=1}^{\tilde{s}}\bigcup_{j=1}^{\tilde{s}}\mathcal{X}_{i,j}$, (4.16b) holds for all $x\in\mathcal{X}_H$. As a result, (4.16) hold for all $x\in\mathcal{X}_H$, which, following in spirit the proof of Theorem 2.2.1 in [25, Chap. 2] for deterministic systems, guarantees that system (4.8) is $\mathrm{UAS}(\mathcal{X}_H)$ according to Definition 3. □

Remark 2. *The complexity of the LP* (4.19) *is the following: there are $n_v=1+\tilde{s}(n+1)$ variables; one inequality is imposed for each vertex of each region $\mathcal{X}_i$, $i=1,...,\tilde{s}$ to fulfill* (4.19a)*; condition* (4.19b) *introduces one inequality constraint; to fulfill* (4.19c)*, for each vertex of each region one has to impose a number of inequalities equal to the number of regions $\mathcal{X}_j$ such that $\mathcal{S}_{i,j}\neq 0$, multiplied by the product $q\cdot\eta$; the overall number of constraints is therefore*

$$n_c=1+\sum_{i=1}^{\tilde{s}}m_i\left(1+q\cdot\eta\cdot card\left(\left\{j\in\tilde{\mathcal{I}}:\mathcal{S}_{i,j}=1\right\}\right)\right)$$

□

Note that asymptotic stability can be proved only if in the regions including the origin in their closure ($0\in\bar{\mathcal{X}_i}$) the additive disturbance has no effect, $E_i(w)=0$, and the affine terms $a_{i,r}=0,\ \forall r=1,...,q$.

4.3.2 Ultimate boundedness

To check ultimate boundedness, we define the following function

$$V(x)\ge\alpha_1\|x\|_\infty \tag{4.22a}$$

$$V(0)=0 \tag{4.22b}$$

$$V(f(x,w,d)) - \lambda V(x) \leq \delta \tag{4.22c}$$

where $\delta \geq 0$, while all the other terms are the same as in (4.16). To study ultimate boundedness, infinity norms will be used instead of the generic p-norms. We call a function $V : \mathbb{R}^n \to \mathbb{R}$ satisfying (4.22) a "Lyapunov-like" function. Similarly to (4.19), the constraints

$$F_i v_{i,h} + g_i \geq \alpha_1 \|v_{i,h}\|_\infty \tag{4.23a}$$

are imposed for all m_i vertices $v_{i,h} \in$ *vert*$(\bar{\mathcal{X}}_i)$, $i \in \tilde{\mathcal{I}}$, $h = 1, \ldots, m_i$,

$$F_j(\tilde{A}_{i,r} v_{ij,h} + \tilde{a}_{i,r} + \tilde{E}_{i,r} d_{i,\mu}) + g_j - \lambda(F_i v_{ij,h} + g_i) \leq \delta \tag{4.23b}$$

for all $v_{ij,h} \in$ *vert*$(\mathcal{X}_{i,j})$, with $h = 1, \ldots, m_i$, for all $\tilde{A}_{i,r},\ \tilde{a}_{i,r},\ \tilde{E}_{i,r}$ with $r = 1, ..., q$, and all d_μ with $\mu = 1, ..., \eta$. The further constraints

$$\delta \geq 0 \tag{4.23c}$$

$$\delta \leq (1-\lambda)(L-\varepsilon)\alpha_1 \tag{4.23d}$$

$$g_i = 0,\ i \in \mathcal{I}_0 \tag{4.23e}$$

$$\alpha_1 > 0 \tag{4.23f}$$

are also imposed, where L is the minimum distance between the boundary of the hyper-rectangle $\mathcal{X}_H$ and the origin, while $0 < \varepsilon \ll 1$. Condition (4.23d) is imposed to ensure that the set where the state is ultimately bounded is well defined, while (4.23e) ensures the fulfillment of (4.22b). The overall optimization vector is denoted by ξ, and is composed by δ, α_1, and the terms F_i and g_i, with $i \in \tilde{\mathcal{I}}$.

Lemma 3. *Let Assumption 1 hold, and assume that the linear-fractional pro-*

gram

$$\text{minimize } \frac{\delta}{\alpha_1} \tag{4.24a}$$

$$\text{subject to (4.23)} \tag{4.24b}$$

associated with the autonomous uncertain PWA dynamics (4.8) *and the candidate Lyapunov-like function* (4.18) *is feasible. Then system* (4.8) *is* $\mathrm{UUB}(\mathcal{X}_H, \mathcal{F})$*, with*

$$\mathcal{F} \triangleq \{x \in \mathcal{X}_H : V(x) \leq V_{\mathcal{F}}^{+}\} \tag{4.25}$$

where

$$V_{\mathcal{F}}^{+} = \max_{x \in conv\{\mathcal{R}(\mathcal{Q})\}} V(x) \tag{4.26}$$

$$\mathcal{Q} = \left\{ x \in \mathbb{R}^n : \|x\|_\infty \leq \frac{\delta}{(1-\lambda)\alpha_1} + \varepsilon \right\} \tag{4.27}$$

Proof. As for the positive definiteness of the Lyapunov-like function V, the proof is exactly the same as in Lemma 2, in the particular case of infinity norms. To prove (4.22c) is fulfilled for all $x \in \mathcal{X}_H$, as in Lemma 2, consider the following relation

$$V(f(x,w,d)) = \sum_{h=1}^{m_i} \beta_{i,h} \left[\sum_{r=1}^{q} w_r \sum_{\mu=1}^{\eta} c_\mu \left(F_j \left(\tilde{A}_{i,r} v_{ij,h} + \tilde{a}_{i,r} + \tilde{E}_{i,r} d_{i,\mu} \right) \right) \right] + g_j \tag{4.28}$$

By applying the decreasing condition (4.23b) at the vertices of the sets $\mathcal{X}_{i,j}$, we obtain

$$\begin{aligned}
V(f(x,w,d)) \leq & \sum_{h=1}^{m_i} \beta_{i,h} \left[\sum_{r=1}^{q} w_r \sum_{\mu=1}^{\eta} c_\mu \left(\delta + \lambda \left(F_i v_{ij,h} + g_i \right) - g_j \right) \right] + g_j \\
= & \delta + \lambda \sum_{h=1}^{m_i} \beta_{i,h} \left(F_i v_{ij,h} + g_i \right) - \sum_{h=1}^{m_i} \beta_{i,h} g_j + g_j \\
= & \delta + \lambda \left(F_i \sum_{h=1}^{m_i} \beta_{i,h} v_{ij,h} + \sum_{h=1}^{m_i} \beta_{i,h} g_i \right) \\
= & \delta + \lambda \left(F_i x + g_i \right) \\
= & \delta + \lambda V(x)
\end{aligned}$$

which proves that (4.22c) holds for all $x \in int\left(\mathcal{X}_{i,j}\right)$. Also, on the boundaries of the sets $\bar{\mathcal{X}}_i$, the decreasing condition (4.22c) is imposed for all $(i,j) \in \mathcal{N}(x) \times \mathcal{N}(f(x,w,d))$, and therefore

$$\max_{j \in \mathcal{N}(f(x,w,d))} \Big(F_j(f(x,w,d)) + g_j \Big) \leq \delta + \lambda \max_{i \in \mathcal{N}(x)} \left(F_i x + g_i \right)$$

Since by definition of the sets $\mathcal{X}_{i,j}$ we have that $\mathcal{X}_H = \bigcup_{i=1}^{\tilde{s}} \bigcup_{j=1}^{\tilde{s}} \mathcal{X}_{i,j}$, (4.22c) holds for all $x \in \mathcal{X}_H$, and then (4.22) holds for all $x \in \mathcal{X}_H$.

By (4.22a) and (4.22c) it follows that $V(f(x,w,d)) - V(x) \leq -(1-\lambda)V(x) + \delta \leq -(1-\lambda)\alpha_1 \|x\|_\infty + \delta = -\alpha_3 \|x\|_\infty + \delta$, where $\alpha_3 = (1-\lambda)\alpha_1$. By definition of $\mathcal{Q}$ in (4.27), for all $x \in \mathcal{X}_H \setminus \mathcal{Q}$ we obtain $-\alpha_3\|x\|_\infty + \delta < -\alpha_3 \varepsilon$, leading to $V(f(x,w,d)) - V(x) < -\alpha_3\varepsilon$. Then, given $\lambda_* \in \mathbb{R}$, $V(f(x,w,d)) - \lambda_* V(x) < (1-\lambda_*)V(x) - \alpha_3 \varepsilon \leq (1-\lambda_*)V_H^+ - \alpha_3\varepsilon$, with $V_H^+ = \max_{x \in \mathcal{X}_H} V(x)$. In order to get $(1-\lambda_*)V_H^+ - \alpha_3\varepsilon \leq 0$ and $\lambda_* \in (0,1)$, it is necessary to impose the condition on λ_* $\max\{0, (V_H^+ - \alpha_3\varepsilon)/V_H^+\} < \lambda_* < 1$, which is always feasible, since $(V_H^+ - \alpha_3\varepsilon)/V_H^+ < 1$. In conclusion, we get

$$V(f(x,w,d)) - \lambda_* V(x) < 0, \; x \in \mathcal{X}_H \setminus \mathcal{Q} \tag{4.29}$$

By definition of $\mathcal{F}$, the fulfillment of the constraint (4.23d) implies $\mathcal{Q} \subseteq \mathcal{X}_H$ and, by relying on the invariance of $\mathcal{X}_H$ (Lemma 1), $conv\,\{\mathcal{R}(\mathcal{Q})\} \subseteq \mathcal{X}_H$. This implies the $\max$ in (4.26) exists and therefore $\mathcal{F}$ in (4.25) is well defined.

To prove invariance of $\mathcal{F}$, we distinguish two cases:

- if $x \in \mathcal{F} \cap \mathcal{Q}$, then $f(x, w, d) \in conv\,\{\mathcal{R}(\mathcal{Q})\} \subseteq \mathcal{F}$;
- if $x \in \mathcal{F} \setminus \mathcal{Q}$, then (4.29) holds, because $\mathcal{F} \setminus \mathcal{Q} \subseteq \mathcal{X}_H \setminus \mathcal{Q}$, and therefore $V(f(x, w, d)) \leq V_{\mathcal{F}}^{+}$, meaning that $f(x, w, d) \in \mathcal{F}$ by definition of $\mathcal{F}$.

Therefore, $\mathcal{F}$ is an RPI set for (4.8).

Let $V_H^- \triangleq \inf_{x \in \mathcal{X}_H \setminus \mathcal{Q}} V(x) > 0$. Then,

$$\forall\, x(0) \in \mathcal{X}_H \setminus \mathcal{Q} \quad \exists\, \bar{k} > 0 \;:\; V(x(\bar{k})) < \lambda_*^{\bar{k}} V(x(0)) < V_H^- \tag{4.30}$$

meaning that $x(\bar{k}) \in \mathcal{Q}$, because $\mathcal{X}_H$ in an RPI set.

Given $a \in \mathbb{R}$ such that $a > \delta/\alpha_3 + \varepsilon$, let $V_a^+ \triangleq \max_{x(0) \in \mathcal{X}_H \setminus \mathcal{Q} : \|x(0)\|_\infty \leq a} V(x(0))$. From (4.30), there exists $\bar{k}$ such that $\lambda_*^{\bar{k}} V(x(0)) \leq \lambda_*^{\bar{k}} V_a^+ < V_H^-$. By letting

$$\tilde{k}(a) \triangleq \left\lfloor \frac{\ln V_H^- - \ln V_a^+}{\ln \lambda_*} \right\rfloor \tag{4.31}$$

we obtain that for all $x(0) \in \mathcal{X}_H \setminus \mathcal{Q}$ with $\|x(0)\| \leq a$, one has $\bar{k} \leq \tilde{k}(a)$. As a consequence, for any initial condition $x(0) \in \mathcal{X}_H \setminus \mathcal{Q}$ such that $\|x(0)\| \leq a$, one has $x(\bar{k} + 1) \in \mathcal{F}$, with $\bar{k} \leq \tilde{k}(a)$. Being $\mathcal{F}$ an RPI set, $x(k) \in \mathcal{F}$ for $k \geq \tilde{k}(a) + 1$.

To finally prove that system (4.1) is $\mathrm{UUB}(\mathcal{X}_H, \mathcal{F})$, consider a scalar $a > 0$, and distinguish two cases:

- for all a such that $0 < a \leq \delta/\alpha_3 + \varepsilon$, there exists $T(a) = 1$ such that, for every $x(0)$ with $\|x(0)\|_\infty \leq a$ (i.e. $x(0) \in \mathcal{Q}$), $x(T) \in \mathcal{F}$ (by definition of $\mathcal{F}$);

- for all $a > \delta/\alpha_3 + \varepsilon$, there exists $T(a) = \tilde{k}(a) + 1$ such that, for every $x(0)$ with $\|x(0)\|_\infty \leq a$, $x(T) \in \mathcal{F}$.

In conclusion, for all $a > 0$, there exists $T = T(a)$ such that, for every $x(0) \in \mathcal{X}_H$ with $\|x(0)\|_\infty \leq a$, $x(T) \in \mathcal{F}$ for any admissible sequence of the disturbance terms w and d. This implies that system (4.1) is $\mathrm{UUB}(\mathcal{X}_H, \mathcal{F})$ according to Definition 4. $\square$

As a practical procedure to calculate $\mathcal{F}$, it is possible to partition $\mathcal{Q}$ into a number of subsets $\mathcal{Q}_i$, $\mathcal{Q}_i = \bar{\mathcal{X}}_i \cap \mathcal{Q}$ with $i \in \tilde{\mathcal{I}}$. Then, one can obtain $conv\,\{\mathcal{R}(\mathcal{Q})\} = \bigcup_{i=1}^{\tilde{s}} conv\,\{\mathcal{R}(\mathcal{Q}_i)\}$, with

$$
\begin{aligned}
&conv\,\{\mathcal{R}(\mathcal{Q}_i)\} = \\
&conv\left\{\tilde{A}_{i,r}v_{i,h}^{\mathcal{Q}} + \tilde{a}_{i,r} + \tilde{E}_{i,r}d_{i,\mu},\ r = 1, ..., q_i,\ \mu = 1, ..., \eta_i,\ h = 1, ..., m_i^{\mathcal{Q}}\right\}
\end{aligned}
$$

where $v_{i,h}^{\mathcal{Q}}$ are the vertices of the sets $\mathcal{Q}_i$, and $m_i^{\mathcal{Q}}$ is the number of vertices of each $\mathcal{Q}_i$. Then, $\mathcal{F}$ can be easily computed as the union of the subsets of the $\mathcal{X}_i$ for which $V(x) \leq \hat{V}$, i.e.,

$$
\mathcal{F} = \bigcup_{i=1}^{\tilde{s}} \mathcal{X}_i^{\mathcal{F}},\ \mathcal{X}_i^{\mathcal{F}} \triangleq \{x \in \mathcal{X}_i :\ V(x) < \hat{V}\} \tag{4.32}
$$

Remark 3. *The aim of* (4.24a) *is to minimize the volume of the hypercube $\mathcal{Q}$ for the given choice of the partitions, in order to obtain a set $\mathcal{F}$ that is as small as possible. Note that* (4.24) *is a quasi-convex optimization problem. Even though quasi-convex optimization problems are solved in general by bisection (solving a convex optimization problem at each step), linear-fractional programs like* (4.24) *can be transformed into an equivalent LP using the following procedure (see [16, Chap. 4] for a formal proof).*

Assume that δ and α_1 are, respectively, the first and second components of the optimization vector ξ in (4.24)*. Let $M_{eq}\xi = 0$ denote the whole set of equality constraints* (4.23e) *and $M_{in}\xi \leq m_{in}$ the whole set of inequality constraints* (4.23a)-(4.23d)*, where M_{eq}, M_{in}, and m_{in} can be easily constructed from* (4.23)*. Moreover, rewrite* (4.23f) *as $e'\xi > 0$, with $e = [0\ 1\ 0\ \dots\ 0]'$. Problem* (4.24) *is therefore equivalent to the following linear-fractional program*

$$\begin{aligned} \text{minimize}\ & \frac{c'\xi}{e'\xi} \\ \text{subject to}\ & M_{in}\xi \leq m_{in} \\ & M_{eq}\xi = 0 \\ & e'\xi > 0 \end{aligned} \tag{4.33}$$

where $c = [1\ 0\ \dots\ 0]'$. If the feasible set of (4.33) *is not empty, an equivalent LP of* (4.33) *is*

$$\begin{aligned} \text{minimize}\ & c'y \\ \text{subject to}\ & M_{in}y - m_{in}z \leq 0 \\ & M_{eq}y = 0 \\ & e'y = 1 \\ & z \geq 0 \end{aligned} \tag{4.34}$$

where y has the same dimension as ξ and $z \in \mathbb{R}$. The optimal vector of (4.24) *is $\xi = \frac{1}{z}y$.*

Remark 4. *The number of variables and constraints in* (4.34) *is comparable with that of* (4.19)*. In particular, two more variables are added, leading to $n_v = 3 + \tilde{s}(n+1)$. To define $M_{eq}y = 0$, one equality constraint is imposed for each region $\mathcal{X}_i$ such that $0 \in \bar{\mathcal{X}}_i$. One inequality is imposed for each vertex* *of* $\mathcal{X}_i$, $i = 1, ..., \tilde{s}$ *to fulfill* (4.23a). *To fulfill* (4.23b), *for each vertex of each region it is necessary to impose a number of inequalities equal to the number of regions* $\mathcal{X}_j$ *such that* $\mathcal{S}_{i,j} \neq 0$, *multiplied by* $q \cdot \eta$. *Condition* (4.23c) *and* (4.23d) *introduce two more inequality constraints, while two other constraints are introduced in the auxiliary problem* (4.34). *The overall number of constraints is overall number of scalar constraints is*

$$n_c = 4 + card(\mathcal{I}_0) + \sum_{i=1}^{\tilde{s}} m_i \left(1 + q \cdot \eta \cdot card\left(\left\{j \in \tilde{\mathcal{I}} : \mathcal{S}_{i,j} = 1\right\}\right)\right)$$

□

4.3.3 Feasibility issues

In case (4.19) or (4.23) is infeasible, besides increasing the value of λ, a possibility is to increase the number of partitions $\mathcal{X}_i$ of $\mathcal{X}_H$, therefore providing more flexibility in synthesizing the PWA Lyapunov (or Lyapunov-like) function.

PWQ Lyapunov function synthesis approaches [20] assume that V is quadratic on each cell $\mathcal{X}_i$. On the other hand, assuming that V is affine on each $\mathcal{X}_i$ would not provide enough degrees of freedom. Therefore, for each polyhedron $\mathcal{X}_i$, its Delaunay triangulation [44] $\{\mathcal{X}^D_{i,1}, \ldots, \mathcal{X}^D_{i,n_i}\}$, $i \in \tilde{s}$ is computed. The PWA Lyapunov synthesis procedure is performed by replacing the sets $\mathcal{X}_i$ with the elements of the simplicial partition $\{\mathcal{X}^D_{1,1}, \ldots, \mathcal{X}^D_{1,n_1}, \mathcal{X}^D_{2,1}, \ldots, \mathcal{X}^D_{2,n_2}, \ldots, \mathcal{X}^D_{\tilde{s},1}, \ldots, \mathcal{X}^D_{\tilde{s},n_{\tilde{s}}}\}$, and consequently by setting $\tilde{\mathcal{I}} = \{1, \ldots, \sum_{i=1}^{\tilde{s}} n_i\}$, and $\mathcal{I}$ the subset of $\tilde{\mathcal{I}}$ of indices for which $\mathcal{X}_i \subseteq \mathcal{X}$.

An alternative way, which is only applicable in the absence of multiplicative disturbances (Section 4.2.3), is to consider the sets $\mathcal{X}_{i,j}$ as the new sets $\bar{\mathcal{X}}_i$ and restart the procedure. Alternatively, in case $\mathcal{X}_i$ are simplices, one can split each of them into $n+1$ new simplices by considering the midpoint $\bar{v} = \frac{1}{n+1}\sum_{i=0}^{n} v_i$ as a new vertex. Note that by iterating such procedures, the complexity of the LP (4.19) or (4.23) grows quite fast.

4.4 Invariance analysis

So far the properties of the extended system (4.8) were analyzed. As this includes the fake dynamics (4.7), we want to derive now conditions on the original system (4.1). Consider again system (4.8) in $\mathcal{X}_H$, assume that a feasible solution to (4.19) or to (4.23) exists, define

$$V_E^- \triangleq \inf_{x \in \mathcal{X}_E} V(x) \tag{4.35}$$

and consider the subset $\mathcal{P}$ of $\mathcal{X}$

$$\mathcal{P} \triangleq \{x \in \mathcal{X} :\ V(x) < V_E^-\} \tag{4.36}$$

Note that the set $\mathcal{P}$ may not be convex, not even connected.

The results for the extended system (4.8) proved in Lemmas 2 and 3 and the definition of $\mathcal{P}$ in (4.36) are exploited next to state the main results of the thesis.

Theorem 1. *Consider system* (4.1)*, whose dynamics are defined on $\mathcal{X}$, and assume that the extended dynamics* (4.7) *are defined in $\mathcal{X}_E$. If a Lyapunov function for system* (4.8) *is found by solving the LP* (4.19)*, $\mathcal{P} \subseteq \mathcal{X}$ defined in* (4.36) *is an RPI set for* (4.1)*. Moreover,* (4.1) *is* $\mathrm{UAS}(\mathcal{P})$. □

Proof. The proof consists in showing that the PWA Lyapunov function

$$V_{\mathrm{UAS}}(x) \triangleq V(x),\ \forall x \in \mathcal{P} \tag{4.37}$$

where $V(x)$ is found as in Lemma 2 for the extended system (4.8) in $\mathcal{X}_H$, is a Lyapunov function for (4.1) over the set $\mathcal{P}$. First of all, considering that $\mathcal{P}$ is an RPI set for (4.8) in $\mathcal{X}_H$, one can note that the state update $f(x, w, d) \in \mathcal{P}$ for $x \in \mathcal{P}$ is always given by (4.1). Then, $\mathcal{P}$ is an RPI set for (4.1), because the dynamics (4.7) is never executed. Considering that $\mathcal{P} \subseteq \mathcal{X}_H$, if (4.16) are satisfied for all $x \in \mathcal{X}_H$ (and then for all $x \in \mathcal{P}$), we conclude that $V_{\mathrm{UAS}}(x)$ is a Lyapunov function for system (4.1) in $\mathcal{P}$, and that system (4.1) is $\mathrm{UAS}(\mathcal{P})$. $\square$

Theorem 2. *Consider system* (4.1) *on $\mathcal{X}$, and assume that dynamics* (4.7) *are defined in $\mathcal{X}_E$. If a Lyapunov-like function for system* (4.8) *is found by solving the LP* (4.34)*, and $\mathcal{F} \subseteq \mathcal{P}$ (i.e. $V_E^- \geq V_{\mathcal{F}}^+$), the sets $\mathcal{P} \subseteq \mathcal{X}$ and $\mathcal{F} \subseteq \mathcal{P}$ are RPI sets with respect to dynamics* (4.1)*. Moreover, system* (4.1) *is* $\mathrm{UUB}(\mathcal{P}, \mathcal{F})$*.*
$\square$

Proof. The proof consists in showing that the PWA Lyapunov-like function

$$V_{\mathrm{UUB}}(x) \triangleq V(x),\ \forall x \in \mathcal{P} \tag{4.38}$$

where $V(x)$ is found as in Lemma 3 for (4.8) in $\mathcal{X}_H$, is a Lyapunov-like function for (4.1) over the set $\mathcal{P}$. Being $\mathcal{P}$ an RPI set for (4.8) in $\mathcal{X}_H$, the state update $f(x, w, d) \in \mathcal{P}$ for $x \in \mathcal{P}$ is always calculated using dynamics (4.1), and then $\mathcal{P}$ is an RPI set also with respect to the dynamics (4.1), because dynamics (4.7) is never executed. Considering that $\mathcal{P} \subseteq \mathcal{X}_H$, if (4.22) hold for all $x \in \mathcal{X}_H$ (and then for all $x \in \mathcal{P}$), the invariance of $\mathcal{F}$ is guaranteed by Lemma 3. Also, since we already proved that $\mathcal{P}$ is invariant, it follows that system (4.1) is $\mathrm{UUB}(\mathcal{P}, \mathcal{F})$. $\square$

The overall procedure proposed in this thesis for uniform asymptotic stability is summarized in Algorithm 1, while Algorithm 2 summarizes the procedure for uniform ultimate boundedness.

As a practical procedure to represent the set $\mathcal{P}$, one can define the polyhedra

$$\mathcal{X}_i^{\mathcal{P}} \triangleq \{x \in \mathcal{X}_i :\ V(x) < V_E^-\},\ i = 1, ..., \mathcal{I}, \tag{4.39}$$

Algorithm 1: Uniform asymptotic stability analysis

Input: PWA system (4.1)
Output: Region of attraction $\mathcal{P}$ and certificate of $\mathrm{UAS}(\mathcal{P})$ for system (4.1)
repeat
  if *a given maximum number of iterations is exceeded* **then**
    $\mathcal{P}$ undefined;
    STOP
  Obtain new regions $\mathcal{X}_i$ by splitting the existing regions;
  Compute $\tilde{\mathcal{R}}_{\cup}(\mathcal{X})$ in (4.6) and $\mathcal{X}_H$;
  Define the fake dynamics (4.7);
  Find $\mathcal{S}$ in (4.9) or (4.15), and the $\mathcal{X}_{i,j}$ in (4.10), (4.13) or (4.14);
  Solve the LP feasibility problem (4.19);
until *the LP has a solution*;
Find the region of attraction $\mathcal{P} \subseteq \mathcal{X}$;
System (4.1) is $\mathrm{UAS}(\mathcal{P})$

and define the invariant set $\mathcal{P}$ as

$$\mathcal{P} = \bigcup_{i=1}^{\mathcal{I}} \mathcal{X}_i^{\mathcal{P}} \tag{4.40}$$

4.5 Input-to-State Stability

In addition to the properties analyzed in the previous sections, the ISS properties of PWA systems of the form (4.11) can be analyzed. System (4.4) can be extended to the set $\mathcal{X}_H$, obtaining

$$x(k+1) = \begin{cases} A_{i,0}x(k) + a_{i,0} & \text{if } x(k) \in \mathcal{X}_i,\ i \in \mathcal{I} \\ \rho x(k) & \text{if } x(k) \in \mathcal{X}_E \end{cases} \tag{4.41}$$

which is a particular case of (4.8). According to [26], it is possible to determine if a system in form (4.11) is ISS relying on the asymptotic stability of the

Algorithm 2: Uniform ultimate boundedness analysis

Input: PWA system (4.1)

Output: Region of attraction $\mathcal{P}$, set $\mathcal{F}$, and certificate of $\mathrm{UUB}(\mathcal{P}, \mathcal{F})$ for system (4.1)

repeat
- **if** *a given maximum number of iterations is exceeded* **then**
  - $\mathcal{P}$ and $\mathcal{F}$ undefined;
  - STOP
- **repeat**
  - **if** *a given maximum number of iterations is exceeded* **then**
    - $\mathcal{P}$ and $\mathcal{F}$ undefined;
    - STOP
  - Obtain new regions $\mathcal{X}_i$ splitting the existing regions;
  - Compute $\tilde{\mathcal{R}}_{\cup}(\mathcal{X})$ in (4.6) and $\mathcal{X}_H$;
  - Define the fake dynamics (4.7);
  - Find $\mathcal{S}$ in (4.9) or (4.15), and the $\mathcal{X}_{i,j}$ in (4.10), (4.13) or (4.14);
  - Solve the LP (4.34);
- **until** *the LP has a solution*;
- find $\mathcal{F}$;

until $\mathcal{F} \subseteq \mathcal{P}$;

System (4.1) is $\mathrm{UUB}(\mathcal{P}, \mathcal{F})$

nominal system (4.4). To this purpose, we define the set of all possible additive disturbance realization for (4.11), as

$$\mathcal{D}' \triangleq \{d' \in \mathbb{R}^n \,:\, d' = E_i d,\ d \in \mathcal{D},\ i \in \mathcal{I}\} \tag{4.42}$$

The following result holds:

Theorem 3. *Let $\mathcal{D}' \subseteq \mathfrak{B}^\infty_\chi$, for a given $\chi > 0$. Let the set $\mathcal{P}^* \subseteq \mathcal{X}$ with $0 \in int(\mathcal{P}^*)$ be defined such that, for system* (4.11)*, $\mathcal{R}(\mathcal{P}^*) \oplus \mathfrak{B}^\infty_\chi \subseteq \mathcal{P}^*$. Suppose that the nominal extended system* (4.41) *is proved to be* $\mathrm{UAS}(\mathcal{X}_H)$ *solving the LP* (4.19)*, and* (4.4) *is* $\mathrm{UAS}(\mathcal{P}^*)$*. Then, system* (4.11) *is* $\mathrm{ISS}(\mathcal{P}^*)$.□

Proof. The result immediately follows by taking into account that the invoked assumptions lead to the fulfillment of the assumptions of Proposition IV.2 in [26]. □

The main problem to check ISS is to define the set $\mathcal{P}^*$. A simple choice consist of taking $\mathcal{P}^* = \mathcal{P}$, with $\mathcal{P}$ defined in (4.36) for the nominal system (4.4). For this simple choice, Algorithm 3 gives a procedure to check the ISS property.

Algorithm 3: Input-to-state stability analysis

Input: PWA systems (4.11) and (4.4)
Output: Region of attraction $\mathcal{P}$ and certificate of $\mathrm{ISS}(\mathcal{P})$ for system (4.11)
Run Algorithm 1 with system (4.4) as input and get $\mathcal{P}$;
Compute $\mathfrak{B}^\infty_\chi$ as the smallest $\mathfrak{B}^\infty_\chi$ such that $\mathfrak{B}^\infty_\chi \supseteq \mathcal{D}'$;
if $\mathcal{R}(\mathcal{P}) \oplus \mathfrak{B}^\infty_\chi \subseteq \mathcal{P}$ *for the dynamics* (4.11) **then**
  system (4.11) is $\mathrm{ISS}(\mathcal{P})$

4.6 Simulation examples

An interesting application of PWA stability analysis is to prove closed-loop stability properties of control systems composed by a linear (or PWA) system and a linear (or PWA) controller. In particular, the proposed stability and invariance analysis procedure is tested on the closed-loop system composed by a discrete-time PWA system and a switched explicit linear MPC controller (see, e.g., [33]).

The second-order open-loop PWA system is defined by

$$x(k+1) = A_j(w(k))x(k) + B_j u(k) + E_j d(k) \ \text{ if } x(k) \in \Omega_j, \tag{4.43}$$

where $j = 1, 2$, $x(k) \in \Omega_i \subset \mathbb{R}^2$, $w(k) \in \mathcal{W} \subset \mathbb{R}^2$, $d(k) \in \mathcal{D} \subset \mathbb{R}^2$ with

$$\begin{aligned} A_j(w) &\triangleq A_{j,0} + A_{j,1}w_{j,1} + A_{j,2}w_{j,2} \\ \mathcal{W} &\triangleq \left\{ w \in \mathbb{R}^2 : w_1 + w_2 = 1,\ w_1, w_2 \geq 0 \right\} \\ \mathcal{D} &\triangleq \left\{ d \in \mathbb{R}^2 : d_1 + d_2 = 0,\ \|d\|_\infty \leq 0.1 \right\} \end{aligned}$$

More specifically, we have

$$\begin{aligned} A_{1,0} = K_A \cdot &\begin{bmatrix} 0.7 & 0.7 \\ 0 & 0.7 \end{bmatrix}, A_{2,0} = \begin{bmatrix} 0.6 & 0.6 \\ 0 & 0.6 \end{bmatrix}, \ K_A \in \mathbb{R}_+ \\ A_{1,1} = &\begin{bmatrix} 0 & 0.1 \\ 0 & 0 \end{bmatrix}, A_{2,1} = \begin{bmatrix} 0 & -0.2 \\ 0 & 0 \end{bmatrix} \\ A_{1,2} = &\begin{bmatrix} 0 & 0 \\ 0 & -0.1 \end{bmatrix}, A_{2,2} = \begin{bmatrix} 0 & 0 \\ 0 & 0.2 \end{bmatrix} \\ B_1 = &\begin{bmatrix} 1.1 \\ 0 \end{bmatrix}, B_2 \begin{bmatrix} 0.9 \\ 0 \end{bmatrix} \\ E_1 = E_2 = K_E \cdot &\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ K_E \in \mathbb{R}_+ \end{aligned}$$

where K_A and K_E are scalar parameters to be assigned.

The regions Ω_i are defined by $\Omega_1 = \{x \in \mathbb{R}^2 : H_1 x \leq h_1\}$, $\Omega_2 = \{x \in \mathbb{R}^2 : H_2 x \leq h_2\} \setminus \Omega_1$, with

$$H_1 = \begin{bmatrix} 0 & 0.1 \\ 0 & -0.1 \\ -0.1 & 0 \\ 1 & 0 \end{bmatrix}, H_2 = \begin{bmatrix} 0 & 0.1 \\ 0 & -0.1 \\ -1 & 0 \\ 0.1 & 0 \end{bmatrix}$$

$$h_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}, h_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

and the whole set where the system dynamics are defined is $\mathcal{X} = \Omega_1 \cup \Omega_2$. The switched explicit linear MPC controller is defined by computing an explicit MPC control law [9] $u_j(x)$ for each nominal linear system

$$x(k+1) = A_{j,0}x(k) + B_j u(k)$$

and by setting

$$u(k) = u_j(k) \ \text{ if } x(k) \in \Omega_j \tag{4.44}$$

with a prediction horizon $N = 5$, a control horizon $N_u = 2$, weight matrices

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \quad R = 0.1$$

on the state and the control, respectively, and terminal weight matrices $P_j \in \mathbb{R}^2$ found as the solution of the Lyapunov equation $A'_{j,0} P A_{j,0} - P = -Q$ on each partition Ω_j. The control constraints are $u \in [-4, 4]$, and the state constraints

$x \in \mathcal{X}$, are also imposed as *soft* constraints in solving the multiparametric programs to obtain the control law (4.44).[1]

The overall closed-loop system, which does not have any a priori stability properties, can be written in form (4.1). In this case, the partitions $\mathcal{X}_i$ are those automatically generated by the multiparametric programming solver that determines the PWA functions $u_j : \Omega_j \to \mathbb{R}$, $j = 1, 2$. Three different stability analysis problems are considered below.

4.6.1 Uniform asymptotic stability

Assume that the additive disturbance is not present in (4.43) ($K_E = 0$). Assume also $K_A = 1$. The set $\mathcal{X}$ is not invariant for the considered overall PWA closed-loop system. For instance, starting at an initial condition $x(0) = \begin{bmatrix}-10 & -9.5\end{bmatrix}' \in \mathcal{X}$, the explicit MPC control variable is $u(0) = 4$. With a disturbance vector $w(0) = \begin{bmatrix}0.9 & 0.1\end{bmatrix}'$, it yields $x(1) = \begin{bmatrix}-10.11 & -6.56\end{bmatrix} \notin \mathcal{X}$. In this case, we can verify if the closed-loop system is UAS in a set $\mathcal{P} \subseteq \mathcal{X}$. Therefore, we find $\tilde{\mathcal{R}}_{\cup}(\mathcal{X})$ in (4.6) and its bounding box $\mathcal{X}_H$, with the extended dynamics (4.8) defined with $\rho = 0.99$. The regions obtained using the switched explicit MPC, together with the extension given by $\mathcal{X}_H \setminus \mathcal{X}$, are shown in Fig. 4.1.

According to Algorithm 1, we found the transition map $\mathcal{S}$ in (4.9), the transition sets $\mathcal{X}_{i,j}$ in (4.10), and solved the LP (4.19) with $\lambda = 0.99$, proving that the closed-loop system is $\mathrm{UAS}(\mathcal{P})$. The LP is composed of 691 constraints and 76 variables, and solved using CPLEX in 7.3 ms on a 2.4 GHz processor. The

[1]Note that the linear MPC control laws are defined for all $x \in \mathbb{R}^2$. The multiparametric program is solved only for $x(k) \in \mathcal{X}$, and predictions $x(k+t)$, $t = 1, ..., N$, may exit $\mathcal{X}$. In order to allow this, the constraints $x \in \mathcal{X}$ are softened.

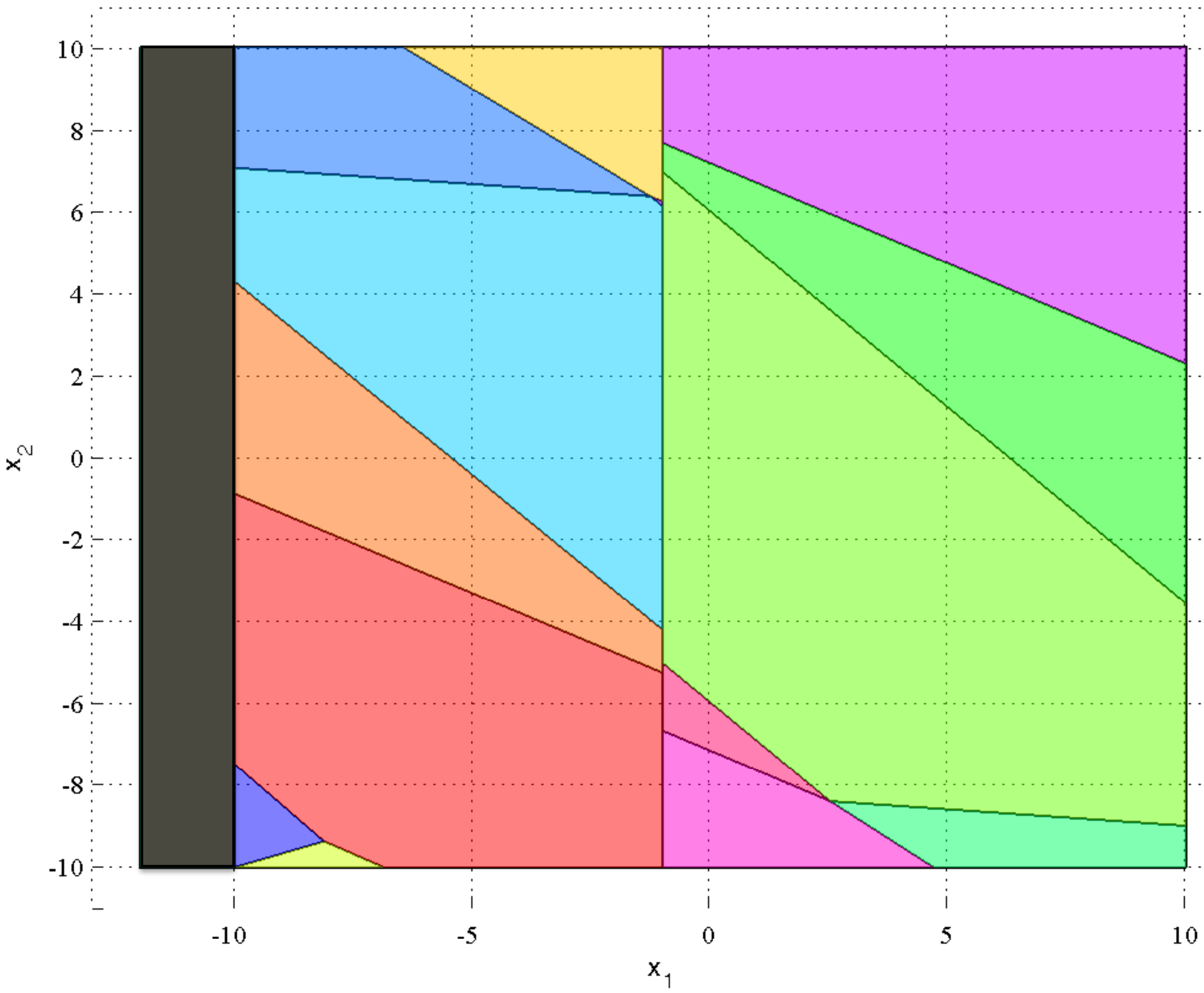

Figure 4.1: The invariant set $\mathcal{X}_H$ for case A is constituted by the union of the regions of the explicit MPC and the box $\mathcal{X}_H \setminus \mathcal{X}$ (the grey rectangle on the left)

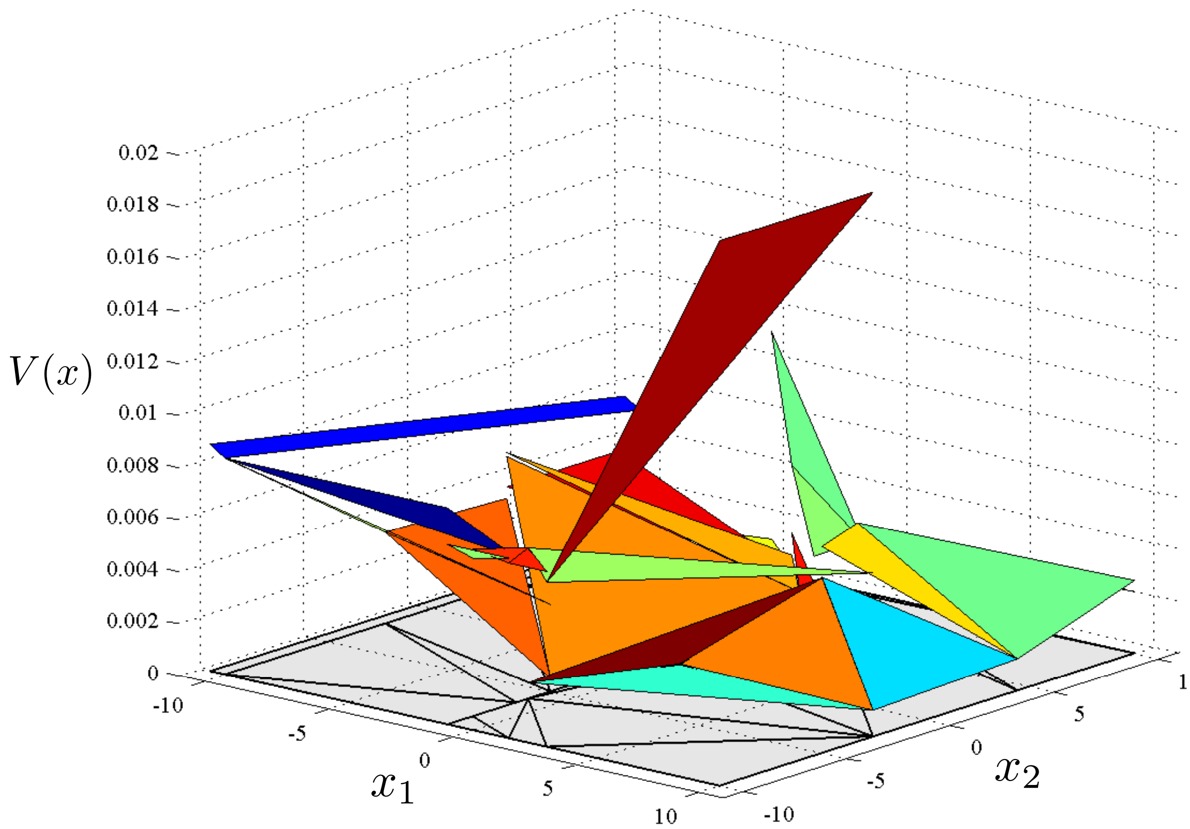

Figure 4.2: The PWA Lyapunov function for the extended system in case A

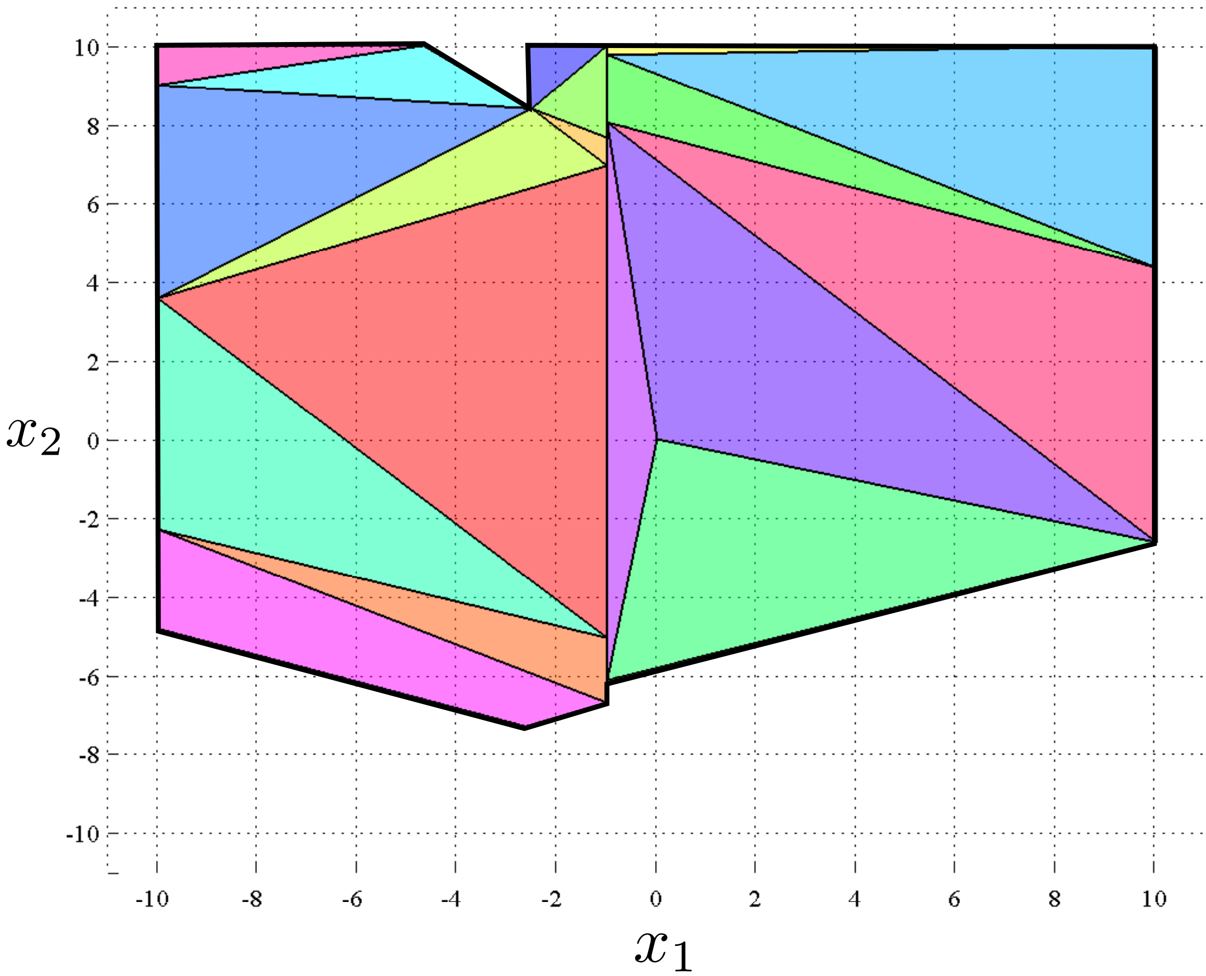

Figure 4.3: The invariant set $\mathcal{P}$ obtained in case A

corresponding Lyapunov function is shown in Figure 4.2 and the RPI set $\mathcal{P}$ is shown in Figure 4.3.

4.6.2 Uniform ultimate boundedness

Assume now that both parametric and additive disturbances are present, with $K_A = 1$ and $K_E = 10^{-2}$. Even though the additive disturbance is a relatively small one, we already know that the system is not uniformly asymptotically stable in any subset of $\mathcal{X}$. However, we can verify if the closed-loop system is UUB from a set $\mathcal{P} \subseteq \mathcal{X}$ to a set $\mathcal{F} \subseteq \mathcal{P}$. Therefore, we find $\tilde{\mathcal{R}}_{\cup}(\mathcal{X})$ in (4.6) and its bounding box $\mathcal{X}_H$, with the extended dynamics (4.8) defined with $\rho = 0.99$, as in case A. According to Algorithm 2, the transition map $\mathcal{S}$ in (4.9) and the transition sets $\mathcal{X}_{i,j}$ in (4.10) are found. Then, the LP (4.34) is solved with $\lambda = 0.99$, proving that the closed-loop system is $\mathrm{UUB}(\mathcal{P}, \mathcal{F})$. The LP is composed of 1308 constraints and 78 variables, and solved using CPLEX in 8.6 ms on a 2.4 GHz processor. The corresponding Lyapunov-like function is shown in Figure 4.4, while Figure 4.5 shows the invariant set $\mathcal{P}$ and the terminal set $\mathcal{F}$.

4.6.3 Input-to-state stability

Finally, assume that the parametric disturbance w is not present in (4.43), i.e. $A_{i,j} = 0,\ i,j = 1,2$. Moreover, $K_A = 0.9$ and $K_E = 20$. In this case, it is possible to check the ISS of the system with respect to the additive disturbance. The set $\mathcal{D}'$ in (4.42) is easily found as $\mathcal{D}' \triangleq \left\{d' \in \mathbb{R}^2 : d_1 + d_2 = 0,\ \|d\|_\infty \leq 2\right\}$. According to Algorithm 3, we found the transition map $\mathcal{S}$ in (4.15) and the transition sets $\mathcal{X}_{i,j}$ in (4.14) for the nominal closed-loop system. The feasibility problem (4.19) with $\lambda = 0.99$ is composed of 435 constraints and 73 variables,

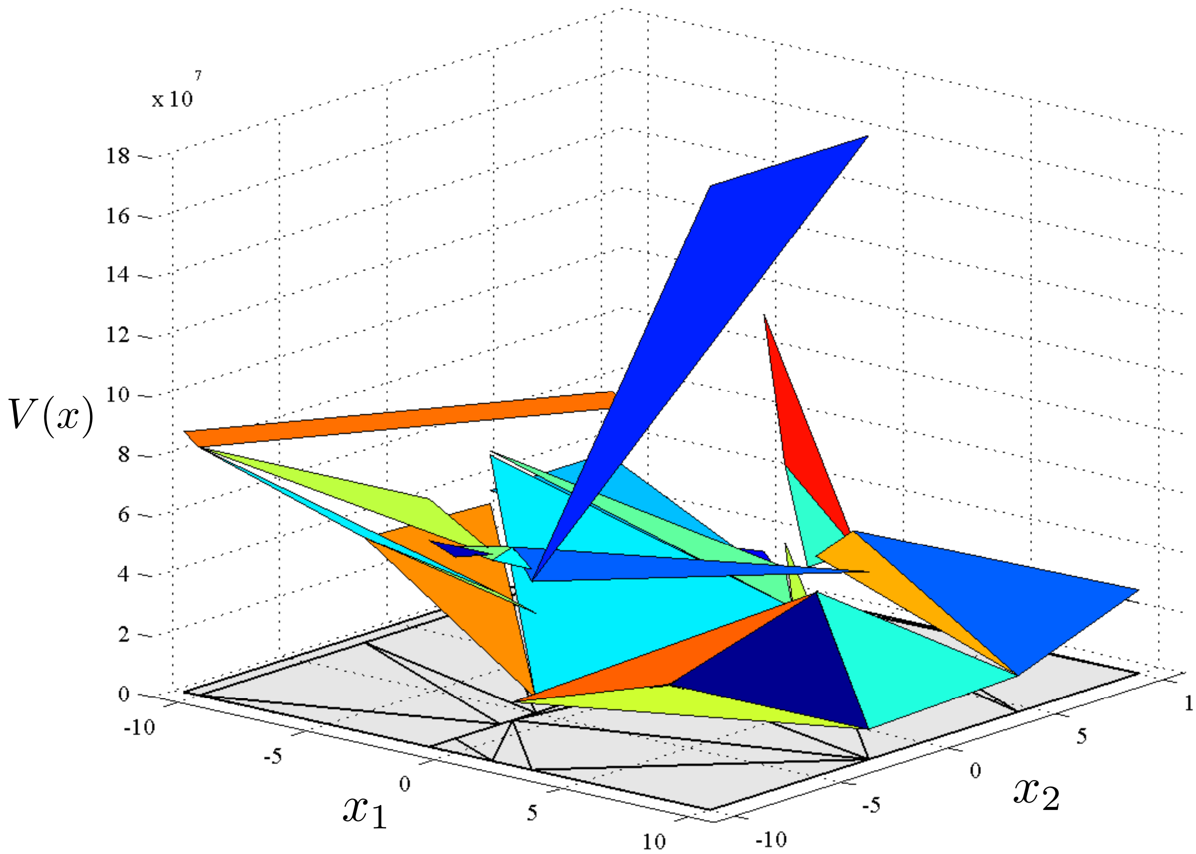

Figure 4.4: The PWA Lyapunov-like function for the extended system in case B

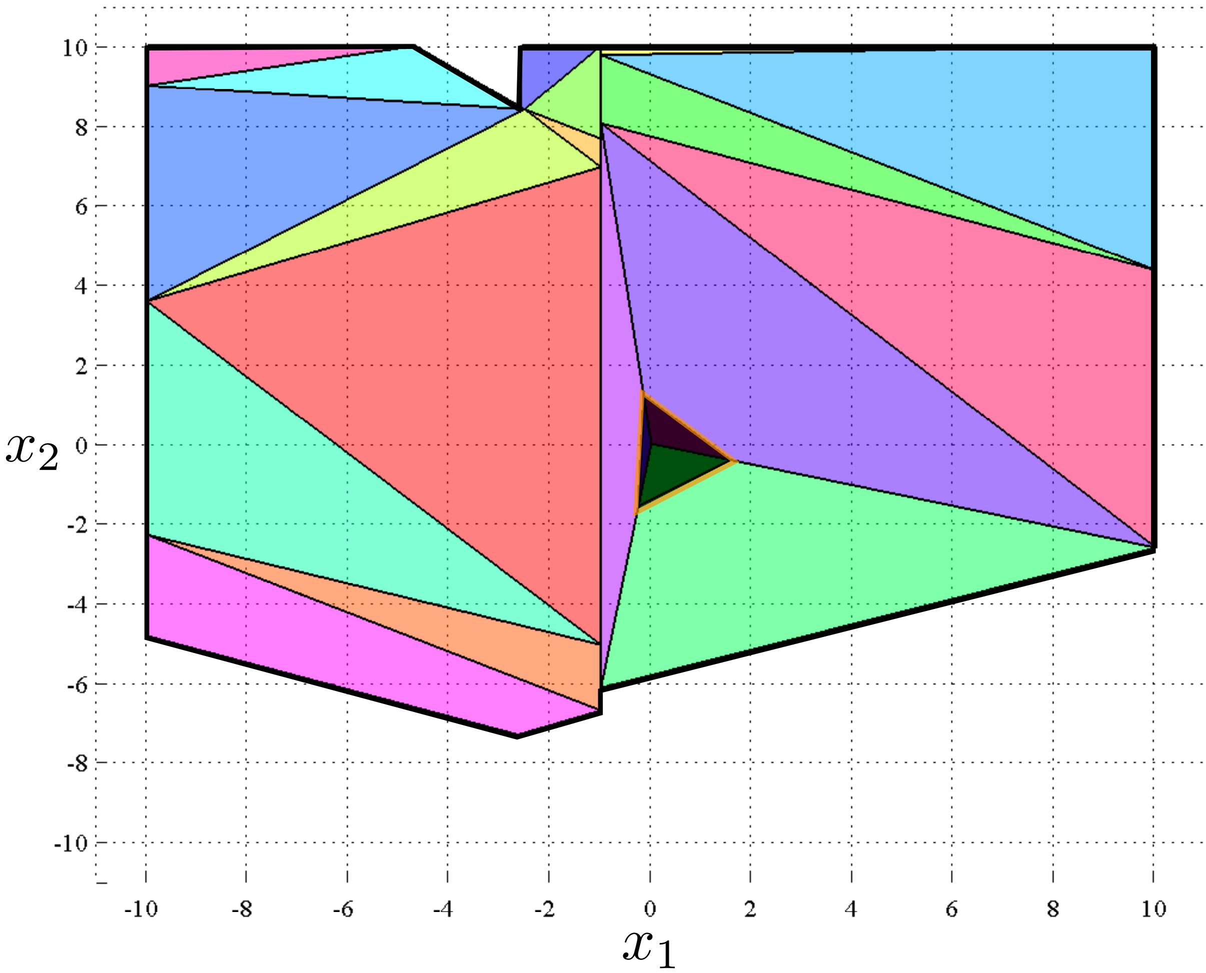

Figure 4.5: The invariant set $\mathcal{P}$ and the terminal set $\mathcal{F}$ (shaded) in case B

and solved using CPLEX in 3.0 ms on a 2.4 GHz processor. Since the problem admits a solution, the system is proved to be $\mathrm{UAS}(\mathcal{X})$, i.e. $\mathcal{X} \equiv \mathcal{P}$ is already an RPI set. The set $\mathfrak{B}_{\mathcal{X}}^{\infty}$ is equal to $\mathfrak{B}_2^{\infty}$, and therefore we can compute $\mathcal{R}(\mathcal{P}) \oplus \mathfrak{B}_2^{\infty}$. The result is shown in Fig. 4.6, where it is possible to see that $\mathcal{R}(\mathcal{P}) \oplus \mathfrak{B}_2^{\infty} \subset \mathcal{P}$. We can conclude that the closed-loop system is $\mathrm{ISS}(\mathcal{X})$.

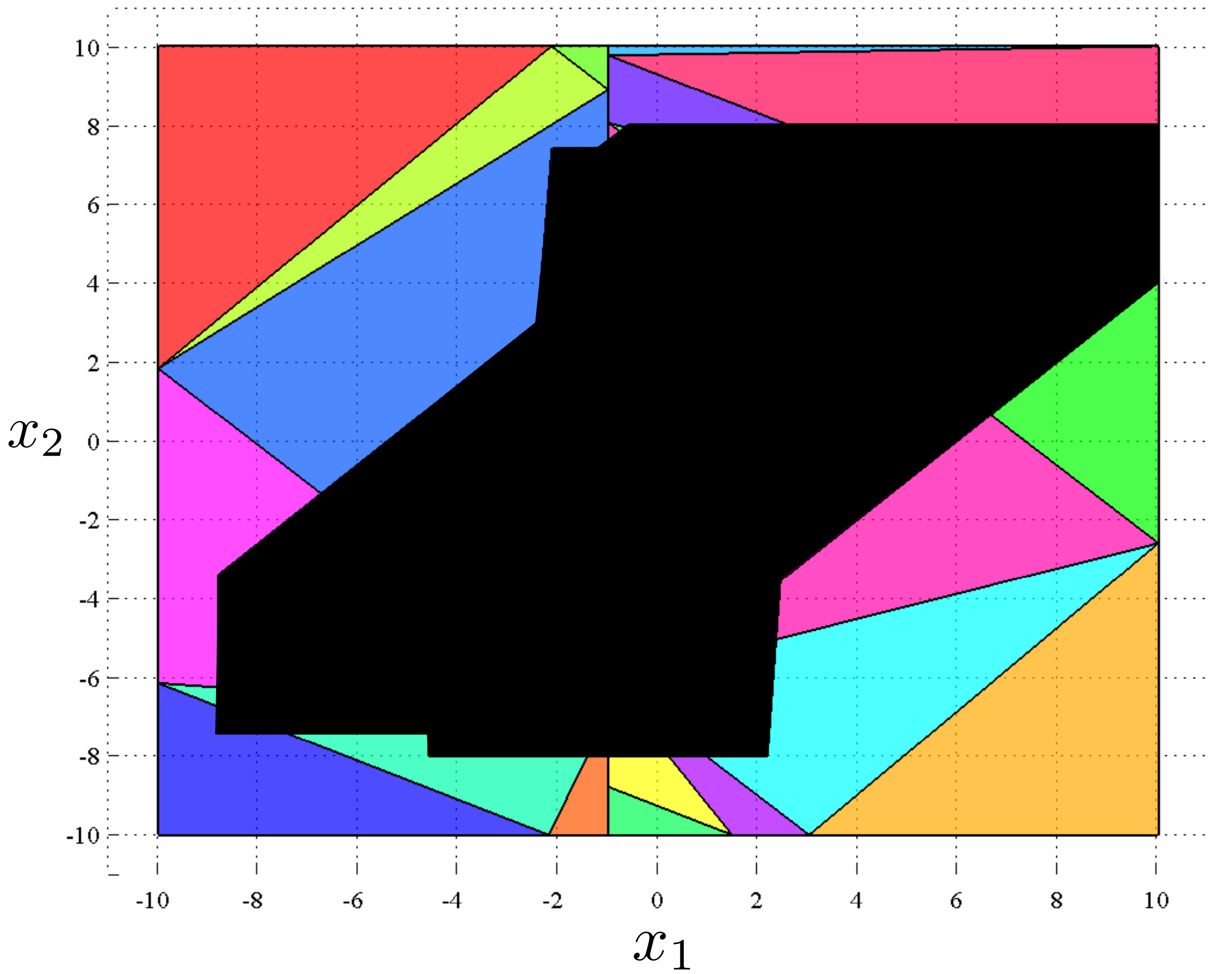

Figure 4.6: The sets $\mathcal{P} \equiv \mathcal{X}$ and $\mathcal{R}(\mathcal{P}) \oplus \mathfrak{B}_2^\infty$ (in black) in case C

Chapter 5

Conclusion

For hybrid MPC controllers approximated as switched linear MPC controllers we have proposed an architecture to evaluate discontinuous PWAS functions by extending the architecture proposed in [40]. The implementation requires the introduction of comparators and 1-bit adders which are very simple and fast devices and so is remarkably efficient. This fact is a direct consequence of the chosen class of discontinuous functions. Thus, our circuit represents a good trade-off between model complexity and circuit performances in terms of area occupation and power consumption. A further generalization of the proposed architecture can be obtained by using a binary-tree search [23,31] instead of the bank of comparators. In this case, we could implement discontinuous functions that are continuous over non hyper-rectangular regions, at the cost of a more complex circuit.

Also, this work has addressed the problem of determining the properties of uniform asymptotic stability, uniform ultimate boundedness, and input-to-state stability of (possibly discontinuous) uncertain discrete-time PWA systems, subject to both additive and parametric disturbances. Since the dynamics are defined in the set $\mathcal{X}$ that is a-priori not invariant, a partially fictitious dynamics is exploited to define a (possibly discontinuous) PWA Lyapunov functions, leading to the formulation of an LP. As an outcome of the optimization problem, the RPI set $\mathcal{P} \subseteq \mathcal{X}$ is obtained, together with the terminal set $\mathcal{F}$, in case uniform ultimate boundedness is analyzed. The reported simulation examples, although referring to a second-order system for the sake of simplicity, confirm the low computational burden required to solve the LPs associated with the proposed method, therefore making the proposed method amenable for analysis of more complex systems arising in real-life control applications.

Open problems and suggestions

In this thesis, the stability analysis problem was formulated and solved using PWA Lyapunov function, as well as the synthesis of PWA controllers through FPGA. Since the flexibility of such functions, it would be subject of further interest to analyze the synthesis problem. Many are the applications that would receive benefit in terms of simplicity of controller synthesis step, after the plant design. As an example, when dealing with vehicular systems, a large mixture of state machines and dynamical models are involved in the controller design steps. The design can be simplified using the hybrid MPC approach, and, where the resulting controller is too complex, using the switched MPC approach. Also, since the large-scale models, a large-scale stability problem arises in such cases. It would be an important step forward to develop an algorithm able to synthesize a switched MPC such that the closed-loop is stable. In this way, one can inherit the controller's performance together with the simplicity of tuning of the single MPC, digging and searching for the optimal tuning in a stable class of controllers.

Bibliography

[1] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In L. Magni, F. Allgower, and D. M. Raimondo, editors, *Nonlinear Model Predictive Control: Towards New Challenging Applications*, volume 384, pages 345–369. Berlin Heidelberg, 2009.

[2] A. Bemporad. Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Transactions on Automatic Control*, 49(5):832–838, 2004.

[3] A. Bemporad. Hybrid Toolbox - User's Guide, 2004. http://www.ing.unitn.it/ bemporad/hybrid/toolbox.

[4] A Bemporad. Model predictive control design: New trends and tools. *45th IEEE Conference on Decision and Control*, pages 6678–6683, 2006.

[5] A Bemporad, F Borrelli, and M Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, pages 1–12, Jul 2001.

[6] A. Bemporad, W. Heemels, and B. De Schutter. On hybrid systems and closed-loop MPC systems. *IEEE Transactions on Automatic Control*, 47(5):863–869, 2002.

[7] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–428, 1999.

[8] A Bemporad and M Morari. Robust model predictive control: A survey. *Lecture notes in control and information sciences*, pages 207–226, 1999.

[9] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[10] A. Bemporad, A. Oliveri, T. Poggi, and M. Storace. Ultra-fast stabilizing model predictive control via canonical piecewise affine approximations. *IEEE Trans. Autom. Control.*, 2011. In press.

[11] P. Biswas, P. Grieder, J. Löfberg, and M. Morari. A survey on stability analysis of discrete-time piecewise affine. In *Proceedings of 16th IFAC World Congress*, Praha, Czech Republic, June 2005.

[12] F. Blanchini and S. Miani. *Set-theoretic methods in control.* Springer, 2008.

[13] L. G. Bleris, P. D. Vouzis, M. G. Arnold, and M. V. Kothare. A coprocessor FPGA platform for the implementation of real-time model predictive control. In *Amer. Control Conf.*, Minneapolis, MN, June 2006.

[14] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41(10):1709–1721, 2005.

[15] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory.* SIAM, 1994.

[16] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[17] F. J. Christophersen, M. Baotic, and M. Morari. Stability analysis of hybrid systems with a linear performance index. In *IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 2004.

[18] S. Di Cairano, H. E. Tseng, D. Bernardini, and A. Bemporad. Steering vehicle control by switched model predictive control. In *6th IFAC Symp. Adv. Automotive Control*, Munich, Germany, July 2009.

[19] P. Echevarria, M. V. Martínez, J. Echanobe, I. del Campo, and J. M. Tarela. Digital hardware implementation of high dimensional fuzzy systems. In *Applications of Fuzzy Sets Theory*, Lecture Notes in Computer Science, pages 245–252. Springer, Berlin, 2007.

[20] G. Ferrari-Trecate, F. A. Cuzzola, D. Mignone, and M. Morari. Analysis of discrete-time piecewise affine and hybrid systems. *Automatica*, 38(12):2139–2146, December 2002.

[21] HJ Ferreau, HG Bock, and M Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[22] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari. Stabilizing low complexity feedback control of constrained piecewise affine systems. *Automatica*, 41(10):1683–1694, 2005.

[23] T. A. Johansen, W. Jackson, R. Schreiber, and P. Tøndel. Hardware synthesis of explicit model predictive controllers. *IEEE Transaction on Control System Technology*, 15(1):191–197, Jan. 2007.

[24] E. C. Kerrigan and D. Q. Mayne. Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Conference on Decision and Control*, Las Vegas, NV, December 2002.

[25] M. Lazar. *Model predictive control of hybrid systems: stability and robustness*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.

[26] M. Lazar, W. P. M. H. Heemels, and A. R. Teel. Lyapunov functions, stability and input-to-state stability subtleties for discrete-time discontinuous systems. *IEEE Transactions on Automatic Control*, 54(10):2421–2425, 2009.

[27] H. Lin and P.J. Antsaklis. Synthesis of uniformly ultimate boundedness switching laws for discrete-time uncertain switched linear systems. In *Proc. Conference on Decision and Control*, Maui, HI, December 2003.

[28] K. Ling, B. Wu, and J. Maciejowski. Embedded model predictive control (MPC) using a FPGA. In *Proc. 17th IFAC World Congress*, pages 15250–15255, Seoul, Korea, July 2008.

[29] K. V. Ling, S. P. Yue, and J. Maciejowski. A FPGA implementation of model predictive control. In *Amer. Control Conf.*, Minneapolis, MN, June 2006.

[30] D Mayne and J Rawlings. Constrained model predictive control: stability and optimality (vol 36, pg 789, 2000). *Automatica*, 36:789–814, Jan 2001.

[31] A. Oliveri, A. Oliveri, T. Poggi, and M. Storace. Circuit implementation of piecewise-affine functions based on a binary search tree. In *Proc. Europ.*

Conf. Circ. Th. Design (ECCTD'09), pages 145–148, Antalya, Turkey, August 23-27 2009.

[32] M. Parodi, M. Storace, and P. Julián. Synthesis of multiport resistors with piecewise-linear characteristics: a mixed-signal architecture. *Int. J. Circ. Th. Appl.*, 33(4):307–319, 2005.

[33] T. Poggi, S. Trimboli, A. Bemporad, and M. Storace. Explicit hybrid model predictive control: discontinuous piecewise-linear approximation and FPGA implementation. In *IFAC World Congress*, Milan, Italy, August 2011.

[34] S. Prajna and A. Papachristodoulou. Analysis of switched and hybrid systems-beyond piecewise quadratic methods. In *Proc. American Control Conference*, Denver, CO, June 2003.

[35] S. V. Raković, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[36] A. Richards, W. Stewart, and A. Wilkinson. Auto-coding implementation of model predictive control with application to flight control. In *Proc. Europ. Control Conf. 2009*, Budapest, Hungary, August 2009.

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

[38] M. Rubagotti, S. Trimboli, D. Bernardini, and A. Bemporad. Stability and invariance analysis of approximate explicit MPC based on PWA Lyapunov functions. In *18th IFAC World Congress*, Milan, Italy, August 2011.

[39] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.

[40] M. Storace and T. Poggi. Digital architectures realizing piecewise-linear multi-variate functions: two FPGA implementations. *Int. J. Circ. Th. Appl.*, 37, 2010. in press, doi: 10.1002/cta.610.

[41] P. Tøndel, T. A. Johansen, and A. Bemporad. Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950, 2003.

[42] S. Trimboli, M. Rubagotti, and A. Bemporad. Stability and invariance analysis of uncertain PWA systems based on linear programming. In *Proc. Conference on Decision and Control*, 2011.

[43] M Vidyasagar. *Nonlinear systems analysis*. Prentice-Hall, 1993.

[44] L. Yepremyan and J. E. Falk. Delaunay partitions in $\mathbb{R}^n$ applied to non-convex programs and vertex/facet enumeration problems. *Computers and Operations Research*, 32(4):793–812, 2005.

[45] G. Zhai, H. Lin, and P. J. Antsaklis. Quadratic stabilizability of switched linear systems with polytopic uncertainties. *International Journal of Control*, 76(7):747–753, 2003.