Computer Languages, Systems & Structures 47 (2017) 44–61

Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

Automatic synthesis of *k*-inductive piecewise quadratic invariants for switched affine control programs ☆

Assalé Adjé, Pierre-Loïc Garoche *

Onera, The French Aerospace Lab, France and Université de Toulouse, Toulouse, France

ARTICLE INFO

Article history:
Received 22 May 2015
Received in revised form
28 November 2015
Accepted 4 December 2015
Available online 17 December 2015

Keywords:
Formal verification
Static analysis
Piecewise affine systems
Piecewise quadratic Lyapunov functions

ABSTRACT

Among the various critical systems that are worth to be formally analyzed, a wide set consists of controllers for dynamical systems. Those programs typically execute an infinite loop in which simple computations update internal states and produce commands to update the system state. Those systems are yet hardly analyzable by available static analysis method, since, even if performing mainly linear computations, the computation of a safe set of reachable states often requires quadratic invariants.

In this paper we consider the general setting of a piecewise affine program; that is a program performing different affine updates on the system depending on some conditions. This typically encompasses linear controllers with saturations or controllers with different behaviors and performances activated on some safety conditions.

Our analysis is inspired by works performed a decade ago by Johansson et al., and Morari et al., in the control community. We adapted their method focused on the analysis of stability in continuous-time or discrete-time settings to fit the static analysis paradigm and the computation of invariants, that is over-approximation of reachable sets using piecewise quadratic Lyapunov functions.

This approach has been further extended to consider *k*-inductive properties of reachable traces (trajectories) of systems.

The analysis has been implemented in Matlab and shown very good experimental results on a very large set of synthesized problems.

1. Introduction

With the success of Astrée [4], static analysis in general and abstract interpretation in particular are now seriously considered by industrials from the critical embedded system community, and more specifically by the engineers developing and validating controllers. The certification norms concerning the V&V of those software have also evolved and now enable the use of such methods in the development process.

These controller software are meant to perform an infinite loop in which values of sensors are read, a function of inputs and internal states is computed, and the value of the result is sent to actuators. In general, in the most critical applications, the controllers used are based on a simple linear update with minor non-linearities such as saturations, i.e. enforcing bounds, or specific behaviors when some conditions are met. The controlled systems range from aircraft flight commands,

☆ This work has been partially supported by an RTRA/STAE BRIEFCASE project grant, the ANR projects INS-2012-007 CAFEIN, and ASTRID VORACE.
* Corresponding author.
E-mail addresses: assale.adje@onera.fr (A. Adjé), pierre-loic.garoche@onera.fr (P.-L. Garoche).

http://dx.doi.org/10.1016/j.cl.2015.12.002
1477-8424/

guidance algorithms, engine control from any kind of device optimizing performance or fuel efficiency, control of railway infrastructure, fan control in tunnels, etc.

It is therefore of utmost importance to provide suitable analyses to verify these controllers. One of the approaches is to rely on quadratic invariants, such as the digital filters abstract domain of Feret [10], since, according to Lyapunov's theorem, any globally asymptotically stable linear system admits a quadratic Lyapunov function. Unfortunately, this theorem does not hold in the presence of disjunction, such as saturation. Moreover checking stability of piecewise systems is undecidable [25].

In static analysis, dealing with disjunction is an import concern. When the join of two abstract element is imprecise, one can consider the disjunctive completion of the domain [11]. This process enriches the set of abstract elements with new ones, but the cost, i.e. the number of new elements, could be exponential in the number of initial elements. Concerning relation abstract domains, one should mention the tropical polyhedra of Allamigeon [2] in which an abstract element characterizes a finite disjunction of zones [21]. However concerning quadratic properties, no static analysis actually performs the automatic computation of disjunctive quadratic invariants.

The goal of this paper is to propose such a computation: produce a disjunctive quadratic invariant as a sub-level of a piecewise quadratic Lyapunov function. Because of the undecidability of the problem, this search is heuristic, but shown to perform well in our experiments.

1.1. Related works

Most relational abstractions used in the static analysis community rely on a linear representation of relationship between variables, e.g. polyhedra [7], octagons [22], zonotopes [12] are not join-complete. Integrating constraints in invariants generation was developed in [9] but for computing linear invariants. As mentioned above, the tropical polyhedra domain [2] admits some disjunctions since it characterizes a family of properties encoded as finite disjunction of zones.

Concerning non-linear properties, the need for quadratic invariant was addressed a decade ago with ellipsoidal abstract domains for simple linear filters [10] and more recently for non-linear template domains [8] and policy iteration based static analysis [13].

More recently, techniques used in the control community have been used to synthesize appropriate quadratic templates using SDP solvers and Lyapunov functions [28].

The proposed technique addresses a family of systems well beyond the ones handled by the mentioned methods. In general, a global quadratic invariant is not enough to bound the reachable value of the considered systems, hence none of these could succeed.

On the control community side, Lyapunov based analysis is typically used to show the good behavior of a controlled system: it is globally asymptotically stable (GAS), i.e. when time goes to infinity the trajectories of the system goes to 0. Since about a decade SDP solvers, i.e. convex optimization algorithms for semi-definite programming, have reached a level of maturity that enable their use to compute quadratic Lyapunov functions. On the theory side, variants of quadratic Lyapunov functions such as the papers motivating our work – Johansson and Rantzer [27,15] as well as Mignone, Ferrari-Trecate and Morari [20] – addressed the study of piecewise linear systems for proving the GAS property.

Another related approach is the line of works supported by Lee and Dullerud [19,17,18] in which the problem is the ability to synthesize a stable controller for a piecewise system. Their approach relies on the computation of a piecewise quadratic Lyapunov for a subset of feasible transitions of the system, considering a bounded fixed number of switches between system behaviors.

In general, computing a safe superset of reachable states, as needed when performing static analysis, is not a common question for control theorist. They would rather address the related notions of controllability or stability under perturbations. In most cases, either the property considered or the technique used relies on the existence of such a bound over reachable state; which we aim to compute in static analysis.

1.2. Contributions

Our contribution is threefold and based on the method of Johansson and Mignone used to prove the GAS property of a piecewise linear system:

- we detailed the method in the discrete setting, computing a piecewise quadratic Lyapunov function of a *discrete-time system*;
- we adapted it to compute an invariant over reachable states of the analyzed system;
- we showed the applicability of the proposed method to a wide set of generated examples.

This paper is an extended version of [1] considering the expression of relationships between quadratic invariants along program traces as inspired by Lee and Dullerud. This approach proposed can be considered as a lift of previous method to k-induction [29,16].

1.3. Organization of the paper

The paper is structured as follows. Section 2 introduces the kind of programs considered. Section 3 introduces the notion of piecewise quadratic Lyapunov function. Section 4 presents the expression of conditions, such as guards in the program, as quadratic constraints. This is required to generate the constraints presented in Section 5, computing a quadratic Lyapunov function per behavior of the piecewise program. Section 6 develops the lift of the previous method to a k-induction setting: considering sequences of up to k transitions when searching for quadratic invariants. Last, Section 7 presents the experimentations while Section 8 concludes and opens future direction of research.

2. Problem statement

The programs we consider here are composed of a single loop with possibly a complicated switch-case type loop body. Our switch-case loop body is supposed to be written as a nested sequence of *ite* statements, or as a switch:

```
switch
  c1 → inst1;
  c2 → instr2;
  c3 → instr3;
  _  → instr4;
```

Moreover, we suppose that the analyzed programs are written in affine arithmetic. Consequently, the programs analyzed here can be interpreted as constrained piecewise affine discrete-time systems. Finally, we reduce the problem to compute automatically an overapproximation of the reachable states of a piecewise affine discrete-time system. The term piecewise affine means that there exists a polyhedral partition $\{X^i, i \in I\}$ of the state-input space $\mathcal{X} \subseteq \mathbb{R}^{d+m}$ such that for all $i \in I$, the dynamic of the system is affine and the system is represented by the following relation:

$$(x_0, u_0) \in X^0 \quad \text{and } \forall\, k \in \mathbb{N}, \quad x_{k+1} = A^i x_k + B^i u_k + b^i, \quad \text{if } (x_k, u_k) \in X^i, \tag{1}$$

where $X^0 \subseteq \mathbb{R}^{d+m}$ is the compact convex polyhedron of possible initial conditions, A^i is a $d \times d$ matrix, B^i a $d \times m$ matrix and b^i a vector of $\mathbb{R}^d$. The variable $x \in \mathbb{R}^d$ refers to the state variable and $u \in \mathbb{R}^m$ refers to some input variable.

For us, a polyhedral partition is a family of convex polyhedra which partitions the state-input space i.e. $\mathcal{X} = \bigcup_{i \in I} X^i \subseteq \mathbb{R}^{d+m}$ such that $X^i \cap X^j = \varnothing$ for all $i, j \in I$, $i \neq j$. From now on, we call X^i *cells*. Cells $\{X^i\}_{i \in I}$ are convex polyhedra which can contain both strict and weak inequalities. Cells can be represented by a $n_i \times (d+m)$ matrix T^i and c^i a vector of $\mathbb{R}^n_i$. We denote by $\mathbb{I}^s_i$ the set of indices which represent strict inequalities for the cell X^i, denote by T^i_s and c^i_s the parts of T^i and c^i corresponding to strict inequalities and by T^i_w and c^i_w the one corresponding to weak inequalities. Finally, we have the matrix representation given by the following formula:

$$X^i = \left\{ (x,u) \in \mathbb{R}^{d+m} \middle| T^i_s \begin{pmatrix} x \\ u \end{pmatrix} \lll c^i_s,\ T^i_w \begin{pmatrix} x \\ u \end{pmatrix} \leq c^i_w \right\} \tag{2}$$

We use the following notation: $y \lll z$ means that for all coordinates l, $y_l < z_l$ and $y \leq z$ means that for all coordinates l, $y_l \leq z_l$.

While the approach we propose can consider arbitrary partitioning of the system dynamics into cells, we infer automatically the cell's definition using the guards of the switch case constructs.

In order to simplify the following analysis, it is easier to consider a linear system rather than an affine one. Therefore we define an homogeneous flavor of the system dynamics: instead of considering a system state in $\mathbb{R}^d$ with inputs in $\mathbb{R}^m$, we manipulate system states in $\mathbb{R}^{1+d+m}$. Thus we introduce the $(1+d+m) \times (1+d+m)$ matrices F^i defined as follows:

$$F^i = \begin{pmatrix} 1 & 0_{1\times d} & 0_{1\times m} \\ b^i & A^i & B^i \\ 0 & 0_{m\times d} & \mathrm{Id}_{m\times m} \end{pmatrix} \tag{3}$$

The system defined in Eq. (1) can be rewritten as $(1, x^\top_{k+1}, u^\top_{k+1})^\top = F^i (1, x^\top_{k+1}, u^\top_k)^\top$. Note that we introduce a "virtual" dynamic law $u_{k+1} = u_k$ on the input variable in Eq. (3). In the point of view of set invariance computation, we will see that it remains to consider such dynamic law. Indeed we suppose that the input is bounded and we write $u_k \in \mathcal{U}$ for all $k \in \mathbb{N}$ with $\mathcal{U}$ being a nonempty compact convex polyhedra (convex polytope). We make the following assumption:

$$X^0 = \{x \in \mathbb{R}^d | \exists u \in \mathcal{U} \text{ s.t. } (x,u) \in X^0\} \times \mathcal{U}. \tag{4}$$

It means that X^0 is actually composed of initial conditions on the state variable x and the part of initial conditions on u is given by $\mathcal{U}$.

We are interested in proving that the reachable states $\mathcal{R}$ are bounded and a proof of this statement can be expressed by directly computing it. Recalling that $\mathcal{R}$ is the smallest set C satisfying $F(C) \subseteq C$ where F is the mapping on subsets of $\mathbb{R}^d$

defined for all $C \subset \mathbb{R}^d$ by:

$$F(C) = \{y \in \mathbb{R}^d | \exists i \in I,\ \exists (x,u) \in (C \times \mathcal{U}) \cap X^i \text{ s.t. } y = A^i x + B^i u + b^i\} \cup \{x \in \mathbb{R}^d | \exists u \in \mathcal{U} \text{ s.t. } (x,u) \in X^0\}$$

and prove that this set is bounded. We can also compute an overapproximation of $\mathcal{R}$ from a set $\mathcal{S} \subseteq \mathbb{R}^{d+m}$ such that $(x_0, u_0) \in \mathcal{S}$, $\mathcal{R} \times \mathcal{U} \subseteq \mathcal{S}$ and $\mathcal{S}$ is an inductive invariant in the sense of, for all $i \in I$:

$$(x,u) \in \mathcal{S} \cap X^i \Rightarrow (A^i x + B^i u + b^i, u) \in \mathcal{S}.$$

Indeed, by induction since X^0 is included in $\mathcal{S}$, then $(x_k, u_k) \in \mathcal{S}$ for all $k \in \mathbb{N}$. Since every image of the dynamic of the system stays in $\mathcal{S}$, a reachable state (y,u) belongs to $\mathcal{S}$. Finally, if we prove that $\mathcal{S}$ is bounded then $\mathcal{R}$ is also bounded.

Working directly on sets can be difficult and usually invariant sets are computed as a sublevel of some function to find. For (convergent) discrete-time linear systems, it is classical in the control community to compute ellipsoidal overapproximation of reachable states. Indeed, sublevel sets of Lyapunov functions are invariant sets for the analyzed linear system. Furthermore computing such an ellipsoid containing the initial states provides an overapproximation of reachable states. Initially, Lyapunov functions are used to prove quadratic asymptotic stability. In this paper, we use an analogue of Lyapunov functions for piecewise affine systems to compute directly an overapproximation of reachable states.

Example 2.1 (*Running example*). Let us consider the following program. It is constituted by a single while loop with two nested conditional branches in the loop body.

```
(x, y) ∈ [−9, 9] × [−9, 9];
while (true)
    ox=x;
    oy=y;
    read(u);  \\ u ∈ [−3, 3]
    if (−9∗ox+7∗y+6∗u<5){
        if(−4∗ox+8∗oy−8∗u<4){
            x=0.4217∗ox+0.1077∗oy+0.5661∗u;
            y=0.1162∗ox+0.2785∗oy+0.2235∗u−1;
            }
        else {  \\4∗ox−8∗oy+8∗u<−4
            x=0.4763∗ox+0.0145∗oy+0.9033∗u;
            y=0.1315∗ox+0.3291∗oy+0.1459∗u+9;
            }
        }
    else {  \\9∗ox−7∗y−6∗u<−5
        if(−4∗ox+8∗oy−8∗u<4){
            x=0.2618∗ox+0.1107∗oy+0.0868∗u−4;
            y=0.4014∗ox+0.4161∗oy+0.6320∗u+4;
            }
        else {  \\4∗ox−8∗oy+8∗u<−4
            x=0.3874∗ox+0.00771∗oy+0.5153∗u+10;
            y=0.2430∗ox+0.4028∗oy+0.4790∗u+7;
            }
        }
```

The initial condition of the piecewise affine systems is $(x,y) \in [-9,9] \times [-9,9]$ and the polytope where the input variable u lives is $\mathcal{U} = [-3,3]$.

We can rewrite this program as a piecewise affine discrete-time dynamical systems using our notations. We give details on the matrices T_s^i and T_w^i and vectors c_s^i and c_w^i (see Eq. (2)) which characterize the cells and on the matrices F^i representing the homogeneous version (see Eq. (3)) of affine laws in the cell X^i:

$$F^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4217 & 0.1077 & 0.5661 \\ -1 & 0.1162 & 0.2785 & 0.2235 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{cases} T_s^1 = \begin{pmatrix} -9 & 7 & 6 \\ -4 & 8 & -8 \end{pmatrix} \\ c_s^1 = (5\ \ 4)^\top \end{cases}, \quad \begin{cases} T_w^1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c_w^1 = (3\ \ 3)^\top \end{cases}$$

$$F^2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4763 & 0.0145 & 0.9033 \\ 9 & 0.1315 & 0.3291 & 0.1459 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{cases} T_s^2 = (-9\ \ 7\ \ 6) \\ c_s^2 = 5 \end{cases}, \quad \begin{cases} T_w^2 = \begin{pmatrix} 4 & -8 & 8 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c_w^2 = (-4\ \ 3\ \ 3)^\top \end{cases}$$

$$F^3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -4 & 0.2618 & 0.1177 & 0.0868 \\ 4 & 0.4014 & 0.4161 & 0.6320 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{cases} T_s^3 = (-4\ \ 8\ \ -8) \\ c_s^3 = 4 \end{cases}, \quad \begin{cases} T_w^3 = \begin{pmatrix} 9 & -7 & -6 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c_w^2 = (-5\ \ 3\ \ 3)^\top \end{cases}$$

$$F^4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 10 & 0.3874 & 0.0771 & 0.5153 \\ 7 & 0.2430 & 0.4028 & 0.4790 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \begin{cases} T_w^4 = \begin{pmatrix} 9 & -7 & -6 \\ 4 & -8 & 8 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c_w^4 = (-5\ \ -4\ \ 3\ \ 3)^\top \end{cases}$$

3. Invariant as sublevel set of Lyapunov functions

In [15,20], the authors proposed a method to prove stability of piecewise affine dynamical discrete-time systems. The method is a generalization of Lyapunov stability equations in the case where affine laws defining the system depend on the current state. Let A be a $d \times d$ matrix and let $x_{k+1} = Ax_k,\ k \in \mathbb{N},\ x_0 \in \mathbb{R}^d$ be a linear dynamical system.

Quadratic Lyapunov functions: We recall that V is a quadratic Lyapunov function iff there exists a $d \times d$ symmetric matrix P such that $V(x) = x^\top Px$ for all $x \in \mathbb{R}^d$ and $P \succ 0$ and $P - A^\top PA \succ 0$. The notation $P \succ 0$ means that P is positive definite i.e. $x^\top Px > 0$ for all $x \in \mathbb{R}^d,\ x \neq 0$ and 0 for $x = 0$. We will denote by $Q \succcurlyeq 0$ when Q is positive semidefinite i.e. $x^\top Px \geq 0$ for all $x \in \mathbb{R}^d$. Positive definite matrices characterize square of norm on $\mathbb{R}^d$.

A Lyapunov function allows us to prove the stability by the latter fact: the norm (associated to the Lyapunov function) of the states x_k decreases along the time. In switched system, similar to the classical case, we exhibited a positive definite matrix (square norm) to prove that the trajectories decrease along the time. The main difficulty in the switched case is the fact that we change the laws and we must decrease whenever a transition from one cell to other is fired. Moreover, we only require the norm to be local i.e. positive only where the law is used.

Therefore, our main goal is to synthesize a Lyapunov function $V(x,u)$ and an associated bound α characterizing the invariant of reachable states as a sublevel-set S_α, such that

$$\forall i \in I, \quad \forall (x,u) \in X^i, \quad V(x,u) \leq \alpha \tag{5}$$

$$\forall i,j \in I, \quad \forall (x,u) \in X^i, \quad \forall (x',u') \in X^j, \quad \text{s.t. } x' = A^i x + B^i u + b^i, \quad V(x,u) \geq V(x',u') \tag{6}$$

In Sections 5 and 6 we will develop different approaches to synthesize such V functions based on a piecewise characterization using quadratic Lyapunov functions. The next section focuses first in the expression of conditions.

4. Expressing conditions

In Eqs. (5) and (6), the inequalities on V are local on cells. In (6), the function has to decrease only on feasible transitions from cell X^i to cell X^j.

In the following we will synthesize piecewise Lyapunov function using SDP solvers. Encoding local constraints requires to be able to express the positivity of a quadratic form over a polyhedron as a semidefinite constraint.

This is performed through multiple transformations: first the implication is rewritten as a copositive constraint. This copositive constraint is further relaxed as quadratic computation with a matrix with nonnegative entries. This is the quadratization of constraints.

4.1. Quadratization of cells

We recall that for us local means that true on a cell and thus true on a polyhedron. Using the homogeneous version of a cell, we can define local positiveness on a polyhedral cone. Let Q be a $d \times d$ symmetric matrix and M be a $n \times d$ matrix. Local positivity in our case means that $My \geq 0 \Rightarrow y^\top Qy \geq 0$. The problem will be to write the local positivity as a constraint without implication. The problem is not new (e.g. the survey paper [14]). The paper [23] proves that local positivity is equivalent, when M has a full row rank, to $Q - M^\top CM \succcurlyeq 0$ where C is a copositive matrix i.e. $x^\top Cx \geq 0$ if $x \geq 0$. First in general (when the rank of M is not necessarily equal to its number of rows), note that if $Q - M^\top CM \succcurlyeq 0$ for some copositive matrix C then Q satisfies $My \geq 0 \Rightarrow y^\top Qy \geq 0$. Secondly every matrix C with nonnegative entries is copositive. Since copositivity seems to be as difficult as local positivity to handle, we will restrict copositive matrices to be matrices with nonnegative entries. The idea is instead of using cells as polyhedral cones, we use a quadratization of cells by introducing nonnegative entries and we will define the quadratization of a cell X^i by:

$$\overline{X^i} = \left\{ (x,u) \in \mathbb{R}^{d+m} \middle| \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top {E^i}^\top W^i E^i \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \geq 0 \right\} \tag{7}$$

where W^i is a $(1+n_i) \times (1+n_i)$ symmetric matrix with nonnegative entries and $E^i = \begin{pmatrix} E^i_s \\ E^i_w \end{pmatrix}$ with $E^i_s = \begin{pmatrix} 1 & 0_{1\times(d+m)} \\ c^i_s & -T^i_s \end{pmatrix}$ and $E^i_w = \begin{pmatrix} c^i_w & -T^i_w \end{pmatrix}$. Recall that n_i is the number of rows of T^i. The matrix E^i is thus of the size $n_i+1 \times (1+d+m)$. The goal of adding the row $(1, 0_{1\times(d+m)})$ is to avoid to add the opposite of a vector of X^i in $\overline{X^i}$. Indeed without this latter vector $\overline{X^i}$ would be symmetric. We illustrate this fact at Example 4.1. Note that during optimization process, matrices W^i will be decision variables.

Example 4.1 (*The reason of adding the row* $(1, 0_{1\times(d+m)})$). Let us take the polyhedra $X = \{x \in \mathbb{R} | x \leq 1\}$. Using our notations, we have $X = \{x | M(1\ x)^\top \geq 0\}$ with $M = (1\ \ -1)$. Let us consider two cases, the first one without adding the row and the second one using it.

Without any modification, the quadratization of X relative to a nonnegative real W is $X' = \{x | (1\ x)M^\top WM(1\ x)^\top \geq 0\}$. But $(1\ x)M^\top WM(1\ x)^\top = W(1\ x)(1\ \ -1)^\top(1\ \ -1)(1\ x)^\top = 2W(1-x)^2$. Hence $X' = \mathbb{R}$ for all nonnegative real W.

Now let us take $E = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}$. The quadratization as defined by Eq. (7) relative to a 2×2 symmetric matrix W with nonnegative coefficients is $\overline{X} = \{x | (1\ x)E^\top WE(1\ x)^\top \geq 0\}$. We have:

$$(1\ x)\begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} w_1 & w_3 \\ w_3 & w_2 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}(1\ x)^\top = w_1 + 2w_3(1-x) + w_2(1-x)^2.$$

To take a matrix W such that $w_2 = w_1 = 0$ and $w_3 > 0$ implies that $\overline{X} = X$.

Now we introduce an example of the quadratization of the cell X^1 for our running example.

Example 4.2. Let us consider the running example and the cell X^1. We recall that X^1 is characterized by the matrices and vectors:

$$\begin{cases} T^1_s = \begin{pmatrix} -9 & 7 & 6 \\ -4 & 8 & -8 \end{pmatrix}, \\ c^1_s = (5\ 4)^\top \end{cases} \quad \begin{cases} T^1_w = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c^1_w = (3\ 3)^\top \end{cases}$$

and

$$E^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 5 & 9 & -7 & -6 \\ 4 & 4 & -8 & 8 \\ 3 & 0 & 0 & -1 \\ 3 & 0 & 0 & 1 \end{pmatrix}$$

As suggested we have added the row $(1, 0_{1\times 3})$. Take for example the matrix:

$$W^1 = \begin{pmatrix} 63.0218 & 0.0163 & 0.0217 & 12.1557 & 8.8835 \\ 0.0163 & 0.0000 & 0.0000 & 0.0267 & 0.0031 \\ 0.0217 & 0.0000 & 0.0000 & 0.0094 & 0.0061 \\ 12.1557 & 0.0267 & 0.0094 & 4.2011 & 59.5733 \\ 8.8835 & 0.0031 & 0.0061 & 59.5733 & 3.0416 \end{pmatrix}$$

We have $\overline{X^1} = \{(x,y,u) | (1,x,y,u)E^1W^1E^1(1,x,y,u)^\top \geq 0\} \supseteq X^1$. In Section 7, we will come back on the generation of W^1.

Local positivity of quadratic forms will also be used when a transition from a cell to an other is fired. For the moment, we are interested in the set of (x,u) such that $(x,u) \in X^i$ and whose the image is in X^j and we denote by X^{ij} the set:

$$\left\{ (x,u) \in \mathbb{R}^{d+m} \middle| (x,u) \in X^i \quad \text{and} \quad (A^i x + B^i u + b^i, u) \in X^j \right\}$$

for all pairs $i,j \in I$. We will discuss in Section 4.2 the computation or a reduction to possible switches using linear programming as suggested in [5]. To construct a quadratization of X^{ij}, we use the same approach as before by introducing a $(1+n_i+n_j) \times (1+n_i+n_j)$ symmetric matrix U^{ij} with nonnegative entries to get a set $\overline{X^{ij}}$ defined as:

$$\overline{X^{ij}} = \left\{ (x,u) \in \mathbb{R}^{d+m} \middle| \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top E^{ij\top} U^{ij} E^{ij} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \geq 0 \right\} \tag{8}$$

where $E^{ij} = \begin{pmatrix} E^{ij}_s \\ E^{ij}_w \end{pmatrix}$ with

$$E^{ij}_s = \begin{pmatrix} 1 & 0_{1\times(d+m)} \\ c^i_s & -T^i_s \\ c^j_s - T^j_s \begin{pmatrix} b^i \\ 0 \end{pmatrix} & -T^j_s \begin{pmatrix} A^i & B^i \\ 0_{d\times m} & \mathrm{Id}_{m\times m} \end{pmatrix} \end{pmatrix}$$

and

$$E^{ij}_w = \begin{pmatrix} c^i_w & -T^i_w \\ c^j_w - T^j_w \begin{pmatrix} b^i \\ 0 \end{pmatrix} & -T^j_w \begin{pmatrix} A^i & B^i \\ 0_{d\times m} & \mathrm{Id}_{m\times m} \end{pmatrix} \end{pmatrix} \tag{9}$$

4.2. Switching cells

We have to manage another constraint which comes from the cell switches. After applying the available law in cell X^i, we have to specify the reachable cells i.e. the cells X^j such that there exists (x,u) satisfying:

$$(x,u) \in X^i \quad \text{and} \quad (A^i x + B^i u + b^i, u) \in X^j$$

We say that a switch from i to j is fireable iff:

$$\left\{ (x,u) \in \mathbb{R}^{d+m} \middle| \begin{array}{l} T^i_s (x,u)^\top \lll c^i_s \\ T^j_s (A^i x + B^i u + b^i, u)^\top \lll c^j_s \\ T^i_w (x,u)^\top \leq c^i_w \\ T^j_w (A^i x + B^i u + b^i, u)^\top \leq c^j_w \end{array} \right\} \neq \emptyset \tag{10}$$

We will denote by $i \to j$ if the switch from i to j is fireable and we denote by Sw the set of fireable switches i.e. $\mathrm{Sw} = \{(i,j) \in I^2 | i \to j\}$. Recall that the symbol $<$ means that we can deal with both strict inequalities and inequalities. To check whether $(i,j) \in \mathrm{Sw}$ is a linear programming feasibility problem with both strict and weak inequalities. However, we only check whether the system is solvable and we can detect infeasibility by using Motzkin transposition theorem [24]. Motzkin's theorem is an alternative type theorem: it considers two alternative linear systems such that exactly one of them is feasible. To describe the alternative system, we have to separate strict and weak inequalities and use the matrices E^{ij}_s and E^{ij}_w defined at Eq. (9). Then $(i,j) \in \mathrm{Sw}$ is equivalent to check whether the set $\{y \in \mathbb{R}^{1+d+m} | E^{ij}_w y \geq 0,\ E^{ij}_s y \ggg 0\}$ is empty or not. To detect

feasibility we test the infeasibility of the alternative system defined by:

$$\begin{cases} (E_s^{ij})^\top p^s + (E_w^{ij})^\top p^w = 0 \\ \sum\limits_{k \in \mathbb{I}_i^s} p_k^s = 1 \\ p_k^s \geq 0, \quad \forall k \in \mathbb{I}_i^s \\ p_k^w \geq 0, \quad \forall k \notin \mathbb{I}_i^s \end{cases} \tag{11}$$

From Motzkin's transposition theorem [24], we get the following proposition.

Proposition 4.1. *The pair* $(i,j) \in \mathrm{Sw}$ *iff Problem* (11) *is not feasible.*

However reasoning directly on the matrices can allow unfireable switches. For certain initial conditions, for all $k \in \mathbb{N}$, the condition $(x_k, u_k) \in X^i$ and $(A^i x_k + B^i u + b^i, u) \in X^j$ does not hold whereas $(i,j) \in \mathrm{Sw}$. To avoid it, we must know all the possible trajectories of the system (which we want to compute) and remove all inactivated switches. A sound way to under-approximate unfireable transitions is to identify unsatisfiable sets of linear constraints.

Example 4.3. We continue to detail our running example. More precisely, we consider the possible switches. We take for example the cell X^2. To switch from cell X^2 to cell X^1 is possible if the following system of linear inequalities has a solution:

$$\begin{aligned} &-9x+7y+6u < 5 \\ &-0.8532x+2.5748y-10.4460 < -68 \\ &-3.3662x+2.1732y-1.1084u < -58 \\ &4x-8y+8u \leq -4 \\ &u \leq 3 \\ &-u \leq 3 \end{aligned} \tag{12}$$

The two first consists in constraining the image of (x,y,u) to belong to X^1 and the four last constraints correspond to the definition of X^2. The representation of these two sets (X^2 and the preimage of X^1 by the law defined in X^2) is given at Fig. 1. We see in Fig. 1 that the system of inequalities defined at Eq. (12) seems to not have solutions. We check that using Eq. (11) and Proposition 4.1. The matrices E_s^{ij} and E_w^{ij} of Eq. (11) are in this example:

$$E_s^{21} = \begin{pmatrix} 5 & 9 & -7 & -6 \\ -68 & 0.8532 & -2.5748 & 10.446 \\ -58 & 3.3662 & -2.1732 & 1.1084 \end{pmatrix}$$

and

$$E_w^{21} = \begin{pmatrix} -4 & -4 & 8 & -8 \\ 3 & 0 & 0 & -1 \\ 3 & 0 & 0 & 1 \end{pmatrix}$$

We thus solve the linear program defined in Eq. (11) (with Matlab and Linprog) and we found that $p = (0.8735, 0.0983, 0.0282)^\top$ and $q = (0.3325, 14.2500, 7.8461)^\top$. This means that the alternative system is feasible and consequently the initial is not from Proposition 4.1. Finally the transition from X^2 to X^1 is not possible.

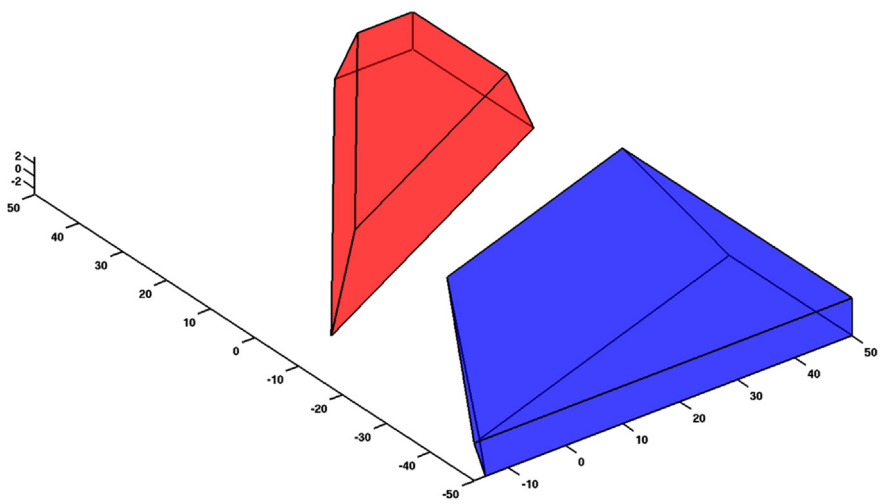

Fig. 1. The truncated representation of X^2 in red (on the left) and the preimage of X^1 by the law inside X^2 in blue (on the right).

5. Piecewise quadratic Lyapunov functions to compute invariant sets

Now we adapt the work of Rantzer and Johansson [15] and the work of Mignone et al. [20] to compute an invariant set for switched systems i.e. a subset $\mathcal{S}$ such that $(x_k,u)\in\mathcal{S}$ implies $(x_{k+1},u)\in\mathcal{S}$. These works are instead focused on deciding whether a piecewise affine system is global asymptotic convergent (GAS) or not. Even if the GAS problem is undecidable [3] the latter authors prove a stronger property on the system: there exists a piecewise Lyapunov functions for the piecewise affine systems. Rantzer and Johansson [15] and Mignone et al. [20] suggest to compute a piecewise quadratic function as Lyapunov function in the case of discrete-time piecewise affine systems to prove GAS property. Recall that a piecewise quadratic function on $\mathbb{R}^d$ is a function defined on a polyhedric partition of $\mathbb{R}^d$ which is quadratic on each polyhedron of the partition. In this paper, we propose to compute a (weaker) piecewise Lyapunov function to characterize an invariant set for our piecewise affine systems. In this section, we will denote by V this function. The pieces are given by the cells of the piecewise affine system and thus V is defined as:

$$\begin{aligned} V(x,u) &= V^i(x,u), \quad \text{if } (x,u)\in X^i \\ &= \begin{pmatrix} x \\ u \end{pmatrix}^\top P^i \begin{pmatrix} x \\ u \end{pmatrix} + 2{q^i}^\top \begin{pmatrix} x \\ u \end{pmatrix}, \quad \text{if } (x,u)\in X^i \end{aligned}$$

The function V^i is thus a local function only defined on X^i.

A sublevel set S_α of V of level $\alpha\in\mathbb{R}$ is represented as:

$$S_\alpha = \bigcup_{i\in I} S_{i,\alpha} = \bigcup_{i\in I}\left\{(x,u)\in X^i \middle| \begin{pmatrix} x \\ u \end{pmatrix}^\top P^i \begin{pmatrix} x \\ u \end{pmatrix} + 2{q^i}^\top x \le \alpha\right\} = \bigcup_{i\in I}\left\{(x,u)\in X^i \middle| \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top \begin{pmatrix} -\alpha & {q^i}^\top \\ q^i & P^i \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \le 0\right\}.$$

The set $S_{i,\alpha}$ is thus the local sublevel set of V^i associated to the level α.

So we are looking a family of pairs of a matrix and a vector $\{(P^i,q^i)\}_{i\in I}$ and a real $\alpha\in\mathbb{R}$ such that S_α is invariant by the piecewise affine system. To obtain invariance property, we have to constraint S_α to contain initial conditions of the system. Finally, to prove that the reachable set is bounded, we have to constraint S_α to be bounded.

Before deriving the semi-definite constraints, let us first state a useful result in Proposition 5.1. This result allows us to encode implications into semi-definite constraint in a safe way. The implication must involve quadratic inequalities on both sides.

Proposition 5.1. *Let A,B,C be $d\times d$ matrices. Then $C+A+B\succcurlyeq 0$ holds implies that the implication $(y^\top Ay\le 0\wedge y^\top By\le 0)\Rightarrow y^\top Cy\ge 0$ holds.*

Proof. Suppose that $C+A+B\succcurlyeq 0$. It is equivalent to say $y^\top(C+A+B)y\ge 0$ for all $y\in\mathbb{R}^d$. Now pick $z\in\mathbb{R}^d$ such that $z^\top Az\le 0$ and $z^\top Bz\le 0$. Since $z^\top Cz\ge -z^\top Az - z^\top Bz$, we conclude that $z^\top Cz\ge 0$ and the implication is true.□

5.1. Writing invariance as semi-definite constraints

We assume that $(x,u)\in X^i\cap S_{i,\alpha}$ (this index i is unique). Invariance means that if we apply the available law to (x,u) and suppose that the image of (x,u) belongs to some cell X^j (notation $(i,j)\in Sw$), then the image of (x,u) belongs to $S_{j,\alpha}$. Note that $(x,u)\in X^i$ and its image is supposed to be in X^j then $(x,u)\in X^{ij}$. Let $(i,j)\in$ Sw, invariance translated in inequalities and implication gives:

$$(x,u)\in X^{ij}\wedge(x,u)\in S_{i,\alpha}\Rightarrow(A^ix+B^iu+b^i,u)\in S_{j,\alpha} \tag{13}$$

We can use the relaxation of Section 4.1 as representation of cells and use matrix variables W^i and U^{ij} to encode their quadratization. We get:

$$\begin{aligned} &\begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top {E^{ij}}^\top U^{ij} E^{ij} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \ge 0 \wedge \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top \begin{pmatrix} -\alpha & {q^i}^\top \\ q^i & P^i \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \le 0 \\ &\Rightarrow \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top \left({F^i}^\top \begin{pmatrix} -\alpha & {q^j}^\top \\ q^j & P^j \end{pmatrix} F^i \right) \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \le 0 \end{aligned} \tag{14}$$

where E^{ij} is the matrix defined at Eq. (8) and F^i is defined at Eq. (3).

Finally, we obtain a stronger condition by considering semi-definite constraint such as the following equation:

$$-{F^i}^\top \begin{pmatrix} 0 & {q^j}^\top \\ q^j & P^j \end{pmatrix} F^i + \begin{pmatrix} 0 & {q^i}^\top \\ q^i & P^i \end{pmatrix} - {E^{ij}}^\top U^{ij} E^{ij} \succcurlyeq 0. \tag{15}$$

Note that the symbol $-\alpha$ is canceled during the computation.

Proposition 5.1 proves that if $(P^i, P^j, q^i, q^j, U^{ij})$ is a solution of Eq. (15) then $(P^i, P^j, q^i, q^j, U^{ij})$ satisfies Eq. (14).

5.2. Integrating initial conditions

Inductive invariants require both inductiveness and initial conditions. We address here the additional constraint for our computed invariant to contain initial states. Recall that possible initial conditions belong to some compact polyhedron $X^0 = \{(x,u) \in \mathbb{R}^{d+m} | T^0_w(x,u) \leq c^0_w,\ T^0_s(x,u) \ll c^0_s\}$. We must have $X^0 \subseteq S_\alpha$. Since $\{X^i, i \in I\}$ defines a partition of $\mathbb{R}^{d+m}$, X^0 is the union over $i \in I$ of the sets $X^0 \cap X^i$. If, for all $i \in I$, the set $X^0 \cap X^i$ is contained in $S_{i,\alpha}$ then $X^0 \subseteq S_\alpha$. Let us denote by Init the set of indices whose the cell meets X^0 i.e. $\text{Init} = \{i \in I | X^0 \cap X^i \neq \emptyset\}$. We can use the same method as before to express that for all $i \in \text{Init}$, $S_{i,\alpha}$ must contain $X^0 \cap X^i$. Now let us take $i \in \text{Init}$. In term of implications, $X^0 \cap X^i \subseteq S_{i,\alpha}$ can be rewritten as:

$$(x,u) \in X^0 \cap X^i \Rightarrow (x,u) P^i (x,u)^\top + 2(x,u) q^i \leq \alpha \tag{16}$$

Since $X^0 \cap X^i$ is a polyhedra, it admits some quadratization that is: $\overline{X^0 \cap X^i} = \{(x,u) \in \mathbb{R}^{d+m} | (1,x,u) {E^{0i}}^\top Z^i E^{0i} (1,x,u)^\top \geq 0\}$ where $E^{0i} = \begin{pmatrix} E^{0i}_s \\ E^{0i}_w \end{pmatrix}$ with:

$$E^{0i}_w = \begin{pmatrix} c^0_w & -T^0_w \\ c^i_w & -T^i_w \end{pmatrix} \quad \text{and} \quad E^{0i}_s = \begin{pmatrix} 1 & 0_{1\times(d+m)} \\ c^0_s & -T^0_s \\ c^i_s & -T^i_s \end{pmatrix}$$

and Z^i is some symmetric matrix whose coefficients are nonnegative.

Then, we obtain a stronger notion by introducing semi-definite constraints:

$$-\begin{pmatrix} -\alpha & {q^i}^\top \\ q^i & P^i \end{pmatrix} - {E^{0i}}^\top Z^i E^{0i} \succcurlyeq 0 \tag{17}$$

Proposition 5.1 proves that if (P^i, q^i, Z^i) is a solution of Eq. (17) then (P^i, q^i) satisfies Eq. (16).

Note since $X^0 \cap X^i$ is a polyhedron then its emptiness can be decided by checking the feasibility of the linear problem (18) and by using of the same argument than Proposition 4.1

$$\begin{cases} (E^{0i}_s)^\top p^s + (E^{0i}_w)^\top p^w = 0 \\ \sum_{k \in \mathbb{I}^s_i} p^s_k = 1 \\ p^s_k \geq 0, \quad \forall k \in \mathbb{I}^s_i \\ p^w_i \geq 0, \quad \forall k \notin \mathbb{I}^s_i \end{cases} \tag{18}$$

Linear program (18) is feasible iff $X^0 \cap X^i = \emptyset$ (notation $i \in \text{Init}$).

5.3. Writing boundedness as semi-definite constraints

The sublevel S_α is bounded if and only if for all $i \in I$, the sublevel $S_{i,\alpha}$ is bounded. The boundedness constraint in term of implications is, for all $i \in I$, there exists $\beta \geq 0$:

$$(x,u) \in X^i \wedge (x,u) \in S_{i,\alpha} \Rightarrow \|(x,u)\|_2^2 \leq \beta \tag{19}$$

where $\|\cdot\|_2$ denotes the Euclidian norm of $\mathbb{R}^{d+m}$.

As invariance, we use the quadratization of X^i and the definition of $S_{i,\alpha}$. We use the fact that $\|(x,u)\|_2^2 = \binom{x}{u}^\top \text{Id}_{(d+m)\times(d+m)} \binom{x}{u}$ and we get for all $i \in I$:

$$\begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top {E^i}^\top W^i E^i \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \geq 0 \quad \text{and} \quad \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top \begin{pmatrix} -\alpha & {q^i}^\top \\ q^i & P^i \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \leq 0$$

$$\Rightarrow \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\top \begin{pmatrix} -\beta & 0_{1\times(d+m)} \\ 0_{(d+m)\times 1} & \text{Id}_{(d+m)\times(d+m)} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \leq 0 \tag{20}$$

where E^i is defined in Eq. (7).

Finally, as invariance we obtain a stronger condition by considering semi-definite constraint such as Eq. (21). Proposition 5.1 proves that (P^i, q^i, W^i) is a solution of Eq. (21) then (P^i, q^i, W^i) satisfies Eq. (20). For all $i \in I$:

$$-{E^i}^\top W^i E^i + \begin{pmatrix} -\alpha & {q^i}^\top \\ q^i & P^i \end{pmatrix} + \begin{pmatrix} \beta & 0_{1\times(d+m)} \\ 0_{(d+m)\times 1} & -\mathrm{Id}_{(d+m)\times(d+m)} \end{pmatrix} \succcurlyeq 0 \tag{21}$$

5.4. Method to compute invariant set for piecewise affine systems and prove the boundedness of its reachable set

Algorithm 1 aims to prove the boundedness of the reachable values set of systems of the form (1).

Algorithm 1.

1. Define the real variables α, β;
2. For $i \in I$, compute the matrix E^i of Eq. (7); define the variable P^i as a symmetric matrix of size $(d+m) \times (d+m)$, the variable matrix W^i with nonnegative coefficients of size (♯ lines of E^i) × (♯lines of E^i) and add the constraint (21). If $i \in$ Init, define a the variable matrix Z^i with nonnegative coefficients of size (♯lines of E^{0i}) × (♯lines of E^{0i}) and add Constraint (17);
3. For all $(i,j) \in$ Sw, construct the matrix E^{ij} defined by Eq. (8) and define the symmetric matrix variable $U^{i,j}$ of the size (♯lines of E^{ij}) × (♯lines of E^{ij}) with nonnegative coefficients and add the constraint (15);
4. Add as linear objective function the sum of α and β to minimize;
5. Solve the semi-definite program.

Theorem 5.1. *Let $\mathcal{R}$ be the reachable values set of a given piecewise affine system of the form (1) and $\mathcal{U}$ be the bounded compact convex polyhedron where the control variable u lives. Assume that the SDP problem solved at Algorithm 1 has a solution $\left(\{P^i_{opt}, q^i_{opt}, i \in I\}, \{Z^i_{opt}, i \in In\}, \{U^{ij}_{opt}, (i,j) \in \mathrm{Sw}\}, \alpha_{opt}, \beta_{opt}\right)$. Then:*

1. *The set $S_{\alpha_{opt}} = \bigcup_{i\in I}\{(x,u) \in X^i | \begin{pmatrix} x \\ u \end{pmatrix}^\top P^i_{opt} \begin{pmatrix} x \\ u \end{pmatrix} + 2q^i_{opt}{}^\top \begin{pmatrix} x \\ u \end{pmatrix} \le \alpha_{opt}\}$ is bounded and $\mathcal{R}\times\mathcal{U} \subseteq S_{\alpha_{opt}}$.*
2. *For all $(x,u) \in \mathcal{R}\times\mathcal{U}$, $\|(x,u)\|_2^2 \le \beta_{opt}$.*

Proof (Sketch). For the first assertion, since from Assumption (4), $\mathcal{R}\times\mathcal{U} = \cup_{k\in\mathbb{N}} T^k(X^0)$ where $T: (x,u) \mapsto (A^i x + B^i u + b^i, u)$ if $(x,u) \in X^i$, we can use an induction. To prove the initialization of the induction, the property holds from the fact that if Eq. (17) holds then by Proposition 5.1, Eq. (16) holds. The induction holds from the fact that if Eq. (15) holds then by Proposition 5.1, Eq. (13). The second assertion follows readily from the fact if Eq. (21) holds then by Proposition 5.1, Eq. (19).□

5.5. Solution

The method is implemented in Matlab and the solution is given by a semi-definite programming solver in Matlab. For our running example, Matlab returns the following the values:

$$\alpha_{opt} = 242.0155$$
$$\beta_{opt} = 2173.8501$$

This means that $\|(x,y,u)\|_2^2 = x^2 + y^2 + u^2 \le \beta_{opt}$. We can conclude, for example, that the values taken by the variables x are between $[-46.6154, 46.6154]$. The value α_{opt} gives the level of the invariant sublevel of our piecewise quadratic Lyapunov function where the local quadratic functions are characterized by the following matrices and vectors:

$$P^1 = \begin{pmatrix} 1.0181 & -0.0040 & -1.1332 \\ -0.0040 & 1.0268 & -0.5340 \\ -1.1332 & -0.5340 & -13.7623 \end{pmatrix} \quad \text{and} \quad q^1 = (0.1252, 1.3836, -29.6791)^\top$$

$$P^2 = \begin{pmatrix} 9.1540 & -7.0159 & -2.6659 \\ -7.0159 & 9.5054 & -2.4016 \\ -2.6659 & -2.4016 & -8.9741 \end{pmatrix} \quad \text{and} \quad q^2 = (-21.3830, -44.6291, 114.2984)^\top$$

$$P^3 = \begin{pmatrix} 1.1555 & -0.3599 & -2.6224 \\ -0.3599 & 2.4558 & -2.8236 \\ -2.6224 & -2.8236 & -2.3852 \end{pmatrix} \quad \text{and} \quad q^3 = (-5.3138, 6.7894, -40.5537)^\top$$

$$P^4 = \begin{pmatrix} 3.7314 & -3.4179 & -3.1427 \\ -3.4179 & 6.1955 & 0.9499 \\ -3.1427 & 0.9499 & -10.6767 \end{pmatrix} \quad \text{and} \quad q^4 = (28.5011, -73.5421, 48.2153)^\top$$

Finally, for conciseness reason, we only give the matrix certificates for the cell X^1. First we give the matrix W^1 which encodes the quadratization of the guard X^1. Recall that this matrix ensures that $(x,y,u)\mapsto\alpha-(x,y,u)P^1(x,y,u)^\top-2(x,y,u)q^i$ is non-negative on X^1:

$$W^1=\begin{pmatrix}63.0218 & 0.0163 & 0.0217 & 12.1557 & 8.8835\\ 0.0163 & 0.0000 & 0.0000 & 0.0267 & 0.0031\\ 0.0217 & 0.0000 & 0.0000 & 0.0094 & 0.0061\\ 12.1557 & 0.0267 & 0.0094 & 4.2011 & 59.5733\\ 8.8835 & 0.0031 & 0.0061 & 59.5733 & 3.0416\end{pmatrix}$$

Secondly, we give the matrices U^{1j} which encodes the quadratization of polyhedron X^{1j}. Recall that those matrices ensure that the image of $(1,x,y,u)$ by F^1 belongs to the set $S_{j,\alpha}$ for all $(1,x,y,u)$ such that $F^1(1,x,y,u)\in X^j$:

$$U^{11}=\begin{pmatrix}0.0004 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0001\\ 0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000\\ 0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000\\ 0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000\\ 0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000\\ 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000\\ 0.0001 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0001\end{pmatrix}$$

$$U^{12}=\begin{pmatrix}2.1068 & 0.4134 & 0.0545 & 1.4664 & 0.1882 & 2.3955 & 2.4132\\ 0.4134 & 0.0008 & 0.0047 & 0.0009 & 0.0819 & 0.5474 & 0.0484\\ 0.0545 & 0.0047 & 0.0050 & 0.0147 & 0.0097 & 0.1442 & 0.2316\\ 1.4664 & 0.0009 & 0.0147 & 0.0041 & 0.3383 & 0.8776 & 0.0999\\ 0.1882 & 0.0819 & 0.0097 & 0.3383 & 0.0675 & 0.4405 & 0.4172\\ 2.3955 & 0.5474 & 0.1442 & 0.8776 & 0.4405 & 8.1215 & 9.6346\\ 2.4132 & 0.0484 & 0.2316 & 0.0999 & 0.4172 & 9.6346 & 0.9532\end{pmatrix}$$

$$U^{13}=\begin{pmatrix}0.3570 & 0.2243 & 0.0031 & 0.0050 & 0.1431 & 0.0388 & 0.7675\\ 0.2243 & 0.0201 & 0.0023 & 0.0050 & 0.1730 & 0.0494 & 0.1577\\ 0.0031 & 0.0023 & 0.0001 & 0.0001 & 0.0071 & 0.0006 & 0.0088\\ 0.0050 & 0.0050 & 0.0001 & 0.0002 & 0.3563 & 0.0009 & 0.0168\\ 0.1431 & 0.1730 & 0.0071 & 0.3563 & 0.0527 & 0.2689 & 0.8979\\ 0.0388 & 0.0494 & 0.0006 & 0.0009 & 0.2689 & 0.0137 & 0.1542\\ 0.7675 & 0.1577 & 0.0088 & 0.0168 & 0.8979 & 0.1542 & 0.2747\end{pmatrix}$$

$$U^{14}=\begin{pmatrix}1.3530 & 0.1912 & 0.0280 & 0.1178 & 2.9171 & 0.7079 & 1.4104\\ 0.1912 & 0.0512 & 0.0068 & 0.0326 & 1.7179 & 0.3764 & 0.6045\\ 0.0280 & 0.0068 & 0.0022 & 0.0048 & 0.1396 & 0.0264 & 0.0679\\ 0.1178 & 0.0326 & 0.0048 & 0.0409 & 0.5231 & 0.1204 & 0.2390\\ 2.9171 & 1.7179 & 0.1396 & 0.5231 & 15.0992 & 5.1148 & 14.3581\\ 0.7079 & 0.3764 & 0.0264 & 0.1204 & 5.1148 & 0.5102 & 1.6230\\ 1.4104 & 0.6045 & 0.0679 & 0.2390 & 14.3581 & 1.6230 & 1.2985\end{pmatrix}$$

We remark that U^{11} has negative coefficients whereas in our method, we are looking for a nonnegative coefficients matrix. It is due to the interior point method which is used to solve the semi-definite programming problems. Interior point methods return ϵ-optimal solution i.e. a solution which belongs to the ball of radius ϵ centered at an optimal solution. Hence, the solution furnished by the solver can slightly violate the constraints of the semi-definite program. We are aware of that and the projection of the returned solution on the feasible set should be studied as a future work.

6. *k*-Inductive piecewise quadratic Lyapunov functions

Another approach by Lee and Dullerud [17–19] uses a similar principle to check the stability of a piecewise dynamical system. Instead of assigning a quadratic function to each cell, they rather consider bounded paths in the graph of possible switches. Their setting is different: they do not intend to prove stability of an existing system or bound its reachable states but rather want to study the subset of possible switches that can make the system controllable. Their algorithm starts from paths of length 1 and, in case of failure, increments the considered path length.

We extend here this idea applied to our search of a piecewise quadratic invariant bounding reachable states of affine systems with switches of the form (1). Instead of considering only paths of length exactly k, we rather map this idea to the k-induction principle [29,16].

Definition 6.1 (k-*induction*). Let (Σ, In, T) be a transition system over states Σ with initial states $In \subseteq \Sigma$ and transition relation $T \subseteq \Sigma \times \Sigma$. A safety property $Prop \subseteq \Sigma$ is said k-inductive with respect to the transition system iff:

- for all system traces of length less than k, all reachable states verify *Prop*:

$$\forall j \leq k \in \mathbb{N}, \quad \forall x_0, \ldots, x_j \in \Sigma, x_0 \in In \wedge \bigwedge_{i \in [0,j-1]} (x_i, x_{i+1}) \in T \Rightarrow x_j \in Prop$$

- for all system subtraces of length k satisfying *Prop* then the next state satisfies *Prop* as well

$$\forall x_0, \ldots, x_k \in \Sigma, \quad \bigwedge_{i \in [0,k-1]} x_i \in Prop \wedge (x_i, x_{i+1}) \in T \Rightarrow x_k \in Prop$$

When proving a property by k-induction, one have to consider all the real transitions, i.e. actual traces of the system, starting from an initial state, up to k transitions. Then prove the inductive case, considering any prefix of length k. Since the systems we are considering are piecewise, it is possible to split to proof search into subcases and consider all transitions from one specific cell to another.

We recall that we consider a system of the form (1) composed of cells X^i indexed by a set I of partition labels. Then to apply the k-induction principle, we take $\Sigma = \mathbb{R}^{d+m} = \bigcup_{i \in I} X^i$, $\text{In} = X^0$ and we define the transition relation T as a piecewise transition relation $f = \{f^i, f^i \text{ defined on } X^i\}$ defined as follows: $(x, y) \in f^i$ iff $(1, y^\top)^\top = F^i(1, x^\top)^\top$ (where F^i is defined at Eq. (3)). The k-inductive property *Prop* denotes here a boundedness property represented by S'. Then, a k-induction proof amounts to find this set S that satisfies:

$$\begin{aligned} &\forall j < k \in \mathbb{N}, \quad \forall i_0, \ldots i_j \in I, \quad \forall x_0, \ldots, x_j \in \Sigma, \\ &\quad x_0 \in X^0 \wedge \bigwedge_{l \in [0,j-1]} x_l \in X^{i_l} \wedge (x_l, x_{l+1}) \in f^{i_l} \Rightarrow x_j \in S' \end{aligned} \tag{22}$$

$$\begin{aligned} &\forall i_0, \ldots i_k \in I, \quad \forall x_0, \ldots, x_k \in \Sigma, \\ &\quad \textstyle\bigwedge_{l \in [0,k-1]} x_l \in (X^{i_l} \cap S'_\alpha) \wedge (x_l, x_{l+1}) \in f^{i_l} \Rightarrow x_k \in S' \end{aligned} \tag{23}$$

Let I^* be the set of finite words of the letters in I, and I^*_k its restriction to words of length exactly k. In the following, we denote by $|w|$ the length of word w, by $a \cdot b$ the concatenation of the words a and b into ab and by $tl(w)$ the tail of a non-empty word w, i.e. w without its first letter. For example $tl(i \cdot w) = w$.

Following Lee and Dullerud approach, we reinforce Eqs. (22) and (23) to gather the states which share the same path w (sequence of switches) in a same set $S_{w,\alpha}$. To consider a finite number of possible paths we bound these paths by k i.e. w is a non-empty sequence of switches of length at most k. We obtain the new (stronger) system of implications:

$$\begin{aligned} &\forall w \in I^*_1, \quad \forall x \in \Sigma, \quad x \in X^0 \cap X^w \Rightarrow x \in S'_w \\ &\quad \forall 1 \leq j < k, \quad \forall w \cdot i \in I^*_j, \quad \forall x, y \in \Sigma, \quad \exists l \in I, \end{aligned} \tag{24}$$

$$\begin{aligned} &(x, y) \in f^i \wedge x \in S'_{w \cdot i} \wedge y \in X^l \Rightarrow y \in S'_{w \cdot i \cdot l} \\ &\quad \forall w \cdot i \in I^*_k, \quad \forall x, y \in \Sigma, \quad \exists l \in I, \end{aligned} \tag{25}$$

$$(x, y) \in f^i \wedge x \in S'_{w \cdot i} \wedge y \in X^l \Rightarrow y \in S'_{tl(w \cdot i) \cdot l} \tag{26}$$

Proposition 6.1. *Suppose that the system of Eqs.* (24)–(26) *has a solution* $\{S'_w | \forall 1 \leq j \leq k, w \in I^*_j\}$. *Let us define the set* S *that satisfies*:

$$S' = \bigcup_{i \in I} \bigcup_{\substack{w \in I^*_j \\ 0 \leq j \leq k-1}} S'_{w \cdot i}$$

Proof (Sketch). It suffices to prove that S' satisfies Eqs. (22) and (23). The first equation follows from $i_0 \cdot \ldots \cdot i_{k-1} = w$ is a word, $X^0 \cap X^{i_0} \subseteq S'_{i_0}$ (Eq. (24)) and by using k times Eq. (25), we conclude that Eq. (23) holds with $S'_{w \cdot i_k} \subseteq S'$. Now for Eq. (23), $x_0 \in S'$ then for some word w, $x_0 \in S'_{w \cdot i_0}$ using k times either Eq. (25) or Eq. (26) (depending on the path length), we conclude that $x_k \in S'$.□

To turn the sets equations in term of sublevel sets equations, we introduce the family of quadratic functions parameterized by words of length at most k. For $w \in I_j^*$ with $1 \leq j \leq k$ we define for all $(x,u) \in \mathbb{R}^{d+m}$:

$$V^w(x,u) = \begin{pmatrix} x \\ u \end{pmatrix}^\top P^w \begin{pmatrix} x \\ u \end{pmatrix} + 2q^w \begin{pmatrix} x \\ u \end{pmatrix},$$

for some $(d+m) \times (d+m)$ symmetric matrix P^w and some $d+m$-vector q^w. Then we can also define:

$$V'(x,u) = V'^i(x,u) \quad \text{if } (x,u) \in X^i \quad \text{where } V'^i = \min_{w' \in I_j^*,\ 0 \leq j \leq k-1} V^{w \cdot i}(x,u).$$

Finally, a sublevel solution of systems of equations consists in:

$$S'_\alpha = \left\{(x,u) \in \mathbb{R}^{d+m} | V'(x,u) \leq \alpha\right\} = \bigcup_{i \in I} \left\{(x,u) \in X^i | V'^i(x,u) \leq \alpha\right\}.$$

In the next, we will need the family of auxiliary ellipsoidal sets parameterized by words w of length at most k:

$$S'_{w,\alpha} = \left\{(x,u) \in \mathbb{R}^{d+m} | V^w(x,u) \leq \alpha\right\}.$$

By introducing quadratic functions, we will reinforce Eqs. (24), (25) and (26) from Proposition 5.1 as semidefinite constraints. In the next, we will adapt the semi-definite constraints of previous section to satisfy the k-inductive based constraints. While it is possible to target directly the synthesis of a k-inductive piecewise quadratic sublevel set, the approach typically starts from $k=1$ and increase to $k+1$ in case of failure to find a minimal k-inductive piecewise quadratic invariant.

6.1. Characterizing the graph of possible switches – enumerating the paths

As a first step, we compute the set of possible paths of given length up to k. First a graph $\mathcal{G} = (I, \text{Init}, \text{Sw})$ denoting possible switches between cells $i \in I$ is computed using the approach presented in Section 4.2.

Recall that Init denotes the subset of cells $i \in I$ that verify the initial conditions i.e. $\text{Init} = \{i \in I | X^0 \cap X^i \neq \emptyset\}$. In this graph context, these indices are used to label the vertices I. Recall also that Sw is the set of possible switches i.e. $\text{Sw} = \{(i,j) \in I^2 | \exists (x,u) \in X^i \text{ s.t. } (A^i x + B^i u + b^i, u) \in X^j\}$. In this graph context, it represents the edges. Recall that Init and Sw are computed using the method presented in Section 4.1.

We then enumerate the possible paths in the graph using classical graph algorithms. Let $Paths^k$ be such set of paths of length up to k.

Example 6.1. Fig. 2 presents the possible transitions as over-approximated by our method presented in Section 4.2. Depending on the target length the following paths are generated:

length	
1	1,2,3,4
2	11, 12, 13, 14, 22, 24, 31, 33, 34, 41, 42, 43, 44
3	111, 112, 113, 114, 122, 124, 131, 133, 134, 141, 142, 143, 144, 222, 224, 241, 242, 243, 244, 311, 312, 313, 314, 331, 333, 334, 341, 342, 343, 344, 411, 412, 413, 414, 422, 424, 431, 433, 434, 441, 442, 443, 444
4	…

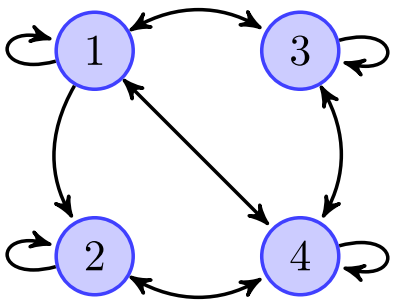

Fig. 2. Switch graph of the running example.

6.2. Integrating initial conditions

The initial condition only applies for the quadratic sublevel associated to initial cells. Let *Init* be the set of cells admitting initial elements, as defined in the graph construction.

By construction of the set of paths $Paths^k$, it contains the single letter words denoting initial cells $\{i | i \in Init\} \subseteq Paths^k$. The set of initial constraints only apply for these one letter word satisfying the initial condition:

$$(x,u) \in X^0 \cap X^i \Rightarrow (x,u)P^i(x,u)^\top + 2(x,u)q^i \leq \alpha \tag{27}$$

We can rely on the same stronger encoding as a semi-definite constraint, using the quadratization of the condition $X^0 \cap X^i$ as the matrix E^{0i}:

$$-\begin{pmatrix} -\alpha & q^{i\top} \\ q^i & P^i \end{pmatrix} - E^{0i\top} Z^i E^{0i} \succcurlyeq 0 \tag{28}$$

Note that, independently of the value of k, a system with n cells is parametrized by at most n Z^i variables.

6.3. Expressing transitions in initial and inductive cases as semi-definite constraints

Eqs. (25) and (26) denoting a transition X^{ij} from cell X^i to cell X^j can be defined as:

$$(x,u) \in X^{ij} \wedge (x,u) \in S_{w\cdot i,\alpha} \Rightarrow (A^i x + B^i u + b^i, u) \in S_{w\cdot i\cdot j,\alpha} \tag{29}$$

$$(x,u) \in X^{ij} \wedge (x,u) \in S_{w\cdot i,\alpha} \wedge |w\cdot i| = k \Rightarrow (A^i x + B^i u + b^i, u) \in S_{tl(w\cdot i)\cdot j,\alpha} \tag{30}$$

As before, these constraints are first relaxed with the use of quadratization of cell transitions E^{ij}, and then expressed as semi-definition constraints using Proposition 5.1.

when $|w\cdot i| = k$:

$$-F^{i\top}\begin{pmatrix} 0 & q^{tl(w\cdot i)\cdot j\top} \\ q^{tl(w\cdot i)\cdot j} & P^{tl(w\cdot i)\cdot j} \end{pmatrix} F^i + \begin{pmatrix} 0 & q^{w\cdot i\top} \\ q^{w\cdot i} & P^{w\cdot i} \end{pmatrix} - E^{ij\top} U^{w\cdot i,j} E^{ij} \succcurlyeq 0. \tag{31}$$

when $|w\cdot i| < k$:

$$-F^{i\top}\begin{pmatrix} 0 & q^{w\cdot i\cdot j\top} \\ q^{w\cdot i\cdot j} & P^{w\cdot i\cdot j} \end{pmatrix} F^i + \begin{pmatrix} 0 & q^{w\cdot i\top} \\ q^{w\cdot i} & P^{w\cdot i} \end{pmatrix} - E^{ij\top} U^{w\cdot i,j} E^{ij} \succcurlyeq 0. \tag{32}$$

Note that we have $|Paths^k|$ variables q^w, P^w and $|Paths^k| \times |I|$ variables $U^{w,j}$.

6.4. Expressing boundedness

The boundedness constraint expressed as a semi-definite constraint is straightforward. We require that all path-associated quadratic sublevel is bounded by the same scalar β.

For all $w\cdot i \in Paths^k$, there exists $\beta \geq 0$:

$$(x,u) \in X^i \wedge (x,u) \in S_{w\cdot i,\alpha} \Rightarrow \|(x,u)\|_2^2 \leq \beta \tag{33}$$

The associated semi-definite constraints is:

$$-E^{i\top} W^{w\cdot i} E^i + \begin{pmatrix} -\alpha & q^{w\cdot i\top} \\ q^{w\cdot i} & P^{w\cdot i} \end{pmatrix} + \begin{pmatrix} \beta & 0_{1\times(d+m)} \\ 0_{(d+m)\times 1} & -\mathrm{Id}_{(d+m)\times(d+m)} \end{pmatrix} \succcurlyeq 0 \tag{34}$$

We have here $|Paths^k|$ variables W^w.

6.5. Algorithm

The invariant computation is performed iteratively, increasing the length of the paths considered, using similar steps as the algorithm presented in Section 5.4. This new method is presented at Algorithm 2.

Algorithm 2.

1. Precompute unfeasible transitions and synthesize the matrix L of fireable switches
2. Start with a path of length $l=1$ and increase until timeout or success:
   (a) Generate from matrix L all feasible paths w of length up to l, including paths of length 1.
   (b) Define real variables α and β
   (c) For each path w, introduce variables P^w, q^w, W^w, and for each cell index $i \in I$, the variables $U^{w\cdot i}$ and Z^i.

(d) Depending on the system definition and the quadratization of guards, characterize the matrices $E^{0i}, E^{ij}, E^{i}, F^{i}$, for all $i, j \in I$.
(e) Solve the semi-definite program defined by the constraints (28), (31), (32), and (34) on such variables and constants.
(f) If there exist a solution, then return the solution $\{(P^w, q^w, W^w, U^{w\cdot i}, Z^i, \alpha_{opt}, \beta_{opt}), i \in I, w \in |Paths^l|\}$. Otherwise, increase the path length l and go to (a).

Theorem 6.1. *Let $\mathcal{R}$ be the reachable values set of a given piecewise affine system of the form (1) and $\mathcal{U}$ be the bounded compact convex polyhedron where the control variable u lives. Assume that Algorithm 2 stops at some $l \in \mathbb{N}^*$ with the solution $\{(P^w, q^w, W^w, U^{w\cdot i}, Z^i, \alpha_{opt}, \beta_{opt}), i \in I, w \in |Paths^l|\}$. Then:*

1. *The set:*

$$S'_{\alpha_{opt}} = \bigcup_{i \in I} \bigcup_{\substack{w \in I^* \\ 0 \le j \le l-1}} \left\{ (x,u) \in X^i \middle| \begin{pmatrix} x \\ u \end{pmatrix}^\top P^{w\cdot i}_{opt} \begin{pmatrix} x \\ u \end{pmatrix} + 2q^{w\cdot i}_{opt}{}^\top \begin{pmatrix} x \\ u \end{pmatrix} \le \alpha_{opt} \right\}$$

is bounded and $\mathcal{R} \times \mathcal{U} \subseteq S'_{\alpha_{opt}}$.

2. *For all $(x,u) \in \mathcal{R} \times \mathcal{U}$, $\|(x,u)\|_2^2 \le \beta_{opt}$.*

Proof (Sketch). Apply the same method of the proof of Theorem 5.1.□

6.6. Remark: special case of length 1

When one considers Eqs. (28), (31), (32), and (34) with the set of paths $Paths^1$ of length up to 1, we obtain exactly Eqs. (17), (15), and (21). In that case, Eq. (32) does not hold since no non-empty word of length strictly less than 1 exists.

6.7. Solution

The analysis of the running example with increased length generates the following results:

Length	$\beta(\sqrt{\beta})$	α	$\lvert Paths^k \rvert$
1	2173 (46.6154)	242.0155	4
2	2133 (46.1844)	233.0847	17
3	1652 (40.6448)	220.8596	73
4	1574 (39.6737)	228.5051	314

Note that the bound α on the piecewise quadratic sublevel applies on different sets of such local Lyapunov function. Their comparison is meaningless.

7. Experimentations

To illustrate the applicability of our method to a wide set of examples, we generated about a thousand (1030) of dynamical systems with at most 16 partition cells, 5 state variables and a single input.

In [3], the authors show (Theorem 2) that to determine the stability a piecewise affine dynamical system is undecidable. In order to generate more stable examples, we restricted the class of program generated. Each partition cell affine semantics

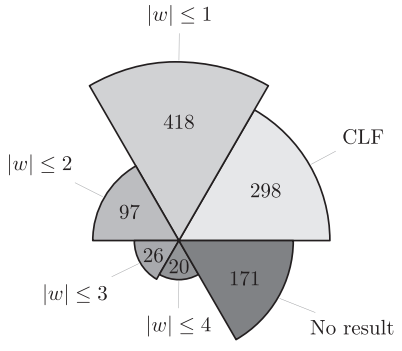

Fig. 3. Results of the analysis of benchmarks. A benchmark is associated to the category $|w| \le l$ iff it has not been bounded with a CLF or a smaller path length $l' < l$.

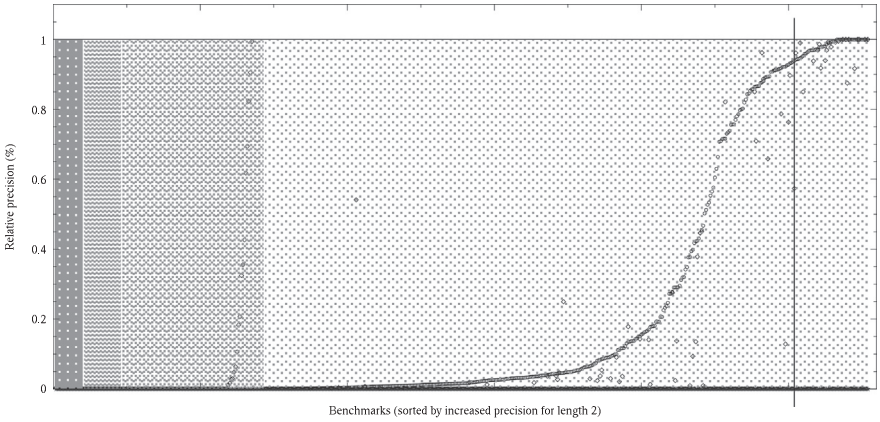

Fig. 4. *Relative precision*: this graph summarizes the relative precision obtained for successful analyses, i.e. excluding system proved with a common Lyapunov function nor the unproved ones. The line $y=1$ represents for each benchmark the normalized results, i.e. the bound on the value β obtained with the minimal succeeding path length. The background color shows the value of such length: e.g. all benchmarks at the right hand-side of the plot are proved bounded with paths of length 1, while the leftmost part required a path of length 4. Within each of such blocks, we sorted the benchmark by the increase in precision obtained with a longer path. For example the benchmark 504 (vertical line) obtained the bound $\beta=23{,}271$ with a path of length 1. This value becomes 21,853, that is 7% smaller, with a path of length 2, $\beta=13{,}343$ (43% smaller) with a path of length 3 and $\beta=888$ with a path of length 4. But in this last case the timing cost becomes prohibitive.

would be (i) generated with small coefficients, since big coefficients are usually avoided in controllers and, (ii) enforced locally stable when needed by updating the values of the coefficients using the spectral radius.

Our example synthesis still does not guarantee to obtain globally stable system, but, with these required properties of local stability and small coefficients, it is more likely that switching from one cell to the other would not break stability and therefore boundedness of the reachable states. The intuition behind is that when we pass from a cell to another cell, we multiply a vector by a small number then all the coordinates of the vector image are strictly smaller than the ones of initial vector.

As illustrated in Fig. 3 about 561 of such 1030 examples are automatically shown to be bounded using our technique while this class of program considered is unlikely to be analyzable with other static analysis tools the author are aware of, including the previous analyzes proposed [26]. For the sake of comparison we also evaluated the existence of a simpler Common Lyapunov function (CLF) which existed in 298 cases. A typical run of the analysis is about 20 s for a path of length 1, a minute for a length 2. In order to avoid long run of the analysis, we did not compute the piecewise invariants in a case of a system and a path length generating more than a thousand paths.

Fig. 4 analyzes the obtained results with respect to the relative precision and the path length.

All the computation have been performed within Matlab, including the synthesis of the examples. The source code of the analysis as well a document summarizing the examples and their analysis is available at https://cavale.enseeiht.fr/piecewisequadratic/.

8. Conclusion

The presented approach is able, considering a piecewise affine system, to compute a piecewise quadratic invariant able to bound the set of reachable state.

The technique extends the classical quadratic Lyapunov function synthesis using SDP solvers by formulating a more complex set of constraints to the SDP solver. This new formulation accounts the definition of the partitioning and encodes within the SDP constraints the relationship between partitions.

In practice our technique has been applied to a wide set of generated examples and was able to bound their reachable state space while a global quadratic invariant was proven not computable.

Our future work will consider the combination of this technique with other formal methods. A first direction will rely on the computed piecewise quadratic form as a template domain, bounding its value on some code using either Kleene iterations [6] or policy iteration [13]. This will require to extend the existing algorithms to fit this piecewise description of the template.

A second direction is to ease the applicability of the method and to integrate the technique in a more common analysis framework. A requirement for the presented work is to obtain a global representation of the program, as matrix updates and conditions. Existing static analysis [26] used for policy iteration extracts such a graph with the appropriate representation. We plan to integrate the two frameworks to ease the applicability on more realistic programs in an automated fashion.

Acknowledgments

We thank Mario Sigalotti for introducing us to Lee and Dullerud approach during a seminar organized by the working group GT Shy of the Labex DigiCosme in Paris. And we own deep gratitude to Xavier Thirioux for serious brainstorming about the *k*-induction proofs.

References

[1] Adjé A, Garoche P-L. Automatic synthesis of piecewise linear quadratic invariants for programs. In: D'Souza D, Lal A, Guldstrand Larsen K, editors. Verification, model checking, and abstract interpretation—16th international conference, VMCAI 2015, Mumbai, India, January 12–14, 2015. Proceedings, Lecture notes in computer science, vol. 8931. Springer; 2015. p. 99–116.
[2] Allamigeon X. Static analysis of memory manipulations by abstract interpretation—algorithmics of tropical polyhedra, and application to abstract interpretation [Ph.D. thesis]. École Polytechnique, Palaiseau, France; November 2009.
[3] Blondel V, Bournez O, Koiran P, Papadimitriou C, Tsitsiklis J. Deciding stability and mortality of piecewise affine dynamical systems. Theoret Comput Sci A 2001;1–2(255):687–96.
[4] Bertrane J, Cousot P, Cousot R, Feret J, Mauborgne L, Min A, et al. Static analysis by abstract interpretation of embedded critical software. In: ACM SIGSOFT Software Engineering Notes, vol. 36, no. 1; 2011. p. 1–8.
[5] Biswas P, Grieder P, Löfberg J, Morari M. A survey on stability analysis of discrete-time piecewise affine systems. In: IFAC world congress. Prague, Czech Republic; July 2005.
[6] Cousot P, Cousot R. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Conference record of the fourth annual ACM SIGPLAN-SIGACT symposium on principles of programming languages. New York, NY, Los Angeles, CA: ACM Press; 1977. p. 238–52.
[7] Cousot P, Halbwachs N. Automatic discovery of linear restraints among variables of a program. In: Aho A, Zilles S, Szymanski T, editors. POPL. ACM Press; 1978. p. 84–96.
[8] Colón M, Sankaranarayanan S. Generalizing the template polyhedral domain. In: Barthe G, editor. ESOP. Lecture notes in computer science, vol. 6602. Springer; 2011. p. 176–95.
[9] Colon M, Sankaranarayanan S, Sipma H. Linear invariant generation using non-linear constraint solving. In: Computer-aided verification (CAV), Lecture notes in computer science, vol. 2725. Springer-Verlag; 2003. p. 420–33.
[10] Feret J. Static analysis of digital filters. In: Schmidt D, editor. ESOP. Lecture notes in computer science, vol. 2986. Springer; 2004. p. 33–48.
[11] Filé G, Ranzato F. Improving abstract interpretations by systematic lifting to the powerset. In: Logic programming. Proceedings of the 1994 international symposium, Ithaca, NY, USA; November 13–17, 1994. p. 655–69.
[12] Ghorbal K, Goubault E, Putot S. The zonotope abstract domain taylor1+. In: Bouajjani A, Maler O, editors. CAV. Lecture notes in computer science, vol. 5643. Springer; 2009. p. 627–33.
[13] Gawlitza T, Seidl H, Adjé A, Gaubert S, Goubault E. Abstract interpretation meets convex optimization. J Symb Comput 2012;47(12):1416–46.
[14] Ikramov KhD, Savel'eva NV. Conditionally definite matrices. J Math Sci 2000;98(1):1–50.
[15] Johansson M. On modeling, analysis and design of piecewise linear control systems. In: Proceedings of the 2003 international symposium on circuits and systems, 2003. ISCAS '03, vol. 3; May 2003. p. III–646–III–649.
[16] Kahsai T, Tinelli c. Pkind: a parallel k-induction based model checker. In: Barnat J, Heljanko K, editors. Proceedings 10th international workshop on parallel and distributed methods in verification, PDMC 2011, Snowbird, UT, USA, vol. 72 of EPTCS; July 14, 2011, p. 55–62.
[17] Lee J-W, Dullerud GE. Uniformly stabilizing sets of switching sequences for switched linear systems. IEEE Trans Automat Contr 2007;52(5):868–74.
[18] Lee J-W, Dullerud GE. Joint synthesis of switching and feedback for linear systems in discrete time. In: Caccamo M, Frazzoli E, Grosu R, editors. Proceedings of the 14th ACM international conference on hybrid systems: computation and control, HSCC 2011, Chicago, IL, USA, April 12-14. ACM; 2011. p. 201–10.
[19] Lee J-W, Dullerud GE, Khargonekar PP. An output regulation problem for switched linear systems in discrete time. In: Proceedings of the 46th IEEE conference on decision and control; 2007. p. 4993–8.
[20] Mignone D, Ferrari-Trecate G, Morari M. Stability and stabilization of piecewise affine and hybrid systems: an lmi approach. In: Proceedings of the 39th IEEE conference on Decision and control, vol. 1; 2000. p. 504–9.
[21] Miné A. A new numerical abstract domain based on difference-bound matrices. In: Programs as data objects, Second symposium, PADO 2001, Aarhus, Denmark; May 21–23, 2001. p. 155–72.
[22] Miné A. The octagon abstract domain. Higher-Order Symbol Comput 2006;19(1):31–100.
[23] Martin DH, Jacobson DH. Copositive matrices and definiteness of quadratic forms subject to homogeneous linear inequality constraints. Linear Algebra Appl 1981;35(0):227–58.
[24] Motzkin TS. Two consequences of the transposition theorem on linear inequalities. Econometrica 1951;19(2):184–5.
[25] Prabhakar Pavithra, Viswanathan Mahesh. On the decidability of stability of hybrid systems. In: Belta Calin, Ivancic Franjo, editors. Proceedings of the 16th international conference on hybrid systems: computation and control, HSCC 2013, April 8–11, 2013, Philadelphia, PA, USA. ACM; 2013. p. 53–62.
[26] Roux P, Garoche P-L. Integrating policy iterations in abstract interpreters. In: Van Hung D, Ogawa M, editors. ATVA. Lecture notes in computer science, vol. 8172. Springer; 2013. p. 240–54.
[27] Rantzer A, Johansson M. Piecewise linear quadratic optimal control. IEEE Trans. Automat. Control 2000;45(April (4)):629–37.
[28] Roux P, Jobredeaux R, Garoche P-L, Feron E. A generic ellipsoid abstract domain for linear time invariant systems. In: Dang T, Mitchell I, editors. HSCC. ACM; 2012. p. 105–14.
[29] Sheeran M, Singh S, Stålmarck G. Checking safety properties using induction and a sat-solver. In: Hunt Jr WA, Johnson SD, editors. Formal methods in computer-aided design, third international conference, FMCAD 2000, Austin, TX, USA, November 1–3, 2000, Proceedings. Lecture notes in computer science, vol. 1954. Springer; 2000. p. 108–25.