Optimization-Based Verification and Stability Characterization of Piecewise Affine and Hybrid Systems

Alberto Bemporad*, Fabio Danilo Torrisi, Manfred Morari

Automatic Control Laboratory, Swiss Federal Institute of Technology
ETH Zentrum - ETL I24.2, CH 8092 Zürich, Switzerland
tel. +41-1-632 6679, fax +41-1-632 1211
bemporad,torrisi,morari@aut.ee.ethz.ch
http://control.ethz.ch/~hybrid

Abstract. In this paper, we formulate the problem of characterizing the stability of a piecewise affine (PWA) system as a verification problem. The basic idea is to take the whole $\mathbb{R}^n$ as the set of initial conditions, and check that all the trajectories go to the origin. More precisely, we test for semi-global stability by restricting the set of initial conditions to an (arbitrarily large) bounded set $\mathcal{X}(0)$, and label as "asymptotically stable in T steps" the trajectories that enter an invariant set around the origin within a finite time T, or as "unstable in T steps" the trajectories which enter a set $\mathcal{X}_{\text{inst}}$ of (very large) states. Subsets of $\mathcal{X}(0)$ leading to none of the two previous cases are labeled as "non-classifiable in T steps". The domain of asymptotical stability in T steps is a subset of the domain of attraction of an equilibrium point, and has the practical meaning of collecting the initial conditions from which the settling time to a specified set around the origin is smaller than T. In addition, it can be computed algorithmically in finite time. Such an algorithm requires the computation of reach sets, in a similar fashion as what has been proposed for verification of hybrid systems. In this paper we present a substantial extension of the verification algorithm presented in [6] for stability characterization of PWA systems, based on linear and mixed-integer linear programming. As a result, given a set of initial conditions we are able to determine its partition into subsets of trajectories which are asymptotically stable, or unstable, or non-classifiable in T steps.

1 Introduction

Hybrid models describe processes which evolve according to dynamics and logic rules. Hybrid systems have recently grown in interest not only for being theoretically challenging [10], but also for their impact on applications, for instance in the automotive industry [3].

* Corresponding author.

An important class of hybrid systems are the so-called *Piecewise Affine* (PWA) systems. These are defined by partitioning the state-space into polyhedral regions, and associating with each region a different linear state-update equation. PWA systems can model a large number of physical processes, such as systems with static nonlinearities (for instance actuator saturation), and can approximate nonlinear dynamics with arbitrary accuracy via multiple linearizations at different operating points. The study of PWA systems is also motivated by the stability and performance analysis of high-performance controllers [20]. In particular, recently in [7] the authors show that a model predictive controller (MPC) for constrained linear systems can be explicitly expressed in closed-form as a continuous and piecewise affine state-feedback law. The resulting closed-loop system is therefore PWA, and criteria for proving stability and robust stability against disturbances and model uncertainties are of fundamental importance.

PWA systems are equivalent to interconnections of linear systems and finite automata, as pointed out by Sontag [26]. Based on different arguments, a similar result was proved constructively in [4], where the authors show that PWA systems are equivalent to the hybrid *mixed logical dynamical* (MLD) systems introduced in [5]. MLD systems are capable to model a broad class of systems arising in many applications: linear hybrid dynamical systems, hybrid automata, nonlinear dynamic systems where the nonlinearity can be approximated by a piecewise linear function, some classes of discrete event systems, linear systems with constraints, etc. Examples of real-world applications that can be naturally modeled within the MLD framework are reported in [5, 6]. The MLD framework allows specifying linear dynamics $x' = Ax + Bu$, any logic proposition, and the interaction between the two. The key idea of the approach consists of embedding the logic part in the state equations by transforming Boolean variables into 0-1 integers, and by expressing the relations as mixed-integer linear inequalities [5].

Despite the fact that PWA systems are just a simple extension of linear systems, they can exhibit very complex behaviors, as typical of nonlinear systems [24]. Blondel and Tsitsiklis [9] showed that even in the simple case of two component subsystems, verifying the stability of autonomous discrete-time PWA systems is either an $\mathcal{NP}$-hard problem (no polynomial-time algorithm), or undecidable. In view of these complexity results, no hope remains of finding criteria for stability of PWA systems as easy as for instance the Routh-Hurwitz rule for linear systems. Stability of each linear subsystem is not enough to guarantee stability of the overall system (and vice versa) [11, 28], as the switching rule between linear dynamics is fundamental for stability of the interconnection. Some criteria for stability of PWA systems were recently proposed, which are based on piecewise quadratic Lyapunov functions computed by solving linear matrix inequalities (LMI) [16], and multiple Lyapunov functions methods [11]. However, LMI based approaches have the drawback of being conservative, the more conservative the larger the number of regions in the polyhedral partition of the state space.

Complexity results were also shown in [4] for $\mathcal{NP}$-completeness of observability analysis, and undecidability of reachability in the context of *formal verifica-*

tion of hybrid automata is well known [1, 18]. The problem of formal verification can be simply stated as follows: For a given set of initial conditions and disturbances, certify that all possible trajectories never enter a set of unsafe states, or possibly provide a counterexample. In spite of this complexity, several tools for formal verification of hybrid systems have been proposed in the literature, mainly for linear hybrid automata [15, 19].

In this paper, we formulate the problem of characterizing the stability of a PWA system as a verification problem. The basic idea is to check for reachability from an (arbitrarily large) bounded set $\mathcal{X}(0)$ of initial conditions to (*i*) a set around the origin, and (*ii*) a set of very large (=unsafe) states. More precisely, we label as "asymptotically stable in T steps" the trajectories that enter an invariant set around the origin within a finite time T, or as "unstable in T steps" the trajectories which enter a (very large) set $\mathcal{X}_{\text{inst}}$. Subsets of $\mathcal{X}(0)$ leading to neither of the two previous cases are non-classified. Such a verification problem of "practical" stability is decidable. Many undecidable problems can be approximated by decidable ones which are equivalent from a practical point of view. The decidable algorithm shown in [4] for analysis of observability is another example of such a philosophy.

In order to solve the problem of verification of stability, we substantially extend the algorithm proposed in [6]. Safety tests and reach set computation are done via linear programming (LP), switching detection via mixed-integer linear programming (MILP), and approximation of the reach set by using tools from computational geometry. In particular, with respect to [6], we make the algorithm more efficient, and use an algorithm for arbitrarily precise inner and outer approximation of polyhedra [8].

The approach followed in this paper is related to the idea of *robust simulation* [17], which consist of simulating entire set evolutions rather than single trajectories for stability and performance analysis. In [17] the author tests for finite time stability by computing an outer approximation of the reach set via mathematical programming. However, an outer approximation is performed at each time step in order to bound the complexity of the reach set. It turns out that the approach provides only a sufficient condition to conclude about the stability of the initial set. On the contrary, in this paper an exact characterization of the initial set is obtained by first applying a verification algorithm to the system, and then by refining the results through linear programming. By removing all conservativeness, this allows partitioning the initial set into three subsets: (*i*) states belonging to the domain of asymptotic stability in T steps, (*ii*) states belonging to the domain of instability in T steps, and (*iii*) states which are non-classifiable in T steps.

2 Hybrid and Piecewise Affine Models

Several modeling frameworks were proposed in the literature. Two main categories were successfully adopted for analysis and synthesis purposes [10]: *hybrid control systems* [1, 2, 5, 21, 22], which consist of the interaction between

continuous dynamical systems and discrete/logic automata, and *switched systems* [11, 16, 25], where the state-space is partitioned into regions, each one being associated to a different continuous dynamics.

Switched systems defined by a polyhedral partition of the state-space and linear dynamic equations are the so-called *piecewise affine* (PWA) systems

$$x(t+1) = A_i x(t) + B_i u(t) + f_i\,, \text{ for } x(t) \in \mathcal{C}_i \triangleq \{x:\ H_i x \leq K_i\} \qquad (1)$$

where $x \in X \subseteq \mathbb{R}^n$, $u \in \mathbb{R}^m$, $\{\mathcal{C}_i\}_{i=0}^{s-1}$ is a polyhedral partition of the sets of states X, and f_i is a constant vector. A *trajectory* is the collection of vectors $\{x(0), \dots, x(t), \dots\}$ satisfying the difference equation (1). Without additional hypotheses on continuity of the piecewise affine state-update mapping, definition (1) is not well posed in general, as the state-update function is twice (or more times) defined over common boundaries of sets $\mathcal{C}_i$ (the boundaries will be also referred to as *guardlines*). This is a technical issue which can be avoided as in [25].

In [4] the authors show that PWA systems are equivalent to the *mixed logic dynamical* (MLD) systems introduced in [5]. These are hybrid (control) systems defined by the interaction of logic, finite state machines, and linear discrete-time systems, defined by the equations

$$x(t+1) = \mathcal{A}x(t) + \mathcal{B}_1 u(t) + \mathcal{B}_2 \delta(t) + \mathcal{B}_3 z(t) \qquad (2a)$$

$$\mathcal{E}_2 \delta(t) + \mathcal{E}_3 z(t) \leq \mathcal{E}_1 u(t) + \mathcal{E}_4 x(t) + \mathcal{E}_5 \qquad (2b)$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is a vector of continuous and binary states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ are the inputs, and $\delta \in \{0,1\}^{r_\ell}$, $z \in \mathbb{R}^{r_c}$ represent auxiliary binary and continuous variables respectively, which are introduced when transforming logic relations into mixed-integer linear inequalities [23, 27], and $\mathcal{A}$, $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{E}_1$, $\dots$,$\mathcal{E}_5$ are matrices of suitable dimensions. Throughout the paper, we will assume that both the PWA and the MLD forms are available. Their complementary role in the verification algorithm will be discussed later.

3 Stability Characterization Problem

As mentioned in the introduction, determining the stability of PWA systems can be a complex task. Nevertheless, we aim at estimating the domains of attraction of equilibrium points, and the set of initial conditions from which the state trajectory reaches magnitudes greater than an arbitrarily large value.

For simplicity of exposition, from now on we will assume that the system is piecewise linear ($f_i = 0$, for all $i = 0, \dots, s-1$), and autonomous ($B_i = 0$ for all $i = 0, \dots, s-1$)[1], and that the only equilibrium point (the origin) belongs to the

[1] Robust stability questions in the presence of disturbances $u(t) \in \mathcal{U}$, where $\mathcal{U}$ is a given bounded set, can be similarly formulated.

interior of one of the sets of the partition[2], which by convention will be referred to as $\mathcal{C}_0$. Denote by $\mathcal{D}_\infty(0) \subseteq \mathbb{R}^n$ the (unknown) domain of attraction of the origin (if the origin is unstable then $\mathcal{D}_\infty(0) = \{0\}$). Given an (arbitrarily large) bounded set $\mathcal{X}(0)$ of initial conditions, we want to characterize $\mathcal{D}_\infty(0) \bigcap \mathcal{X}(0)$.

A necessary condition for the origin to be asymptotically stable is that the matrix A_0 associated with the region $\mathcal{C}_0$ is strictly Hurwitz. Under this assumption, we can compute an invariant set in $\mathcal{C}_0$. In particular, we compute the *maximum output admissible set* (MOAS) $\mathcal{X}_\infty \subseteq \mathcal{C}_0$. $\mathcal{X}_\infty$ is the largest invariant set contained in $\mathcal{C}_0$, which by [14, Th.4.1] is a polyhedron with a finite number of facets, and is computed through a finite number of linear programs (LP's) [14][3].

In order to circumvent the undecidability of stability mentioned above, we define the following

Definition 1. *Consider the PWA system (1), and let the origin* $0 \in \mathring{\mathcal{C}}_0 \triangleq \{x : H_0 x < K_0\}$*, and* A_0 *be strictly Hurwitz. Let* $\mathcal{X}_\infty$ *be the maximum output admissible set (MOAS) in* $\mathcal{C}_0$*, which is an invariant for the linear system* $x(t+1) = A_0 x(t)$*. Let* T *be a finite time horizon. Then, the set* $\mathcal{X}(0) \subseteq \mathbb{R}^n$ *of initial conditions is said to belong to the* domain of attraction in T steps $\mathcal{D}_T(0)$ *of the origin if* $\forall x(0) \in \mathcal{X}(0)$ *the corresponding final state* $x(T) \in \mathcal{X}_\infty$*.*

Note that $\mathcal{D}_T(0) \subseteq \mathcal{D}_{T+1}(0) \subseteq \mathcal{D}_\infty(0)$, and $\mathcal{D}_T(0) \to \mathcal{D}_\infty(0)$ as $T \to \infty$. The horizon T is a practical information about the speed of convergence of the PWA system to the origin.

Definition 2. *Consider the PWA system (1), and let* $\mathcal{X}_{\text{inst}} \subseteq \mathbb{R}^n$ *The set* $\mathcal{X}(0) \subseteq \mathbb{R}^n$ *of initial conditions is said to belong to the* domain of instability in T steps $\mathcal{I}_T(0)$ *if* $\forall x(0) \in \mathcal{X}(0)$ *there exists* t*,* $0 \le t \le T$ *such that* $x(t) \in \mathcal{X}_{\text{inst}}$*.*

In Definition (2), the set $\mathcal{X}_{\text{inst}}$ must be interpreted as a set of "very large" states. Although instability in T steps does not guarantee instability (for any finite T, a trajectory might reach $\mathcal{X}_{\text{inst}}$ and converge back to the origin), it has the practical meaning of labeling as "unstable" the trajectories whose magnitude is unacceptable, for instance because the PWA system is no longer valid as a model of the real system. Instability in T steps represents a condition of *loss of safety* for the PWA system.

As $\mathcal{D}_T(0)$ and $\mathcal{I}_T(0)$ can have a nonempty intersection, we introduce the following

[2] The hypothesis of having equilibria only in the interiors of sets $\mathcal{C}_i$, although restrictive, is certainly satisfied when (1) is the result of the linearization of a nonlinear system around different equilibria, and is needed later for easily computing nonempty invariant sets. Moreover, the approach of this paper can be straightforwardly extended to handle multiple equilibria of the PWA system which are not on the border of the polyhedral partition. These can be easily detected by standard linear analysis, and a maximum output admissible sets can be computed for each equilibrium.

[3] If the effect of perturbations $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, where $\mathcal{U}$ is a given bounded set of disturbances and $B_0 \neq 0$, has to be taken into account $\mathcal{X}_\infty$ is the largest invariant set under disturbance excitation, and can be computed as proposed in [13].

Definition 3. *Consider the PWA system (1). The set $\mathcal{X}(0) \subseteq \mathbb{R}^n$ of initial conditions is said to belong to the* domain of safe stability in T steps $\mathcal{S}_T(0)$ *if* $\mathcal{S}_T(0) \subseteq \mathcal{D}_T(0)$ *and* $\mathcal{S}_T(0) \bigcap \mathcal{I}_T(0) = \emptyset$.

Definition 3 describes trajectories which asymptotically converge to the origin without crossing the set $\mathcal{X}_{\text{inst}}$.

Given a set of initial conditions $\mathcal{X}(0)$, we aim at finding subsets of $\mathcal{X}(0)$ which are safely asymptotically stable ($\mathcal{X}(0) \bigcap \mathcal{S}_T(0)$), and subsets which lead to practical instability in T steps ($\mathcal{X}(0) \bigcap \mathcal{I}_T(0)$). Subsets of $\mathcal{X}(0)$ leading to none of the two previous cases are labeled as *non-classifiable in T steps* As we will use linear optimization tools, we assume that $\mathcal{X}(0)$ and $\mathbb{R}^n \backslash \mathcal{X}_{\text{inst}}$ are convex polyhedral sets. Typically, non-classifiable subsets shrink and eventually disappear for increasing T.

3.1 Switching Sequences

The evolution of the PWA system (1) for $u(t) = 0$, $f_i = 0$, $\forall i = 0, \dots, s-1$, is given by

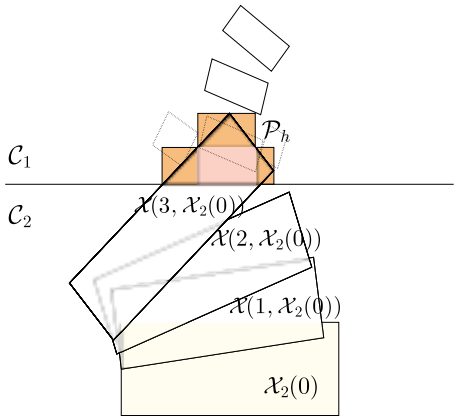
$$x(t) = A_{i(t-1)} A_{i(t-2)} \cdots A_{i(0)} x(0) \tag{3}$$

where in (3) $i(k) \in \{0, \dots, s-1\}$ is the index such that $H_{i(k)} x(k) \leq K_{i(k)}$, $k = 0, \dots, t-1$, is satisfied. The previous questions of practical stability can be answered once all the switching sequences $I(t) \triangleq \{i(0), \dots, i(t-1)\}$ leading to $\mathcal{X}_\infty$ or $\mathcal{X}_{\text{inst}}$ from $\mathcal{X}(0)$ are known. In fact, for safe stability in T steps it is enough to check that the reach set at time T, $\mathcal{X}(T, \mathcal{X}(0)) \triangleq A_{i(T-1)} A_{i(T-2)} \cdots A_{i(0)} \mathcal{X}(0)$, satisfies the set inclusion $\mathcal{X}(T, \mathcal{X}(0)) \subseteq \mathcal{X}_\infty$ for all admissible switching sequences $I(T)$. However, the number of all possible switching sequences $I(T)$ is combinatorial with respect to T and s, and any enumeration method would be impractical. In the next section we show that a verification algorithm can be used to avoid such an enumeration.

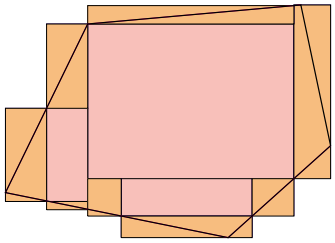
4 Verification

In order to determine admissible switching sequences $I(t)$, we need to exploit the special structure of PWA systems (1). This allows an easy computation of the reach set, as long as the evolution remains within a single region $\mathcal{C}_i$. Whenever the reach set crosses a guardline and enters a new region $\mathcal{C}_j$, a new reach set computation based on the j-th linear dynamics is computed, as shown in Fig. 1(a).

Let $\mathcal{X}(0)$ be a convex polyhedral set, and partition it into subregions $\mathcal{X}_i(0) \triangleq \mathcal{X}(0) \bigcap \mathcal{C}_i$, $i = 0, \dots, s-1$. For all nonempty sets $\mathcal{X}_i(0)$, computing the evolution $\mathcal{X}(T, \mathcal{X}_i(0))$ requires: (*i*) the reach set $\mathcal{X}(t, \mathcal{X}_i(0)) \bigcap \mathcal{C}_i$, i.e. the set of evolutions at time t in $\mathcal{C}_i$ from $\mathcal{X}_i(0)$; (*ii*) crossing detection of the guardlines $\mathcal{P}_h \triangleq \mathcal{X}(t, \mathcal{X}_i(0)) \bigcap \mathcal{C}_h \neq \emptyset$, $\forall h = 0, \dots, i-1, i+1, \dots, s-1$; (*iii*) elimination of redundant constraints and approximation of the polyhedral representation of



(a) Reach set evolution, guardline crossing, outer approximation of a new intersection



(b) Outer rectangular approximation of a polytope

Fig. 1. Reachability Analisys

the new regions $\mathcal{P}_h$ (approximation is desirable, as the number of facets of $\mathcal{P}_h$ can grow linearly with time); (*iv*) detection of emptiness of $\mathcal{X}(t, \mathcal{P}_h)$ (emptiness happens when all the evolutions have crossed the guardlines), detection of safe stability $\mathcal{X}(t, \mathcal{P}_h) \subseteq \mathcal{X}_\infty$, detection of practical instability $\mathcal{X}(t, \mathcal{P}_h) \subseteq \mathcal{X}_{\text{inst}}$ (these three will be referred to as *fathoming* conditions).

4.1 Reach Set Computation

Let the set of initial conditions be defined by the polyhedral representation $\mathcal{X}(0) \triangleq \{x : \ S_0 x \leq T_0\}$. The subset S of $\mathcal{X}(0)$ whose evolution lies in $\mathcal{C}_i$ for t steps is given by

$$S = \left\{ x \in \mathbb{R}^n : \begin{array}{c} S_0 x \leq T_0 \\ H_i A_i^k x \leq K_i, \ k = 0, \ldots, t \end{array} \right\} \tag{4}$$

As S is a polyhedral set, the reach set $\mathcal{X}(t, \mathcal{X}_i(0)) \bigcap \mathcal{C}_i = A_i^t S$ is a polyhedral set as well. In the presence of input disturbances and nonzero offsets f_i, $S = \{x \in \mathbb{R}^n : S_0 x \leq T_0, \ H_i(A_i^k x + \sum_{j=0}^{k-1} A_i^j [B_i u(k-1-j) + f_i]) \leq K_i, \ k = 0, \ldots, t\}$, which is a polyhedron in the augmented space of tuples $(x, u(0), \ldots, u(t-1))$. A compact representation of the set $\mathcal{X}(t, \mathcal{X}_i(0)) \bigcap \mathcal{C}_i$ (as inequalities over the final state $x(t)$) can be computed by a geometric projection procedure, for which efficient tools exist, e.g. [12].

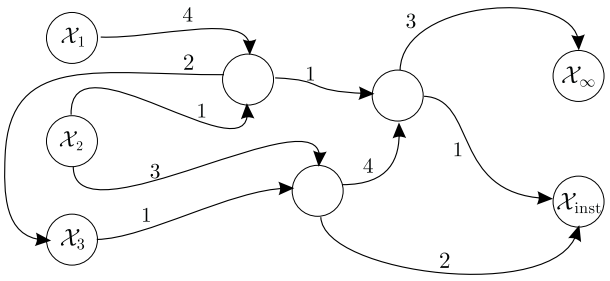


Fig. 2. Graph of evolution G

4.2 Guardline Crossing Detection

Switching detection amounts to finding all possible new regions $\mathcal{C}_h$'s entered by the reach set at the next time step, i.e. nonempty sets $\mathcal{P}_h \triangleq \mathcal{X}(t, \mathcal{X}_i(0)) \bigcap \mathcal{C}_h$, $h \neq i$. Rather then enumerating and checking nonemptiness for all $h = 0, \ldots, i-1, i+1, \ldots, s-1$, we can exploit the equivalence between PWA systems and MLD models (2), and solve the switching detection problem via mixed-integer linear programming. More in detail, in the MLD form the condition $x(t) \in \mathcal{C}_h$ is associated to the condition $\delta(t) = \delta_h \in \{0,1\}^{r_\ell}$, for instance $x(t) \in \mathcal{C}_5 \Leftrightarrow \delta(t) = [1\ 0\ 1]'$. Switching detection amounts to finding all feasible vectors $\delta(t) \in \{0,1\}^{r_\ell}$ which are compatible with the constraints in (2) plus the constraint $x(t-1) \in \mathcal{X}(t-1, \mathcal{X}_i(0)) \cap \mathcal{C}_i$. Such a problem is a mixed-integer linear feasibility test (MILFT), and can be efficiently solved through standard recursive branch and bound procedures. Thus, in the average case the MLD form (through the branch and bound algorithm) requires only a very small number of feasibility tests, while the PWA form would require for enumerating and solving a feasibility test for all the possible regions.

4.3 Approximation of Intersection

The computation of the reach set proceeds in each region $\mathcal{C}_h$ from each new intersection $\mathcal{P}_h$. A new reach set computation is started from $\mathcal{P}_h$, unless $\mathcal{P}_h$ is contained in some larger subset of $\mathcal{C}_h$ which has already been explored. As in principle the number of facets of $\mathcal{P}_h$ grows linearly with time, we need to approximate $\mathcal{P}_h$ so that its complexity is bounded (and therefore reach set computation from $\mathcal{P}_h$ has a limited complexity with respect to the initial region), and checking for set inclusion is a simple task. Hyper-rectangular approximations are the best candidates, as set inclusion between hyper-rectangles reduces to a simple comparison of the coordinates of the vertices. On the other hand, a crude rectangular outer approximation of $\mathcal{P}_h$ can lead to explore large regions which are not reachable from the initial set $\mathcal{X}(0)$, as they are just introduced by the approximation itself. In [8] the authors propose an iterative method for inner and outer approximation which is based on linear programming, and approximates with

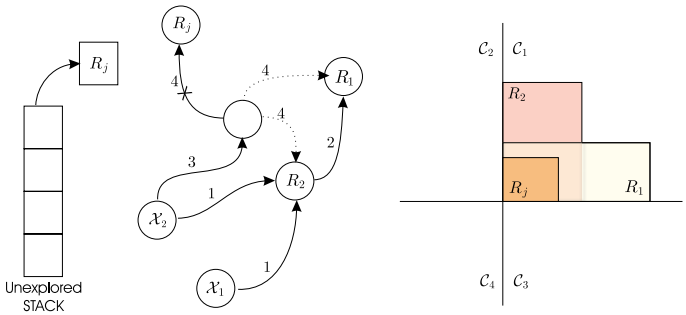


Fig. 3. Adding and removing nodes to the graph G

arbitrary precision polytopes by a collection of hyper-rectangles, as depicted in Fig. 1(b).

4.4 Fathoming

In Sect. 4.1 we showed how to compute the evolution of the reach set $\mathcal{X}(t, \mathcal{P}_h)$ inside a region $\mathcal{C}_i$. The computation is stopped once one of the following happens:

1. The set $\mathcal{X}(t, \mathcal{P}_h) \bigcap \mathcal{C}_i$ is empty. This means that the whole evolution has left region $\mathcal{C}_i$.
2. $\mathcal{X}(t, \mathcal{P}_h) \subseteq \mathcal{X}_\infty$, i.e. all possible evolutions from $\mathcal{P}_h$ are safely stable.
3. $\mathcal{X}(t, \mathcal{P}_h) \subseteq \mathcal{X}_{\text{inst}}$, i.e. all possible evolutions from $\mathcal{P}_h$ have violated the condition for safe stability.
4. The time $t > T$.

These conditions can be checked through linear programming.

4.5 Graph of Evolution

The result of the exploration algorithm detailed in the previous sections can be conveniently represented on a graph G (Fig. 2). The nodes of G represent sets from which a reach set evolution is computed, and an oriented arc of G connects two nodes if a transition exists between the two correspoding sets. Each arc has an associated weight which represents the time-steps needed for the transition. The graph has initially no arc, and nonempty initial sets $\mathcal{X}_i(0)$ and $\mathcal{X}_\infty$, $\mathcal{X}_{\text{inst}}$ as nodes. As long as a new intersection $\mathcal{X}(t, \mathcal{X}_i(0)) \bigcap \mathcal{C}_h$ is detected, it is approximated by a collection of hyper-rectangles, as described in Sect. 4.3. Each hyper-rectangle becomes a new node in G, and is connected by a weighted arc from $\mathcal{X}_i(0)$. In addition, each hyper-rectangle is pushed on a stack of sets to be explored.

Before starting a new reach set computation from a set R_j extracted from the stack, we check for inclusion of R_j in other nodes of G. If this happens, say $R_j \subseteq R_1$ and $R_j \subseteq R_2$ as in Fig. 3, the node associated with R_j is removed from

G, and all arcs pointing to R_j are directed to both R_1 and R_2 (dotted arrows). Finally, whenever the reach set hits $\mathcal{X}_\infty$ (or $\mathcal{X}_{\text{inst}}$), an arc is drawn from $\mathcal{P}_h$ to $\mathcal{X}_\infty$ (or $\mathcal{X}_{\text{inst}}$).

After the verification algorithm terminates, the oriented paths on G from initial nodes $\mathcal{X}_i(0)$ to terminal nodes $\mathcal{X}_\infty$ and $\mathcal{X}_{\text{inst}}$ determine a superset of feasible switching sequences $I(t) = \{i(0), \dots, i(t-1)\}$. In fact, because of the outer approximation of new intersections $\mathcal{P}_h$, not all switching sequences are feasible. Nevertheless, feasibility can be simply tested via linear programming. Once all feasible switching sequences $I(t)$ have been identified, the partition of the initial set into safely stable and unstable regions is determined by the sets $A_{i(t-1)}A_{i(t-2)}\dots A_{i(0)}\mathcal{X}(0)$, $t \le T$.

Algorithm 1.

1 initialize GRAPH with nonempty initial nodes $\mathcal{X}_i(0)$, $i = 0, \dots, n_0$, and disjoint final nodes $\mathcal{F}_j$, $j = 1, \dots, n_f$;
2 **push** in STACK $\mathcal{X}_i(0)$, $i = 0, \dots, n_0$;
3 **while** STACK nonempty **do**
4 **pop** region R_j from STACK, and **let** i such that $R_j \subseteq \mathcal{C}_i$;
5 **if** no region in GRAPH includes R_j **then**
6 $t \leftarrow t^* \triangleq$ minimum arrival time from initial nodes to R_j;
7 **for** $j = 1, \dots, n_f$ **do**
8 **if** $\mathcal{X}(t, R_j) \subseteq \mathcal{F}_j$ **then go to** *20*;
9 **if** $\mathcal{X}(t, R_j) \cap \mathcal{F}_j \neq \emptyset$ **then**
10 connect R_j to $\mathcal{F}_j$ with weight $t - t^*$;
11 $t \leftarrow t + 1$;
12 $\mathcal{X}(t, R_j) = A_i\mathcal{X}(t-1, R_j) + B_i\mathcal{U} + \{f_i\}$;
13 **for** all $h \neq i$ such that $\mathcal{P}_h \triangleq \mathcal{C}_h \bigcap \mathcal{X}(t, R_j) \neq \emptyset$ **do**
14 insert $\mathcal{P}_h$ in GRAPH and connect R_j to $\mathcal{P}_h$ with weight $t - t^*$;
15 **push** $\mathcal{P}_h$ on STACK;
16 $\mathcal{X}(t, R_j) \leftarrow \mathcal{X}(t, R_j) \bigcap \mathcal{C}_j$;
17 **if** $\mathcal{X}(t, R_j) \neq \emptyset$ and $t < T$ **then go to** *9*;
18 **else**
19 redirect all arcs to R_j to all regions R_h in GRAPH, $R_h \supseteq R_j$;
20 **end** .

4.6 Verification Algorithm

The techniques proposed in the previous sections for verification of PWA systems are summarized in Algorithm 1. In step *1*, $\mathcal{F}_1 = \mathcal{X}_\infty$ and $\mathcal{F}_2 = \mathcal{X}_{\text{inst}}$. Step *6* is computed by standard techniques for shortest path computation, while step *13*

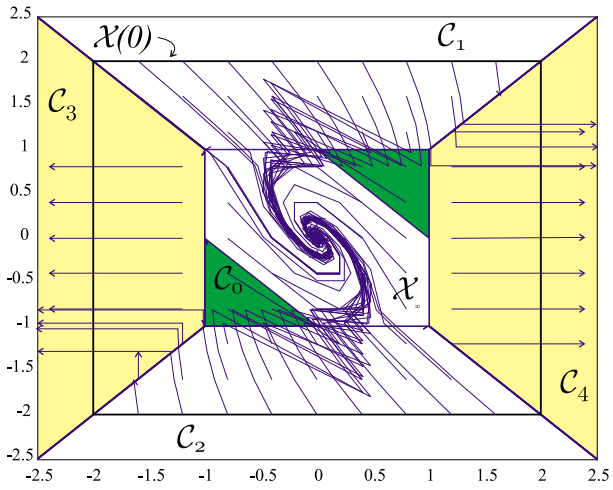


Fig. 4. PWA system (5), initial region $\mathcal{X}(0)$, MOAS $\mathcal{X}_\infty$, and trajectories of the system

by branch and bound. In step *14*, the collection of hyper-rectangles computed by outer approximating $\mathcal{P}_h$ are put on the stack, rather than $\mathcal{P}_h$.

Note that Algorithm 1 can be generalized to verification purposes, by interpreting $\mathcal{F}_1$ as a set of target states, and $\mathcal{F}_2$ as a set of unsafe states. Moreover, linear programs can be performed during reach set computation in order to determine the range of given state components. The algorithm can be extended to include disturbances $u(t) \in \mathcal{U}$, where $\mathcal{U}$ is a given bounded polyhedral set, at the price of more complicate computations (see footnote 3).

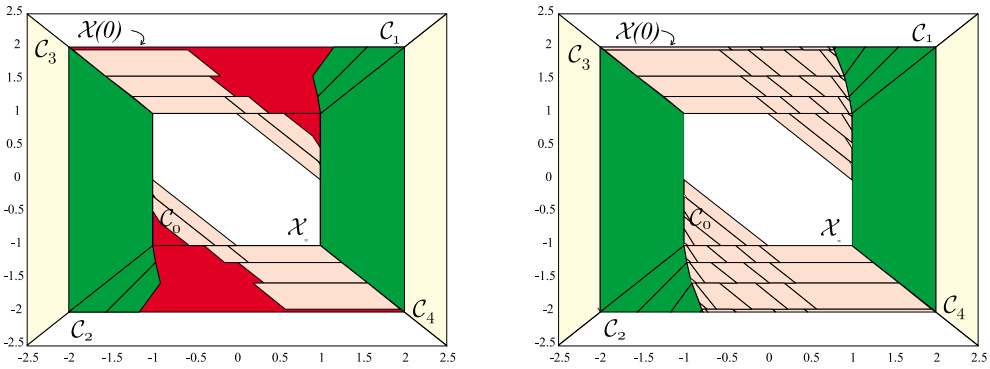
We finally remark that the termination of Algorithm 1 after a finite time is guaranteed because no exploration is performed for $t > T$ (step *17*).

5 An Example

Consider the PWA system

$$x(t+1) = \begin{cases} \begin{bmatrix} 0 & -.5 \\ 1 & 1 \end{bmatrix} x(t) & \text{if } \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x(t) \le \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} (\mathcal{C}_0) \\ \begin{bmatrix} .9 & .1 \\ 0 & .8 \end{bmatrix} x(t) & \text{if } \begin{bmatrix} 0 & -1 \\ 1 & -1 \\ -1 & -1 \end{bmatrix} x(t) \le \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} (\mathcal{C}_1) \\ \begin{bmatrix} .9 & .1 \\ 0 & .8 \end{bmatrix} x(t) & \text{if } \begin{bmatrix} 0 & 1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} x(t) \le \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} (\mathcal{C}_2) \\ \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} x(t) & \text{if } \begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} x(t) \le \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} (\mathcal{C}_3) \\ \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} x(t) & \text{if } \begin{bmatrix} -1 & 0 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} x(t) \le \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} (\mathcal{C}_4) \end{cases} \tag{5}$$

and let $\mathcal{X}(0) = \{x \in \mathbb{R}^2 : \|x\|_\infty \le 2\}$, $\mathcal{X}_{\text{inst}} = \{x \in \mathbb{R}^2 : \|x\|_\infty \ge 10\}$. The origin is asymptotically stable, as A_0 has eigenvalues $\frac{1}{2} \pm j\frac{1}{2}$. The corresponding



(a) $T = 5$ (b) $T = 12$

Fig. 5. Stability characterization of system (5)

maximum output admissible set in $\mathcal{C}_0$

$$\mathcal{X}_\infty = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\} \tag{6}$$

was computed by the algorithm in [14]. A simulation of the system from different initial conditions is depicted in Fig. 4, which shows that the trajectories either converge to the origin or diverge to infinity. We characterize the set of initial conditions by running Algorithm 1. The results are shown in Fig. 5. With the time horizon $T = 5$, not all the set of initial conditions is classified for stability (the darkest subsets are non-classifiable in 5 steps). By augmenting the time horizon, the region of states which are non-classifiable in T steps shrinks, and disappears for $T = 12$. Algorithm 1 is implemented in Matlab 5.3 on a Pentium II 400, and requires 57 s to produce the plot in Fig. 5(b) ($T = 12$).

Acknowledgments

The authors thank the partners of the Esprit Project 26270 and Giancarlo Ferrari Trecate for interesting discussions. This research has been supported by the Swiss National Science Foundation.

References

[1] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In A.P. Ravn R.L. Grossman, A. Nerode and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer Verlag, 1993.

[2] A. Asarin, O. Maler, and A. Pnueli. On the analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138:35–65, 1995.

[3] A. Balluchi, M. Di Benedetto, C. Pinello, C. Rossi, and A. Sangiovanni-Vincentelli. Hybrid control for automotive engine management: the cut-off case. In T.A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 13–32. Springer Verlag, 1998.

[4] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Trans. Automatic Control*, to appear. `http://control.ethz.ch/`.

[5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[6] A. Bemporad and M. Morari. Verification of hybrid systems via mathematical programming. In F.W. Vaandrager and J.H. van Schuppen, editors, *Hybrid Systems: Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 31–45. Springer Verlag, 1999.

[7] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. Technical Report AUT99-16, Automatic Control Lab, ETH Zürich, Switzerland, 1999.

[8] A. Bemporad and F.D. Torrisi. Inner and outer approximation of polytopes using hyper-rectangles. Technical Report AUT00-02, Automatic Control Lab, ETH Zurich, 2000.

[9] V.D. Blondel and J.N. Tsitsiklis. Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35:479–489, March 1999.

[10] M. S. Branicky. *Studies in hybrid systems: modeling, analysis, and control*. PhD thesis, LIDS-TH 2304, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[11] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automatic Control*, 43(4):475–482, April 1998.

[12] K. Fukuda. *cdd/cdd+ Reference Manual*. Institute for operations Research ETH-Zentrum, ETH-Zentrum, CH-8092 Zurich, Switzerland, 0.61 (cdd) 0.75 (cdd+) edition, December 1997.

[13] E.G. Gilbert and I. Kolmanovsky. Maximal output admissible sets for discrete-time systems with disturbance inputs. In *Proc. American Contr. Conf.*, pages 2000–2005, 1995.

[14] E.G. Gilbert and K. Tin Tan. Linear systems with state and control constraints: the theory and applications of maximal output admissible sets. *IEEE Trans. Automatic Control*, 36(9):1008–1020, 1991.

[15] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: a model checker for hybrid systems. *Software Tools for Technology Transfer*, 1:110–122, 1997.

[16] M. Johannson and A. Rantzer. Computation of piece-wise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Automatic Control*, 43(4):555–559, 1998.

[17] M. Kantner. Robust stability of piecewise linear discrete time systems. In *Proc. American Contr. Conf.*, pages 1241–1245, Evanston, IL, USA, 1997.

[18] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: a class of decidable hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 179–208. Springer Verlag, 1993.

[19] S. Kowalewski, O. Stursberg, M. Fritz, H. Graf, I. Hoffmann, J. Preußig, M. Remelhe, S. Simon, and H. Treseler. A case study in tool-aided analysis of

discretely controlled continuos systems: the two tanks problem. In *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*, pages 163–185. Springer-Verlag, 1999.

[20] D. Liberzon and A.S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, October 1999.

[21] J. Lygeros, D.N. Godbole, and S. Sastry. A game theoretic approach to hybrid system design. In R. Alur and T. Henzinger, editors, *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, 1996.

[22] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.

[23] R. Raman and I. E. Grossmann. Relation between milp modeling and logical inference for chemical process synthesis. *Computers Chem. Engng.*, 15(2):73–84, 1991.

[24] E. Sontag. From linear to nonlinear: Some complexity comparisons. In *Proc. 34th IEEE Conf. on Decision and Control*, pages 2916–2920, 1995.

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

[26] E.D. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete-time. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III - Verification and Control*, number 1066 in Lecture Notes in Computer Science, pages 436–448. Springer-Verlag, 1996.

[27] M.L. Tyler and M. Morari. Propositional logic in control and monitoring problems. *Automatica*, 35(4):565–582, 1999.

[28] V. I. Utkin. Variable structure systems with sliding modes. *IEEE Trans. Automatic Control*, 22(2):212–222, April 1977.