Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Computation of Observability Regions for Piecewise Affine Systems: A Projection-Based Algorithm

Mehdi Gati and Giancarlo Ferrari-Trecate

Abstract— **In this paper we consider the problem of computing sets of observable states for discrete-time, piecewise affine systems. When the maximal set of observable states is full-dimensional, we provide an algorithm for reconstructing it up to a zero measure set. The core of the method is a quantifier elimination procedure that, in view of basic results on piecewise linear algebra, can be performed via the projection of polytopes on subspaces. We also provide a necessary condition on the minimal length of the observability horizon in order to expect a full-dimensional set of observable states. Numerical experiments highlight that the new procedure is considerably faster than the one proposed in [1].**

I. INTRODUCTION

In the last ten years there has been a significant progress in the study of Piece-Wise Affine (PWA) Systems and various algorithms for checking their structural properties have been proposed. In particular, some research focused on exploiting the PWA structure in order to derive verifiable conditions for observability. For continuous-time PWA systems, observability tests have been proposed in [2]. Much richer is the literature on observability for discrete-time PWA systems. To our knowledge, the first contribution to this topic has been given by Sontag in the eighties [3]. By exploiting tools of Piecewise Linear (PL) algebra [4] he discussed the existence of observers for PWA models. He also showed that observability over an infinite time-horizon is undecidable while observability over a finite time-horizon is decidable [3], although NP-complete [5]. These results promoted the interest in computational methods for checking the latter property [1]. Algorithms for testing the observability of PWA systems have been proposed in [6]. In particular, by exploiting the equivalence between PWA and Mixed Logic-Dynamical (MLD) models, methods based on Mixed-Integer Linear Programming (MILP) have been given for checking if a set of states is observable or not. Some algebraic conditions for observability of piecewise linear autonomous systems have been derived in [7].

Note that the papers [6] and [7] focus on algorithms for checking if all admissible states (or the states in a prescribed set [6]) of a PWA system are observable, thus providing a yes/no answer. A different problem is the computation of observability regions, i.e. subsets of state space composed by observable states. As recalled in Section II-C, the *maximal* observability region $\bar{\mathbb{O}}_T$ (on a horizon of T steps) can be defined through a sentence in the language of PL algebra and this implies that it is a finite union of possibly non closed polytopes. A first algorithm for computing $\bar{\mathbb{O}}_T$ has been proposed in [1] where, under some minor assumptions, it has been shown that $\bar{\mathbb{O}}_T$ can be reconstructed by solving a suitable multi-parametric MILP (mp-MILP). However, this procedure presents some drawbacks, in terms of computational time, inherent to the use of multi-parametric programming.

The present paper proposes a new algorithm for computing $\bar{\mathbb{O}}_T$, when it is full-dimensional, for discrete-time PWA systems. The method combines results from PL algebra with *ad hoc* observability sub-tests for reconstructing $\bar{\mathbb{O}}_T$ up to a zero measure set. At the implementation level our procedure relies on three basic algorithms: one for projecting polyhedra over a subspace (see [8] for a review of various methods), one for checking the rank of a matrix and one for computing the set difference between collections of polytopes [9]. Given the availability of methods for performing these operations, the overall algorithm can be easily coded.

The paper is structured in three main parts. The first one (Section II) presents some basic results on PWA systems, observability theory and PL algebra. In Section III, we detail our algorithm. Finally, in Section IV we illustrate the computational advantages of the procedure through numerical examples. In particular, we show that the new algorithm is considerably faster than the one proposed in [1].

II. THEORETICAL BACKGROUND

A. *Piecewise affine systems*

Consider the following discrete-time PWA system

$$\begin{array}{l} x(t+1) = \mathscr{A}_i x(t) + \mathscr{B}_i u(t) + a_i \\ y(t) = \mathscr{C}_i x(t) + \mathscr{D}_i u(t) + c_i \end{array} \quad \text{if } [x', u']' \in Q_i \qquad (1)$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, $y \in \mathbb{R}^p$ is the output, $\mathscr{A}_i$, $\mathscr{B}_i$, a_i, $\mathscr{C}_i$, $\mathscr{D}_i$ and c_i are matrices of suitable dimensions and $Q_i \subset \mathbb{R}^{n+m}$, $i \in I_r \subset \mathbb{N}^+$ are disjoint, full dimensional and not necessarily closed polytopes. Apparently, states and inputs of (1) are subject to the constraints

$$x \in \mathbb{X} = \bigcup_i \mathrm{Proj}_x(Q_i) \text{ and } u \in \mathbb{U} = \bigcup_i \mathrm{Proj}_u(Q_i) \qquad (2)$$

where $\mathrm{Proj}_x(Q_i)$ (resp. $\mathrm{Proj}_u(Q_i)$) denotes the projection of Q_i on the $x-$coordinates (resp. the $u-$coordinates). Note that the sets $\mathbb{X}$ and $\mathbb{U}$ are not necessarily connected.

Corresponding author: M. Gati

M. Gati is with the Renault Research Department, Technocentre, 1, avenue du Golf, 78288 Guyancourt Cedex, France. Email: mehdi.gati@renault.com. Tel: +33 (0)1 76 85 71 08, Fax: +33 (0)1 76 85 77 23.

G. Ferrari-Trecate is with the Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, B.P.105, 78153, Le Chesnay Cedex, France. Email: Giancarlo.Ferrari-Trecate@inria.fr. Tel: +33 (0)1 39 63 59 21, Fax: +33 (0)1 39 63 57 86.

This work has been partially done in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368

[1]For sake of conciseness, in the sequel "observability" will be used instead of "observability on a finite horizon".

To a discrete-time vector-valued signal $w(t)$ and a time horizon $T > 0$, we associate the capitalized vector $\mathbf{W} = \begin{bmatrix} w(0)' & \dots & w(T-1)' \end{bmatrix}'$ that collects the samples of w. In the sequel, we will occasionally use the notation $\mathbf{X}(x,\mathbf{U})$ and $\mathbf{Y}(x,\mathbf{U})$ for highlighting the dependence of $\mathbf{X}$ and $\mathbf{Y}$ on the initial state x and the inputs $\mathbf{U}$. The system evolution is *blocked* at time t if $x(t+1) \notin \mathbb{X}$. When the system evolution is blocked at some $t \leq T-1$, the definition of $\mathbf{X}(x,\mathbf{U})$ and $\mathbf{Y}(x,\mathbf{U})$ are meaningless. Then, we say that $\mathbf{X}(x,\mathbf{U})$ and $\mathbf{Y}(x,\mathbf{U})$ are *well-defined* if x and $\mathbf{U}$ are such that the evolution of system (1) is not blocked at any time $t \leq T-1$. This leads to the introduction of the T-step feasibility set $\mathbb{X}_T^*$

$$\mathbb{X}_T^* = \{x \in \mathbb{X} : \exists \mathbf{U} \in \mathbb{U}^T \text{ s.t. } \mathbf{X}(x,\mathbf{U}) \in \mathbb{X}^T\} \tag{3}$$

By using results from [4] (see also [7] for the case of autonomous piecewise linear systems), it is easy to prove that the function $\{[x',\ \mathbf{U}']' : \mathbf{Y} \text{ is well defined}\} \mapsto \mathbf{Y}$ is piecewise affine, i.e. it can be written as

$$\mathbf{Y} = C_i x + D_i \mathbf{U} + f_i \text{ if } \begin{bmatrix} x' & \mathbf{U}' \end{bmatrix}' \in P_i,\ i \in \mathscr{M} \subset \mathbb{N}^+ \tag{4}$$

where C_i, D_i and f_i are suitably defined matrices and the regions $P_i \subset \mathbb{X} \times \mathbb{U}^T$, are disjoint, full dimensional and not necessarily closed polytopes. Apparently, the set $P = \cup_{i \in \mathscr{M}} P_i$ coincides with the set $\{[x',\ \mathbf{U}']' : \mathbf{Y} \text{ is well defined}\}$.

Moreover, the T-step feasible set $\mathbb{X}_T^*$ defined in (3) can be written as as

$$\mathbb{X}_T^* = \bigcup_{i \in \mathscr{M}} \text{Proj}_x(P_i)$$

A possible way for computing the matrices and the regions in (4) is to resort to the MLD representation of system (1) and then use an algorithm based on mode enumeration.

As illustrated in [6], if the regions Q_i are closed, we can represent system (1) in the MLD form [10]

$$x(t+1) = Ax(t) + B_1 u(t) + B_2 \delta(t) + B_3 z(t) \tag{5a}$$

$$y(t) = Cx(t) + D_1 u(t) + D_2 \delta(t) + D_3 z(t) \tag{5b}$$

$$g(\delta(t), z(t), u(t), x(t)) \leq 0 \tag{5c}$$

$$g(\delta, z, u, x) = E_2 \delta(t) + E_3 z(t) - E_1 u(t) - E_4 x(t) - E_5 \tag{5d}$$

where $\delta \in \{0,1\}^{r_\ell}$, $z \in \mathbb{R}^{r_c}$ represent auxiliary binary and continuous variables, respectively, and A, C, B_i, D_i, $i = 1,\dots,3$, E_j, $j = 1,\dots,5$ are matrices of suitable dimensions. The map $g : \{0,1\}^{r_\ell} \times \mathbb{R}^{r_c} \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^q$ defines q linear constraints through (5c).

Using the linearity of (5), the state trajectory of an MLD system over a time window of length T can be written in compact form as [10]

$$\mathbf{X} = \tilde{A}x + \tilde{B}_1 \mathbf{U} + \tilde{B}_2 \Delta + \tilde{B}_3 \mathbf{Z} \tag{6a}$$

$$\tilde{E}_2 \Delta + \tilde{E}_3 \mathbf{Z} \leq \tilde{E}_1 \mathbf{U} + \tilde{E}_4 x + \tilde{E}_5 \tag{6b}$$

$$\mathbf{U} \in \mathbb{U}^T, \mathbf{X} \in \mathbb{X}^T \tag{6c}$$

where $\tilde{A}$, $\tilde{B}_1$, $\tilde{B}_2$, $\tilde{B}_3$ and $\tilde{E}_j$, $j = 1,\dots,5$ are suitable matrices and x is the initial state. Analogously, the output trajectory can be represented by (6) complemented with the equation

$$\mathbf{Y} = \tilde{C}x + \tilde{D}_1 \mathbf{U} + \tilde{D}_2 \Delta + \tilde{D}_3 \mathbf{Z} \tag{7}$$

where $\tilde{C}$, $\tilde{D}_1$, $\tilde{D}_2$, $\tilde{D}_3$ are suitably defined.

B. Mode enumeration

In order to represent the vector $\mathbf{Y}$ appearing in (7) as a piecewise affine function of x and $\mathbf{U}$, we adopt a mode enumeration procedure inspired by the algorithms for solving mp-MILPs (see Section 1.5 in [11]). The key idea is to enumerate all the vectors Δ of binary variables that verify (6). To this purpose, let us define the following mixed-integer feasibility problem

$$\mathscr{P}: \text{ find } \begin{bmatrix} \mathbf{U}' & \mathbf{Z}' & \Delta' & x' \end{bmatrix}' \text{ subject to (6b) and (6c)}$$

Mode Enumeration Algorithm

i) Solve problem $\mathscr{P}$
ii) **If** $\mathscr{P}$ is feasible, let Δ_i be a feasible combination of binary variables. **Else** Stop the procedure;
iii) Deduce the expression of $\mathbf{Z}$ from (6b). Actually, for $\Delta = \Delta_i$, it is possible to extract from the constraints (6b) a set of equalities expressing $\mathbf{Z}$ as an affine function z_i of x and $\mathbf{U}$ (see [10] for further details);
iv) Define the polytopic region $P_i \subset \mathbb{X} \times \mathbb{U}^T$ by replacing $\mathbf{Z}$ with $z_i(x,\mathbf{U})$ in the non active inequalities in (6b) and (6c);
v) By using $\mathbf{Z} = z_i(x,\mathbf{U})$ and $\Delta = \Delta_i$ in (7) the output can be expressed as $\mathbf{Y} = C_i x + D_i \mathbf{U} + f_i$ for $[x',\ \mathbf{U}']' \in P_i$;
vi) Add the constraints $M_i \Delta \leq N_i$ in the definition of problem $\mathscr{P}$, where M_i is a $1 \times T n_l$ vector with elements that
$$M_i(j) = \begin{cases} 1 & \text{if } \Delta_i(j) = 1 \\ -1 & \text{if } \Delta_i(j) = 0 \end{cases}$$
and $N_i = \sum_j \Delta_i(j) - 1$. The constraint $M_i \Delta \leq N_i$ is equivalent to imposing $\Delta \neq \Delta_i$ in problem $\mathscr{P}$. Go to step (i).

At the end of the procedure one gets the collection of disjoint polytopes $P_i \subset \mathbb{X} \times \mathbb{U}^T,\ i \in \mathscr{M} = \{1,\dots,N_r\}$ and the affine expression of $\mathbf{Y}$ given in (4). The integer N_r is the total number of the regions found before that the termination condition in step (ii) is fulfilled.

C. Observability Theory

We recall some basic notions of observability theory that we specialize to PWA systems.

Definition 2.1: Two states $x, \hat{x} \in \mathbb{X}$ are indistinguishable in T steps if there exists an admissible input sequence $\mathbf{U} \in \mathbb{U}^T$ such that $[x',\ \mathbf{U}']' \in P$, $[\hat{x}',\ \mathbf{U}']' \in P$ and the system (1) produces the same output sequence over the horizon T, i.e. $\mathbf{Y}(x,\mathbf{U}) = \mathbf{Y}(\hat{x},\mathbf{U})$.

We use the notation $\mathscr{R}_T(x,\hat{x})$ for denoting that x and $\hat{x}$ are indistinguishable. Note that $\mathscr{R}_T$ defines a relation over $\mathbb{X} \times \mathbb{X}$, but, differently from the case of linear systems, it may not be an equivalence relation because the transitive property may not hold [1].

For a state $x \in \mathbb{X}_T^*$, it is clear that $\mathscr{R}_T(x,x)$ and thus the graph of the relation $\mathscr{R}_T$ covers the set $\mathbb{X}_T^* \times \mathbb{X}_T^*$. Next, we provide the definition of observable states and observable regions.

Definition 2.2: A state $x \in \mathbb{X}$ is observable in T steps if for all $\hat{x} \in \mathbb{X}$ and for any input sequence $\mathbf{U} \in \mathbb{U}^T$ such that $[x',\ \mathbf{U}']' \in P$, $[\hat{x}',\ \mathbf{U}']' \in P$, x is distinguishable from $\hat{x} \in \mathbb{X}$, i.e.

$$\forall \hat{x} \in \mathbb{X}, \forall \mathbf{U} \in \mathbb{U}^T \left([x',\ \mathbf{U}']' \in P, [\hat{x}',\ \mathbf{U}']' \in P, \right.$$
$$\left. \mathbf{Y}(x,\mathbf{U}) = \mathbf{Y}(\hat{x},\mathbf{U})\right) \Rightarrow x = \hat{x} \tag{8}$$

Definition 2.3: A set $\mathbb{O}_T \subseteq \mathbb{X}$ is an observability region (in T steps) for the PWA system (1) if all states $x \in \mathbb{O}_T$ are observable. The maximal observability region (in T steps) $\bar{\mathbb{O}}_T$ is the union of all the observability regions. i.e.

$$\bar{\mathbb{O}}_T = \{x \in \mathbb{X} : (8)\} \tag{9}$$

Remark 2.1: The finiteness of the parameter T has a practical meaning. In fact, T is the horizon over which output data must be collected before being able to reconstruct the initial state. Then, in a realistic scenario it is reasonable to fix a maximal horizon of interest T_{max} and classify states that are observable for $T > T_{max}$ as *practically unobservable* [6]. We also highlight that Definition 2.2 is slightly different from the definition of incremental observability given in [6] because a minimum level of distinguishability between different states is not required.

Remark 2.2: Definition 2.2 of observability tailored to the use of observers together with a regulator for PWA systems. Indeed, a state x must be observable not only for a given input sequence $\mathbf{U} \in \mathbb{U}^T$ but for all admissible input sequences in $\mathbb{U}^T$.

D. Piecewise Linear Algebra

In this section we briefly review relevant definitions and results of PL algebra [4], [3]. Definitions are given over $\mathbb{R}^n$, but remain valid over any finite dimensional real vector space V.

Definition 2.4: **PL sets** [4] A PL set of $\mathbb{R}^n$ is the union of a finite number of relatively open polyhedra, i.e. of sets defined by finitely many linear equations $f(x) = a$ and linear inequalities $f(x) < a$.
Note that, in the above definition, the component polyhedra are not necessarily full dimensional.

A general way of constructing PL sets is the following one. Let L be the first order language over an alphabet having the following terms : constant symbols r, free variables x, unary function symbols $r.(\ )$, binary function symbol $+$ and relational symbols $<$ and $=$.

Lemma 2.1: [4] For every sentence S in L with free variables $x_1, \dots, x_n$ the set:

$$Dom(S) \triangleq \{(x_1, \dots, x_n) \in \mathbb{R}^n | S(x_1, \dots, x_n)\}$$

is a PL set. Conversely, for every PL set P in $\mathbb{R}^n$ there exists a sentence S in L such that $P = Dom(S)$.
Moreover, if $S_1(x,y)$ and $S_2(x,y)$ are sentences in L with free variables $x \in \mathbb{R}^{n_1}$, $y \in \mathbb{R}^{n_2}$, $n = n_1 + n_2$, then

$$Dom(\neg S_1) = \mathbb{R}^n \backslash Dom(S_1) \tag{10a}$$
$$Dom(S_1\ \&\ S_2) = Dom(S_1) \cap Dom(S_2) \tag{10b}$$
$$\{x : \exists y,\ S(x,y)\} = \text{Proj}_x(Dom(S)) \tag{10c}$$

Formulae (10a) and (10b) allow to relate the basic Boolean operators appearing in sentences to set operations. Note also that (10c) performs the elimination of the $\exists$ quantifier. Then, a consequence of Lemma 2.1 is that any set defined using existential and universal quantifiers can also be defined using only propositional connectives [4]. Moreover, as highlighted by Sontag [4], formula (10c) shows the central role played by the projection of polyhedra in quantifier elimination. Finally, from Lemma 2.1, one can include in L terms for sets already known to be PL sets.

We can now define the set of unobservable states for system (1) via sentences in L. Let $S(x,\hat{x},\mathbf{U})$ be the sentence: $\left[x',\mathbf{U}'\right]' \in P$, $\left[\hat{x}',\mathbf{U}'\right]' \in P$, $x \neq \hat{x}$ and $\mathscr{R}_T(x,\hat{x})$. Let $S_1(x)$ be the sentence : $\exists \hat{x}$, $\exists \mathbf{U}$ s.t. $S(x,\hat{x},\mathbf{U})$. In view of Definition 2.3, the maximal set of observable states $\bar{\mathbb{O}}_T$ is given by

$$\bar{\mathbb{O}}_T = Dom(\neg S_1)$$

Since S_1 is a sentence in L, from Lemma 2.1 it follows that $\bar{\mathbb{O}}_T$ is a PL set and hence the union of finitely many, possibly non closed polytopes. The main purpose of the paper is to provide an algorithm for computing $\bar{\mathbb{O}}_T$. To this purpose, from (10a) and (10c) one obtains

$$\bar{\mathbb{O}}_T = \mathbb{X}^*_T \backslash Dom(S_1) = \mathbb{X}^*_T \backslash \text{Proj}_x(Dom(S)) \tag{11}$$

The next problem is to express $Dom(S)$ as the union of finitely many polytopes. Quoting Sontag [3], if $\hat{S}$ is a generic sentence in L, $Dom(\hat{S})$ can be found "via three basic algorithms: one for projecting polyhedra on hyperplanes, another for checking feasibility of a linear program and a standard Boolean table". However, this observation does not prevent from using ad-hoc computational tools for speeding up the computation of $Dom(\hat{S})$ for a *specific* sentence $\hat{S}$. In the next section we provide an efficient algorithm for the computation of the set $\text{Proj}_x(Dom(S))$.

III. Computation of Observability Region

For computing $\text{Proj}_x(Dom(S))$ we proceed in two steps: first, we establish a list $\mathscr{I}$ of possibly observable regions. This list is created by checking if there exist pairs of indistinguishable states in the same polytope P_i. This test will be performed using a rank condition on the matrices C_i. Once the list $\mathscr{I}$ is obtained, we look for indistinguishable states in different, possibly observable regions.

For detecting if a region P_i is possibly observable we exploit the following Lemma, inspired by the results in [7].

Lemma 3.1: Let P_i be a not necessarily closed, full dimensional polyhedron and $\text{Int}(P_i)$ be its interior.

i) If rank $(C_i) < n$, then $\forall\ [x',\mathbf{U}']' \in \text{Int}(P_i)$, $\exists\ [\hat{x}',\mathbf{U}']' \in \text{Int}(P_i)$ such that $x \neq \hat{x}$ and $\mathscr{R}_T(x,\hat{x})$.
ii) If $\text{rank}(C_i) = n$, there is no pair of vectors $[x',\mathbf{U}']'$ and $[\hat{x},\mathbf{U}']'$ in P_i such that $x \neq \hat{x}$ and $\mathbf{Y}(x,\mathbf{U}) = \mathbf{Y}(\hat{x},\mathbf{U})$.

Proof: for two vectors $[x',\mathbf{U}']'$ and $[\hat{x}',\mathbf{U}']'$ in P_i, formula (4) shows that $\mathscr{R}_T(x,\hat{x})$ is equivalent to $C_i(x-\hat{x}) = 0$.

i) Let $B([x',\mathbf{U}']')$ be a ball of center $[x',\mathbf{U}']'$ included in P_i. If we suppose that $\text{rank}(C_i) < n$, then $Ker([C_i, 0_{Tm}]) \cap B([x',\mathbf{U}']') \neq \emptyset$. The proof is concluded by noting that it is always possible to choose a point $[\hat{x}',\mathbf{U}']' \in B([x',\mathbf{U}']')$ verifying $x - \hat{x} \in Ker(C_i)$ and $x \neq \hat{x}$. This is a consequence of the full dimensionality of $B([x',\mathbf{U}']')$.
ii) Suppose that $\text{rank}(C_i) = n$, and that there exists two states x and $\hat{x}$ and a control sequence $\mathbf{U}$ such that, $[x',\mathbf{U}']', [\hat{x}',\mathbf{U}']' \in P_i$ and $C_i(x-\hat{x}) = 0$. Since C_i is full

column rank then, $x=\hat{x}$. We conclude that there is no pair of indistinguishable states in $\text{Proj}_x(P_i)$. ■

Remark 3.1: For autonomous piecewise linear systems, the rank condition in point (ii) of Lemma 3.1 matches the condition (19) of Theorem 2 in [7]. At first sight, it may be surprising that Theorem 2 in [7] provides a necessary and sufficient condition for observability while we use the same test just for determining if a region is possibly observable. The difference is that we do not assume that the sequence of switches over the observability horizon is known beforehand. ■

By using the condition $\text{rank}(C_i) < n$ for checking if all states in $\text{Proj}_x(P_i)$ are unobservable, some possibly observable parts of the boundary of $\text{Proj}_x(P_i)$ will be wrongly classified. Our algorithm will thus produce a set of observable states which differs from the maximal observability region by a set of zero Lebesgue measure.

From Lemma 3.1, we deduce that the maximal observability region has zero Lebesgue measure if all the matrices C_i, $i \in \mathscr{M}$ have rank less than n. This enforces a condition on the minimal length of the observation horizon.

Lemma 3.2: A necessary condition for having a full dimensional set of observable states is that T verifies

$$Tp \geq n \tag{12}$$

Proof: C_i are $Tp \times n$ matrices and, apparently, condition (12) is necessary for having $\text{rank}(C_i) = n$. ■

Let $\mathscr{I} \subseteq \mathscr{M}$ be the set collecting all indexes of possibly observable regions (i.e. $k \in \mathscr{I} \Leftrightarrow$ rank $(C_k) = n$). In the second step of our algorithm we test the indistinguishability of pair of states belonging to different regions. For $i \in \mathscr{I}$ and for $j \in \mathscr{M} \backslash \{i\}$ let

$$R_{ij} \triangleq \{(x,\hat{x},\mathbf{U}) : \begin{bmatrix} x \\ \mathbf{U} \end{bmatrix} \in P_i \text{ and } \begin{bmatrix} \hat{x} \\ \mathbf{U} \end{bmatrix} \in P_j \text{ and } \mathscr{R}_T(x,\hat{x})\} \tag{13}$$

Note that the definition of R_{ij} mimics the sentence S given at the end of Section II-D. Moreover, since P_i are polytopes and the map $\mathbf{Y}(x,\mathbf{U})$ is affine on each P_i, the constraints in (13) are linear. We also highlight that the constraint $x \neq \hat{x}$ is implicit in the definition of R_{ij}. Indeed, the same vector $\mathbf{U}$ is used in the conditions $[x', \mathbf{U}']' \in P_i$ and $[\hat{x}', \mathbf{U}']' \in P_j$. Then, $x \neq \hat{x}$ follows from $P_i \cap P_j = \emptyset$.

If the region R_{ij} is non empty (a fact that can be checked through a single linear program) a subset of unobservable states can be computed by projecting R_{ij} on the $x-$coordinates. This is a consequence of formula (10c). In order to discover all possible full dimensional regions of unobservable states the sets R_{ij} must be computed for all $i \in \mathscr{I}$ and for all $j \in \mathscr{M} \backslash \{i\}$.

Let $\mathbb{O}_T$ be defined as

$$\mathbb{O}_T = \mathbb{X}_T^* \backslash \{ \Big(\bigcup_{i \notin \mathscr{I}} \text{Proj}_x(P_i) \Big) \bigcup \Big(\bigcup_{i \in \mathscr{I}} \bigcup_{j \in \mathscr{M}\backslash\{i\}} \text{Proj}_x(R_{ij}) \Big) \} \tag{14}$$

In view of formula (11), an on the basis of the previous discussion, one has that $\mathbb{O}_T$ differs from $\bar{\mathbb{O}}_T$ by a set of zero measure. Since all the sets appearing in the right hand side of (14) are polytopes, the set difference can be computed easily [9]. In particular, free software for computing the *closure* of $\mathbb{O}_T$ exists [12].

Remark 3.2: Note that we did not compute the set R_{ij} for $i = j$. This case has to be treated separately, i.e. using the rank test given in Lemma 3.1. Indeed, since $\mathscr{R}_T(x,x)$, $\forall\ x \in \mathbb{X}_T^*$ and since the constraint $x \neq \hat{x}$ does not appear in (13), it follows that $\text{Proj}_x(R_{ii}) = \text{Proj}_x(P_i)$, $\forall\ i$. Then, from (14), one would always obtain $\mathbb{O}_T = \emptyset$.

IV. Numerical Examples

We first demonstrate, through toy examples, the computational advantages of our algorithm (termed "projection-based") with respect to the one proposed in [1] and based on mp-MILP. The results are compared both in term of computational time and solution complexity, i.e. the number of polytopes used to describe the observability region. We highlight that theoretical results on the worst-case computational complexity of mp-MILP algorithms and algorithms for projecting polyhedra on subspaces are available (see [13] and [8], respectively). However, their use for making a rigorous comparison of the projection-based method with the one given in [1] is far from being trivial. Secondly, we present the results of the projection-based algorithm on a more complex example (a vehicle powertrain with backlash) for which the maximal observability region is not obvious. All the experiments have been performed on a Pentium 4 2.4 Ghz running Matlab 6.5.1. Consider the following academic examples

Example 1:

$$x(t+1) = \begin{cases} x(t) & \text{if } \ |x(t)| \leq 2 \\ \frac{1}{2}x(t) & \text{if } \ |x(t)| \geq 2+\varepsilon \end{cases} \tag{15a}$$

$$y(t) = \begin{cases} x(t) & \text{if } \ |x(t)| \leq 2 \\ 0 & \text{if } \ |x(t)| \geq 2+\varepsilon \end{cases} \tag{15b}$$

Example 2:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t+1) = \begin{cases} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t) & \text{if } \ x_1(t) \geq x_2(t)+\varepsilon \\ \begin{bmatrix} 0.5 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t) & \text{if } \ x_1(t) \leq x_2(t) \end{cases} \tag{16a}$$

$$y(t) = \begin{cases} x_1(t) & \text{if } \ x_1(t) \geq x_2(t)+\varepsilon \\ x_2(t) & \text{if } \ x_1(t) \leq x_2(t) \end{cases} \tag{16b}$$

where ε is a small tolerance that had been set to 0.1 in [1]. Tables I and II summarize the computational performance of the two algorithms for different lengths of the observation horizon. The computational times of the projection-based procedure include also the running time of the mode enumeration algorithm presented in Section II-B.

T	3	4	5	6
mp-MILP based algorithm	28s/ 5 reg	108s/ 7 reg	429s/ 9 reg	1217s/ 11 reg
Projection-based algorithm	0.28s/ 5 reg	0.70s/ 7 reg	2.10s/ 9 reg	5.41s/ 11 reg

TABLE I

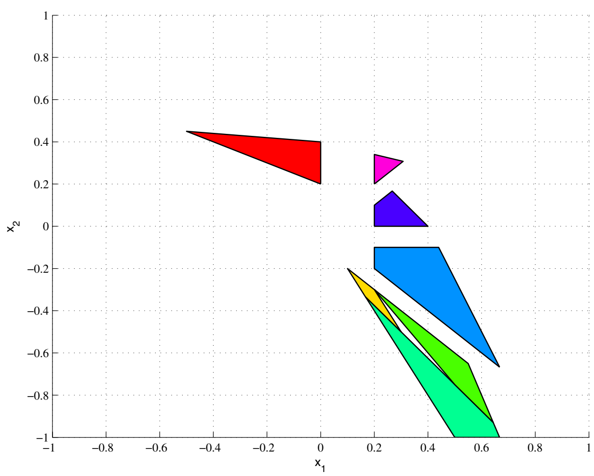
Comparison of results for the example 1

The tables highlight that the projection-based algorithm is considerably faster than the one based on mp-MILP,

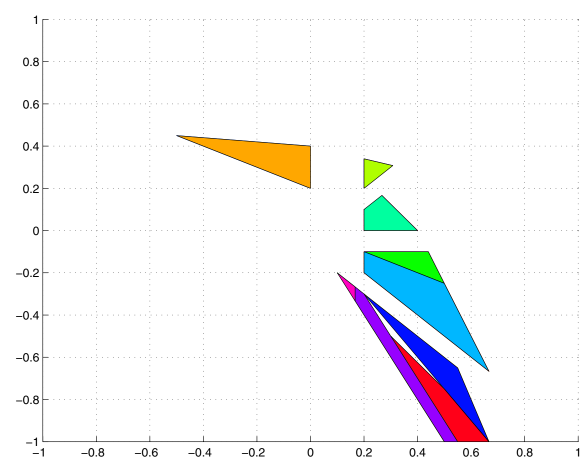
T	3	4	5	6
mp-MILP based algorithm	27s/ 6 reg	92s/ 9 reg	191s/ 6 reg	329s/ 5 reg
Projection-based algorithm	0.27s/ 5 reg	0.42s/ 7 reg	0.45s/ 6 reg	0.49s/ 5 reg

TABLE II
COMPARISON OF RESULTS FOR THE EXAMPLE 2

especially for large horizons. The results also show that the complexity of the projection-based algorithm is always equal or less than the complexity of the other algorithm. To give an idea of the simplification capabilities (in term of solution complexity) of our algorithm we plot in figure 1 the polytopes composing the observability region $\mathbb{O}_4$ for the example 2 found by the two methods.



(a) Projection-based Algorithm



(b) mp-MILP based Algorithm

Fig. 1. The Observability region $\mathbb{O}_4$ for Example 2

Example 3 : Vehicle Powertrain
In the sequel, we compute the maximal observability region for a simplified model of a vehicle powertrain with backlash reported in [14]. The powertrain, represented in figure 2, is

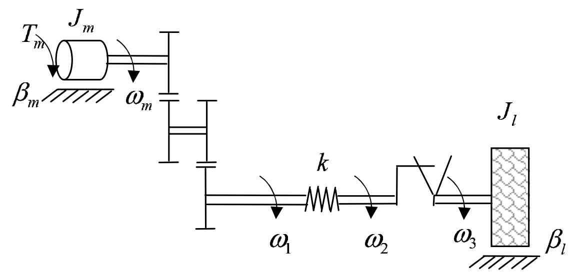


Fig. 2. Powertrain Model

modeled as a two mass system

$$\begin{cases} \dot{\theta} = \tau\omega_m - \omega_l \\ J_m\dot{\omega}_m = T_m - \tau T_w - \beta_m\omega_m \\ J_l\dot{\omega}_l = T_w - \beta_l\omega_l \end{cases} \qquad (17)$$

where the input T_m is the engine torque, ω_m is the engine speed, τ is the gear box ratio, ω_l is the wheel speed, θ is the twist angle and $T_w = kD_\alpha(\theta)$ is the torque due to the shaft elasticity. Moreover, $D_\alpha(.)$ is the dead zone operator defined as:

$$D_\alpha(\theta) = \begin{cases} \theta - \alpha & \text{if } \theta > \alpha \\ 0 & \text{if } |\theta| \leq \alpha \\ \theta + \alpha & \text{if } \theta < -\alpha \end{cases}$$

All the parameters values used can be found in Appendix A of [14]. Bounds on system states are $[0, 500]$ rad/s^{-1} for ω_m, $[-50, 50]$ rad/s^{-1} for ω_l and $[-2, 2]$ rad for θ. The engine torque takes values in $[0, 200]$ $N \cdot m$. The discrete-time PWA model is obtained through the discretization scheme $\dot{x} \simeq (x_{k+1} - x_k)/T_s$, where T_s is the sampling period, chosen equal to $1ms$. The measured output is $y = \omega_m$.

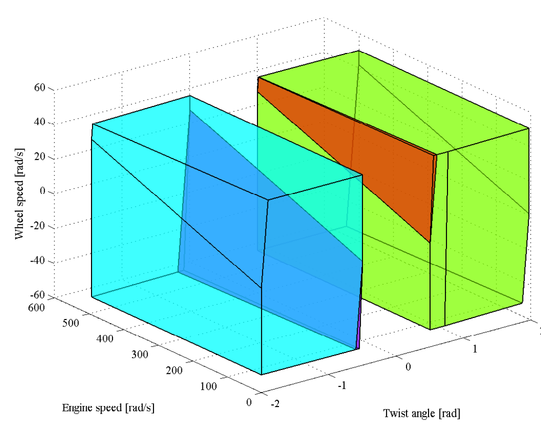


Fig. 3. Observability region for the powertrain example with $T = 3$

We first apply our algorithm with $T = 3$. By Lemma 3.2, this is the minimal length of the observation horizon that may produce a full dimensional observability region. The region $\mathbb{O}_T$ is expected to contain states such that the twist angle lies and stays out of the backlash during the whole observation horizon and for all admissible inputs. Actually, if $|\theta| \leq \alpha$ one has $T_w = 0$ and from (17) it follows that the dynamics of ω_m and ω_l are decoupled. It is apparent that in this case the wheel speed cannot be estimated from the measure of the engine speed. Figure 3 shows the results produced by the projection-based algorithm. If we increase the horizon length to $T = 5$, the maximal observability region (plot in figure 4) becomes larger since more initial states that "lie in the backlash domain" (i.e. with $|\theta| \leq \alpha$) will evolve out of it within the observation horizon, thus becoming observable. This phenomenon is emphasized in

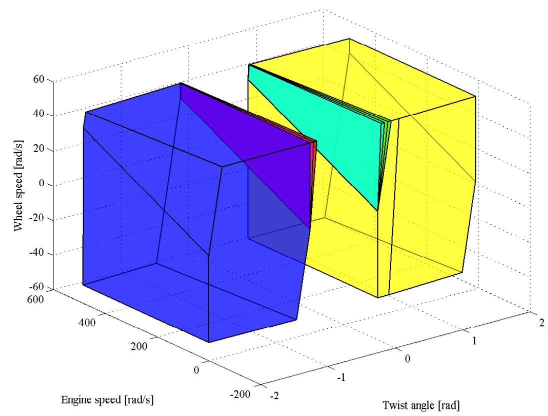


Fig. 4. Observability region for the powertrain example with $T = 5$

figure 5 where the observability regions computed for $T = 3$ (figure 3) and $T = 5$ (figure 4) are projected over the twist angle θ.

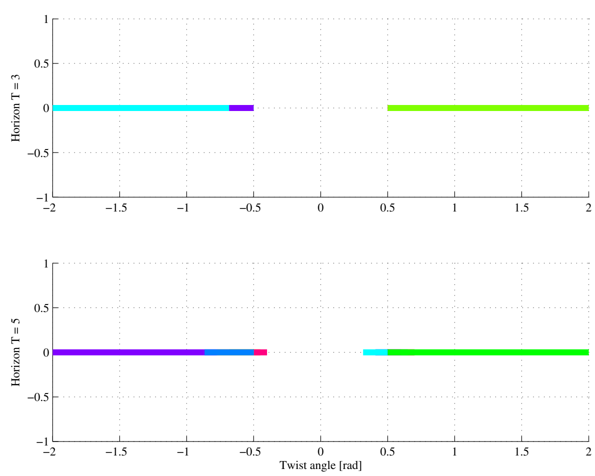


Fig. 5. Projection of $\mathbb{O}_3$ and $\mathbb{O}_5$ on θ for the powertrain example

V. Conclusions

In this paper we proposed an algorithm for computing the maximal observability region of a piecewise affine system. Our method hinges on results from PL algebra and exploits polyhedral computation algorithms for finding the polytopes composing $\bar{\mathbb{O}}_T$ (up to a zero-measure set). In particular, the projection of polytopes on suitable subspaces is used as a tool for performing quantifier elimination. The procedure appears to be computationally very efficient and this fact is partially motivated by the use of the rank test given in Lemma 3.1. We also derived a necessary condition on the minimal length of the observability horizon in order to have a full-dimensional maximal observability region.

The maximal observability region usually has a non trivial structure. Nevertheless, as remarked in [1], its knowledge might play a key role in the synthesis of state observers and output-feedback controllers for PWA systems.

References

[1] G. Ferrari-Trecate and M. Gati, "Computation of observability regions for discrete time hybrid systems," in *Proc. 42th Conference on Decision and Control*, 2003, pp. 1153–1158.

[2] P. Collins and J. van Schuppen, "Observability of piecewise-affine hybrid systems," in *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC 2004*, ser. Lecture Notes in Computer Science, vol. 2993. Philadelphia, Pennsylvania, USA: Springer Verlag, 2004, pp. 265 – 279.

[3] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Trans. Automat. Contr.*, vol. 26, no. 2, 1981.

[4] ——, "Remarks on piecewise-linear algebra," *Pacific Journal of Mathematics*, vol. 98, no. 1, pp. 183–201, 1981.

[5] ——, "Real addition and the polynomial hierarchy," *Inform. Proc. Letters*, vol. 20, pp. 115–120, 1985.

[6] A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 45, no. 10, pp. 1864–1876, 2000.

[7] M. Babaali and M. Egersted, "On the observability of piecewise linear systems," in *Proc. 43th Conference on Decision and Control*, 2004, pp. 26–31.

[8] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski, "Equality set projection: A new algorithm for the projection of polytopes in halfspace representation," Department of Engineering, University of Cambridge, Tech. Rep. CUED/F-INFENG/TR.463, March 2004.

[9] S. V. Rakovic, E. C. Kerrigan, and D. Q. Mayne, "Reachability computations for constrained discrete-time systems with state- and input-dependent disturbances," in *Proc. 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, December 2003.

[10] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–428, March 1999.

[11] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag, 2003, vol. 290.

[12] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari, "Multi parametric toolbox (mpt)," in *Hybrid Systems: Computation and Control,7th International Workshop, HSCC 2004*, ser. Lecture Notes in Computer Science, vol. 2993. Philadelphia, Pennsylvania, USA: Springer Verlag, Mar. 2004, pp. 448–462, http://control.ee.ethz.ch/~mpt.

[13] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," Automatic Control Laboratory, Swiss Federal Institute of Technology, Tech. Rep. AUT02-16, December 2002.

[14] A. Lagerberg, "Control and estimation of automotive powertrains with backlash," Ph.D. dissertation, Chamlers University of Technology, 2004.