Automatic Synthesis of Piecewise Linear Quadratic Invariants for Programs*

Assalé Adjé and Pierre-Loïc Garoche

Onera, the French Aerospace Lab, France
Université de Toulouse, Toulouse, France
{firstname.lastname}@onera.fr

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.

Keywords: formal verification, static analysis, piecewise affine systems, piecewise quadratic lyapunov functions.

1 Introduction

With the success of Astrée [BCC+11], 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 validation 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,

* This work has been partially supported by an RTRA/STAE BRIEFCASE project grant, the ANR projects INS-2012-007 CAFEIN, and ASTRID VORACE.

D. D'Souza et al. (Eds.): VMCAI 2015, LNCS 8931, pp. 99–116, 2015.

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, 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 outmost importance to provide suitable analyses to verify these controllers. One of the approach is to rely on quadratic invariants, such as the digital filters abstract domain of Feret [Fer04], since, according to Lyapunov theorem, any globally asymptotically stable linear system admits a quadratic Lyapunov function. This theorem does not hold in presence of disjunction, such as saturations.

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 [FR94]. 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 [All09] in which an abstract element characterizes a finite disjunction of zones [Min01]. 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.

Related works. Most relational abstractions used in the static analysis community rely on a linear representation of relationship between variables, e.g. polyhedra [CH78], octagons [Min06], zonotopes [GGP09] are not join-complete. Integrating constraints in invariants generation was developed in [CSS03] but for computing linear invariants. As mentioned above, the tropical polyhedra domain [All09] 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 [Fer04] and more recently for non linear template domains [CS11] and policy iteration based static analysis [GSA+12].

More recently, techniques used in the control community have been used to synthesize appropriate quadratic templates using SDP solvers and Lyapunov functions [RJGF12].

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 are 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 [RJ00, Joh03] as well as Mignone, Ferrari-Trecate and Morari [MFTM00] – addressed the study of piecewise linear systems for proving the GAS property.

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 controllabilty or stability under perturbations. In most case, either the property considered, or the technique used, relies on the existence of a such a bound over reachable state; which we aim to compute in static analysis.

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.

Organization of the paper. The paper is structured as follow. Section 2 introduces the kind of programs considered. Section 3 details our version of the piecewise quadratic Lyapunov function as well as the characterization of invariant sets. Section 4 presents the experimentations while Section 5 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 a *switch* $c1 \rightarrow inst1; c2 \rightarrow instr2; c3 \rightarrow instr3$. Moreover, we suppose that the analyzed programs are written in affine arithmetic. Consequently, the programs analyzed here can be interpreted as 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 represented by the following relation for all $k \in \mathbb{N}$:

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

where 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 $X^i \cap X^j = \emptyset$ 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 $\mathbb{I}_i^s$ the set of indices which represent strict inequalities for the cell X^i, denote by T_s^i and c_s^i the parts of T^i and c^i corresponding to strict inequalities and by T_w^i and c_w^i the one corresponding to weak inequalities. Finally, we have the matrix representation given by Formula (2).

$$X^i = \left\{ \begin{pmatrix} x \\ u \end{pmatrix} \in \mathbb{R}^{d+m} \,\middle|\, T_s^i \begin{pmatrix} x \\ u \end{pmatrix} \ll c_s^i,\ T_w^i \begin{pmatrix} x \\ u \end{pmatrix} \leq c_w^i \right\} \tag{2}$$

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

We will need homogeneous versions of laws and thus 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 Equation (1) can be rewritten as $(1, x_{k+1}, u_{k+1})^\intercal = F^i(1, x_{k+1}, u_k)$. Note that we introduce a "virtual" dynamic law $u_{k+1} = u_k$ on the input variable in Equation (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 $\mathcal{U}$ is a nonempty compact set (polytope).

We are interested in proving that the reachable states $\mathcal{R}$ is bounded and a proof of this statement can be expressed by directly computing $\mathcal{R}$ that is:

$$\mathcal{R} = \{y \in \mathbb{R}^d \mid \exists\ k \in \mathbb{N},\ \exists\ i \in I,\ \exists (x_k, u_k) \in X^i,\ y = A^i x_k + B^i u_k + b^i\} \cup \{x_0\}$$

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 $i \in I$:

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

Indeed, by induction since (x_0, u_0) belongs to $\mathcal{S}$, $(x_k, u_k) \in \mathcal{S}$ 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 to compute ellipsoidal overapproximation of reachable states. Indeed, sublevel sets of Lyapunov functions are invariant set for the

analyzed linear system and to compute 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 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 Equation (2)) which characterize the cells and on the matrices F^i representing the homogeneous version (see Equation (3)) of affine laws in the cell X^i.

$$F^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4217 & 0.1077 & 0.5661 \\ -1 & 0.1162 & 0.2785 & 0.2235 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \begin{cases} T_s^1 = \begin{pmatrix} -9 & 7 & 6 \\ -4 & 8 & -8 \end{pmatrix} \\ c_s^1 = (5\ 4)^\intercal \end{cases}, \quad \begin{cases} T_w^1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c_w^1 = (3\ 3)^\intercal \end{cases}$$

$$F^2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4763 & 0.0145 & 0.9033 \\ 9 & 0.1315 & 0.3291 & 0.1459 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{cases} T_s^2 = \begin{pmatrix} -9 & 7 & 6 \end{pmatrix} \\ c_s^2 = 5 \end{cases}, \quad \begin{cases} T_w^2 = \begin{pmatrix} 4 & -8 & 8 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c_w^2 = (-4\ 3\ 3)^\intercal \end{cases}$$

$$F^3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -4 & 0.2618 & 0.1177 & 0.0868 \\ 4 & 0.4014 & 0.4161 & 0.6320 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{cases} T_s^3 = \begin{pmatrix} -4 & 8 & -8 \end{pmatrix} \\ c_s^3 = 4 \end{cases}, \quad \begin{cases} T_w^3 = \begin{pmatrix} 9 & -7 & -6 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \\ c_w^2 = (-5\ 3\ 3)^\intercal \end{cases}$$

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

3 Invariant Computation

In [Joh03, MFTM00], the authors propose 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. We recall that L is a quadratic Lyapunov function iff there exists a $d \times d$ symmetric matrix P such that $L(x) = x^\intercal Px$ for all $x \in \mathbb{R}^d$ and $P \succ 0$ and $P - A^\intercal PA \succ 0$. The notation $P \succ 0$ means that P is positive definite i.e. $x^\intercal Px > 0$ for all $x \in \mathbb{R}^d,\ x \neq 0$ and 0 for $x = 0$. We will denote by $Q \succeq 0$ when Q is positive semidefinite i.e. $x^\intercal Px \geq 0$ for all $x \in \mathbb{R}^d$. Positive definite matrices characterize square of norm on $\mathbb{R}^d$. A Lyapunov function allows 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, similarly to the classical case, we exhibited a positive 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.

3.1 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 $My \geq 0 \implies y^\intercal Qy \geq 0$. The problem will be to write the local positivity as a constraint without implication. The problem is not new (e.g. the survey paper [IS00]). The paper [MJ81] proves that local positivity is equivalent, when M has a full row rank, to $Q - M^\intercal CM \succeq 0$ where C is a copositive matrix i.e. $x^\intercal Cx \geq 0$ if $x \geq 0$. First in general (when the rank of M is not necessarily equal to its number of rows), note that if $Q - M^\intercal CM \succeq 0$ for some copositive matrix C then Q satisfies $My \geq 0 \implies y^\intercal Qy \geq 0$. 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 which 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:

$$\overline{X^i} = \left\{ \begin{pmatrix} x \\ u \end{pmatrix} \in \mathbb{R}^{d+m} \;\middle|\; \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal E^{i\intercal} W^i E^i \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \geq 0 \right\} \tag{4}$$

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 2. Note that during optimization process, matrices W^i will be decision variables.

Example 2 (The reason of adding the row $(1, 0_{1\times(d+m)})$). Let us take the polyhedra $X = \{x \in \mathbb{R} \mid x \leq 1\}$. Using our notations, we have $X = \{x \mid M(1\ x)^\intercal \geq 0\}$ with $M = \begin{pmatrix} 1 & -1 \end{pmatrix}$. 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 $X' = \{x \mid (1\ x)M^\intercal WM(1\ x)^\intercal \geq 0\}$. But $(1\ x)M^\intercal WM(1\ x)^\intercal = W(1\ x)(1\ -1)^\intercal(1\ -1)(1\ x)^\intercal = 2W(1-x)^2$. 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 Equation (4) relative to a 2×2 symmetric matrix W with nonnegative coefficients is $\overline{X} = \{x \mid (1\ x)E^\intercal WE(1\ x)^\intercal \geq 0\}$. We have:

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

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

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

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 $\overline{X^1} = \{(x, y, u) \mid (1, x, y, u)E^1W^1E^1(1, x, y, u)^\intercal \geq 0\} \supseteq X^1$. In Section 4, 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:

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

for all pairs $i, j \in I$. Note that in [MFTM00], the authors take into account all pairs (i, j) such that there exists a state x_k at moment k in X^i and the image of x_k that is x_{k+1} is in X^j. We will discuss in Subsection 3.2 the computation or a reduction to possible switches using linear programming as suggested in [BGLM05]. To construct a quadratization of X^{ij}, we use the same approach than 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:

$$\overline{X^{ij}} = \left\{ \begin{pmatrix} x \\ u \end{pmatrix} \in \mathbb{R}^{d+m} \middle| \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal E^{ij\intercal} U^{ij} E^{ij} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \geq 0 \right\} \tag{5}$$

where $E^{ij} = \begin{pmatrix} E_s^{ij} \\ E_w^{ij} \end{pmatrix}$ with

$$E_s^{ij} = \begin{pmatrix} 1 & 0_{1\times(d+m)} \\ c_s^i & -T_s^i \\ c_s^j - T_s^j \begin{pmatrix} b^i \\ 0 \end{pmatrix} & -T_s^j \begin{pmatrix} A^i & B^i \\ 0_{d\times m} & \mathrm{Id}_{m\times m} \end{pmatrix} \end{pmatrix}$$

and

$$E_w^{ij} = \begin{pmatrix} c_w^i & -T_w^i \\ c_w^j - T_w^j \begin{pmatrix} b^i \\ 0 \end{pmatrix} & -T_w^j \begin{pmatrix} A^i & B^i \\ 0_{d\times m} & \mathrm{Id}_{m\times m} \end{pmatrix} \end{pmatrix} \tag{6}$$

3.2 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:

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

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

$$\left\{ (x,u) \in \mathbb{R}^{d+m} \left| \begin{array}{l} T_s^i (x,u)^\intercal \ll c_s^i \\ T_s^j (A^i x + B^i u + b^i, u)^\intercal \ll c_s^j \\ T_w^i (x,u)^\intercal \leq c_w^i \\ T_w^j (A^i x + B^i u + b^i, u)^\intercal \leq c_w^j \end{array} \right. \right\} \neq \emptyset \tag{7}$$

We will denote by $i \to j$ if the switch from i to j is fireable. Recall that the symbol $<$ means that we can deal with both strict inequalities and inequalities. Problem (7) 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 [Mot51]. Motkin's theorem is an alternative type theorem, that is we oppose two linear systems such that exactly one of the two is feasible. To describe the alternative system, we have to separate strict and weak inequalities and use the matrices E_s^{ij} and E_w^{ij} defined at Equation (6). Problem (7) is equivalent to check whether the set $\{y = (z, x, u) \in \mathbb{R}^{1+d+m} \mid E_w^{ij} y \geq 0,\ E_s^{ij} y \gg 0\}$ is empty or not. To detect feasibility we test the infeasibility of the alternative system defined as:

$$\begin{cases} (E_s^{ij})^\intercal p^s + (E_w^{ij})^\intercal p = 0 \\ \sum_{k \in \mathbb{I}} p_k^s = 1 \\ p_k^s \geq 0,\ \forall\, k \in \mathbb{I} \\ p_i \geq 0,\ \forall\, i \notin \mathbb{I} \end{cases} \tag{8}$$

From Motzkin's transposition theorem [Mot51], we get the following proposition.

Proposition 1. *Problem* (7) *is feasible iff Problem* (8) *is not.*

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 Problem (7) is feasible. 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 underapproximate unfireable transitions is to identify unsatisfiable sets of linear constraints.

Example 4. 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:

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

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 Figure 1. We see at Figure 1 that the system of inequalities defined at

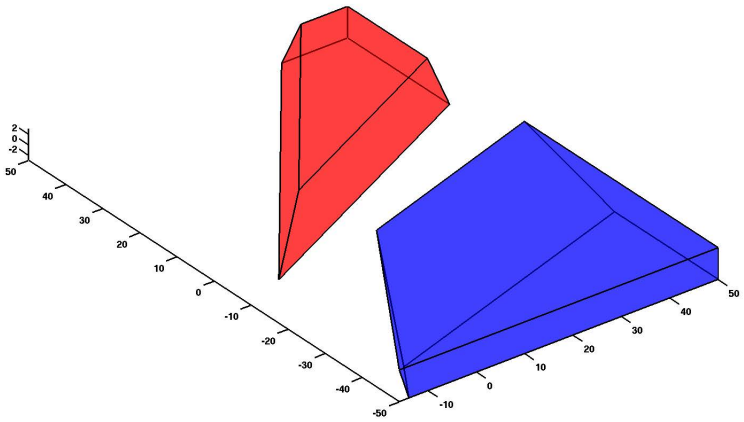

Fig. 1. The truncated representation of X^2 in red and the preimage of X^1 by the law inside X^2 in blue

Equation (9) seems to not have solutions. We check that using Equation (8) and Proposition 1. The matrices E_s^{ij} and E_w^{ij} of Equation (8) are in this example:

$$E_s^{21} = \begin{pmatrix} 5 & 9 & -7 & -6 \\ -68 & 0.8532 & -2.5748 & 10.446 \\ -58 & 3.3662 & -2.1732 & 1.1084 \end{pmatrix} \text{ and } E_w^{21} = \begin{pmatrix} -4 & -4 & 8 & -8 \\ 3 & 0 & 0 & -1 \\ 3 & 0 & 0 & 1 \end{pmatrix}$$

We thus solve the linear program defined in Equation (8) (with Matlab and Linprog) and we found $p = (0.8735, 0.0983, 0.0282)^\intercal$ and $q = (0.3325, 14.2500, 7.8461)^\intercal$. This means that the alternative system is feasible and consequently the initial is not from Proposition 1. Finally the transition from X^2 to X^1 is not possible.

3.3 Piecewise Quadratic Lyapunov Functions to Compute Invariant Sets

Now we adapt the work of Rantzer and Johansson [Joh03] and the work of Mignone et al [MFTM00] to compute of 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 or not. Even if the problem is undecidable [BBK$^+$01] the latter authors prove a stronger property on the system: there exists a piecewise Lyapunov functions for the piecewise affine systems. Rantzer and Johansson [Joh03] and Mignone et al [MFTM00] 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:

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

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:

$$\begin{aligned} S_\alpha &= \bigcup_{i\in I} S_{i,\alpha} \\ &= \bigcup_{i\in I} \left\{ \begin{pmatrix} x \\ u \end{pmatrix} \in X^i \mid \begin{pmatrix} x \\ u \end{pmatrix}^\intercal P^i \begin{pmatrix} x \\ u \end{pmatrix} + 2q^{i\intercal} x \leq \alpha \right\} \\ &= \bigcup_{i\in I} \left\{ \begin{pmatrix} x \\ u \end{pmatrix} \in X^i \mid \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal \begin{pmatrix} -\alpha & q^{i\intercal} \\ q^i & P^i \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \leq 0 \right\}. \end{aligned}$$

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 2. This result allows to encode implications into semi-definite constraint in a safe way safe. The implication must involve quadratic inequalities on both sides.

Proposition 2. *Let A, B, C be $d \times d$ matrices. Then $C + A + B \succeq 0$ holds implies that the implication $(y^\intercal A y \leq 0 \wedge\ y^\intercal B y \leq 0) \implies y^\intercal C y \geq 0$ holds.*

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

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 $i \to j$), 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 $(i, j) \in I^2$ such that $i \to j$, invariance translated in inequatilities and implication gives :

$$\begin{pmatrix} x \\ u \end{pmatrix} \in X^{ij} \wedge \begin{pmatrix} x \\ u \end{pmatrix} \in S_{i,\alpha} \implies \begin{pmatrix} A^i x + B^i u + b^i \\ u \end{pmatrix} \in S_{j,\alpha} \tag{10}$$

We can use the relaxation of Subsection 3.1 as representation of cells and use matrix variables W^i and U^{ij} to encode their quadratization. We get, for $(i, j) \in I^2$ such that $i \to j$:

$$\begin{aligned} &\begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal E^{ij\intercal} U^{ij} E^{ij} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \geq 0 \wedge \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal \begin{pmatrix} -\alpha & q^{i\intercal} \\ q^i & P^i \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \leq 0 \\ \implies &\begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal \left(F^{i\intercal} \begin{pmatrix} -\alpha & q^{j\intercal} \\ q^j & P^j \end{pmatrix} F^i \right) \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \leq 0 \end{aligned} \tag{11}$$

where E^{ij} is the matrix defined at Equation (5) and F^i is defined at Equation (3).

Finally, we obtain a stronger condition by considering semi-definite constraint such as Equation (12). Proposition 2 proves that if $(P^i, P^j, q^i, q^j, U^{ij})$ is a solution of Equation (12) then $(P^i, P^j, q^i, q^j, U^{ij})$ satisfies Equation (11). For $(i, j) \in I^2$ such that $i \to j$:

$$-F^{i\intercal} \begin{pmatrix} 0 & q^{j\intercal} \\ q^j & P^j \end{pmatrix} F^i + \begin{pmatrix} 0 & q^{i\intercal} \\ q^i & P^i \end{pmatrix} - E^{ij\intercal} U^{ij} E^{ij} \succeq 0 \ . \tag{12}$$

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

Integrating Initial Conditions . To complete invariance property, invariant set must contain initial conditions. Suppose that initial condition is a polyhedron $X^0 = \{(x, u) \in \mathbb{R}^{d+m} \mid T_w^0(x, u) \leq c_w^0,\ T_s^0(x, u) \ll c_s^0\}$. We must have $X^0 \subseteq S_\alpha$. But X^0 is contained in the union of X^i. Hence 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 $X^0 \subseteq S_\alpha$. We can use the same method as before to express that all sets $S_{i,\alpha}$ such that $X^0 \cap X^i \neq \emptyset$ must contain $X^0 \cap X^i$. In term of implications, it can be rewritten as for all $i \in I$ such that $X^0 \cap X^i \neq \emptyset$:

$$(x, u) \in X^0 \cap X^i \implies (x, u)P^i(x, u)^\intercal + 2(x, u)q^i \leq \alpha \tag{13}$$

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

$$E_w^{0i} = \begin{pmatrix} c_w^0 & -T_w^0 \\ c_w^i & -T_w^i \end{pmatrix} \text{ and } E_s^{0i} = \begin{pmatrix} 1 & 0_{1\times(d+m)} \\ c_s^0 & -T_s^0 \\ c_s^i & -T_s^i \end{pmatrix}$$

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

For all $i \in I$ such that $X^0 \cap X^i \neq \emptyset$, we obtain a stronger notion by introducing semi-definite constraints:

$$-\begin{pmatrix} -\alpha & q^{i\intercal} \\ q^i & P^i \end{pmatrix} - E^{0i\intercal}Z^iE^{0i} \succeq 0 \tag{14}$$

Proposition 2 proves that if (P^i, q^i, Z^i) is a solution of Equation (14) then (P^i, q^i, Z^i) satisfies Equation (13).

Note since $X^0 \cap X^i$ is a polyhedron then its emptyness can be decided by checking the feasibility of the linear problem (15) and by using of same argument than Proposition 1.

$$\begin{cases} (E_s^{0i})^\intercal p^s + (E_w^{0i})^\intercal p = 0 \\ \sum_{k\in\mathbb{I}} p_k^s = 1 \\ p_k^s \geq 0, \ \forall\, k \in \mathbb{I} \\ p_i \geq 0, \ \forall\, i \notin \mathbb{I} \end{cases} \tag{15}$$

Linear program (15) is feasible iff $X^0 \cap X^i = \emptyset$.

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 $\beta \geq 0$:

$$(x,u) \in X^i \wedge \begin{pmatrix} x \\ u \end{pmatrix} \in S_{i,\alpha} \implies \|(x,u)\|_2^2 \leq \beta \tag{16}$$

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 $\|(x,u)\|_2^2 = \begin{pmatrix} x \\ u \end{pmatrix}^\intercal \mathrm{Id}_{(d+m)\times(d+m)} \begin{pmatrix} x \\ u \end{pmatrix}$ and we get for all $i \in I$:

$$\begin{aligned} &\begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal E^{i\intercal}W^iE^i \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \geq 0 \text{ and } \begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal \begin{pmatrix} -\alpha & q^{i\intercal} \\ q^i & P^i \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \leq 0 \\ \implies &\begin{pmatrix} 1 \\ x \\ u \end{pmatrix}^\intercal \begin{pmatrix} -\beta & 0_{1\times(d+m)} \\ 0_{(d+m)\times 1} & \mathrm{Id}_{(d+m)\times(d+m)} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ u \end{pmatrix} \leq 0 \end{aligned} \tag{17}$$

where E^i is defined in Equation (4).

Finally, as invariance we obtain a stronger condition by considering semi-definite constraint such as Equation (18). Proposition 2 proves that (P^i, q^i, W^i) is a solution of Equation (18) the (P^i, q^i, W^i) satisfies Equation (17). For all $i \in I$:

$$-{E^i}^\intercal W^i E^i + \begin{pmatrix} -\alpha & {q^i}^\intercal \\ q^i & P^i \end{pmatrix} + \begin{pmatrix} \beta & 0_{1\times(d+m)} \\ 0_{(d+m)\times 1} & -\mathrm{Id}_{(d+m)\times(d+m)} \end{pmatrix} \succeq 0 \qquad (18)$$

Method to Compute Invariant Set for Piecewise Affine Systems and Prove the Boundedness of Its Reachable Set. To compute a piecewise ellipsoidal invariant set for a piecewise affine systems of the form (1) whose initial conditions is a polyhedron, we can proceed as follows:

1. Define a matrix L of size $I \times I$ following Equation (7): set $L(i,j) = 1$ if Problem (8) is not feasible and $L(i,j) = 0$ otherwise;
2. Define the real variables α, β;
3. For $i \in I$, compute the matrix E^i of Equation (4) 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 ($\sharp$ lines of E^i) $\times$ ($\sharp$ lines of E^i) and add the constraint (18). If Problem (15) is not feasible, add Constraint (14);
4. For all $(i,j) \in I^2$, if $L(i,j) = 1$ construct the matrix E^{ij} defined by Equation (5) and define the symmetric matrix variable $U^{i,j}$ of the size ($\sharp$ lines of E^{ij}) $\times$ ($\sharp$ lines of E^{ij}) with nonnegative coefficients and add the constraint (12);
5. Add as linear objective function the sum of α and β to minimize;
6. Solve the semi-definite program;
7. If there exists a solution then the set $\bigcup_{i\in I}\{(x,u) \in X^i \mid (x,u)P^i_{opt}(x,u)^\intercal + 2(x,u)q^i_{opt} \leq \alpha_{opt}\}$ is a bounded invariant of system (1) and the norm $\|(x,u)\|$ is less than β_{opt} for all the reachable (x,u) of the system.

3.4 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 $\|(x,y,u)\|_2^2 = x^2+y^2+u^2 \leq \beta_{opt}$. 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:

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

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

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

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

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 $(x, u) \mapsto \alpha - (x, u)P^1(x, u)^\intercal - 2(x, u)q^i$ is nonnegative 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 $(1, x, u)$ by F^1 belongs to the set $S_{j,\alpha}$ for all $(1, x, u)$ such that $F^1(1, x, u) \in X^j$.

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

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

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

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

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 returns 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.

4 Experimentations

To illustrate the applicability of our method to a wide set of examples, we generated about a thousand of dynamical systems with at most 4 partition cells, 4 state variables and a single input.

In [BBK+01], the authors show (Theorem 2) that to determine the stability a piecewise affine dynamical system is undecidable. In order to generated more stable examples, we restricted the class of program generated. Each partition cell affine semantics 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 guaranty 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.

About 300 of such 1000 examples are automatically shown to be bounded using our technique while the class of program considered is unlikely to be analyzable with other static analysis tools the author are aware of, including the previous analyzes proposed [RG13]. A typical run of the analysis, including the time to generate the problem instance, is about 20s.

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/vmcai15/`.

5 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 [CC77] or policy iteration [GSA+12]. 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 [RG13] 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.

Acknowledgement. We thank the anonymous referees for their useful comments regarding the paper.

References

[All09] Allamigeon, X.: Static analysis of memory manipulations by abstract interpretation — Algorithmics of tropical polyhedra, and application to abstract interpretation. PhD thesis, École Polytechnique, Palaiseau, France (November 2009)

[BBK+01] Blondel, V., Bournez, O., Koiran, P., Papadimitriou, C., Tsitsiklis, J.: Deciding stability and mortality of piecewise affine dynamical systems. Theoretical Computer Science A 1–2(255), 687–696 (2001)

[BCC+11] Bertrane, J., Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Rival, X.: Static analysis by abstract interpretation of embedded critical software. ACM SIGSOFT Software Engineering Notes 36(1), 1–8 (2011)

[BGLM05] 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)

[CC77] 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, Los Angeles, California, pp. 238–252. ACM Press, New York (1977)

[CH78] Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: Aho, A., Zilles, S., Szymanski, T. (eds.) POPL, pp. 84–96. ACM Press (1978)

[CS11] Colón, M.A., Sankaranarayanan, S.: Generalizing the template polyhedral domain. In: Barthe, G. (ed.) ESOP 2011. LNCS, vol. 6602, pp. 176–195. Springer, Heidelberg (2011)

[CSS03] Colón, M.A., Sankaranarayanan, S., Sipma, H.B.: Linear invariant generation using non-linear constraint solving. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 420–432. Springer, Heidelberg (2003)

[Fer04] Feret, J.: Static analysis of digital filters. In: Schmidt, D. (ed.) ESOP 2004. LNCS, vol. 2986, pp. 33–48. Springer, Heidelberg (2004)

[FR94] Filé, G., Ranzato, F.: Improving abstract interpretations by systematic lifting to the powerset. In: Logic Programming, Proc. of the 1994 International Symposium, Ithaca, New York, USA, November 13-17, pp. 655–669 (1994)

[GGP09] Ghorbal, K., Goubault, E., Putot, S.: The zonotope abstract domain taylor1+. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 627–633. Springer, Heidelberg (2009)

[GSA+12] Gawlitza, T., Seidl, H., Adjé, A., Gaubert, S., Goubault, E.: Abstract interpretation meets convex optimization. J. Symb. Comput. 47(12), 1416–1446 (2012)

[IS00] Ikramov, K.D., Savel'eva, N.V.: Conditionally definite matrices. Journal of Mathematical Sciences 98(1), 1–50 (2000)

[Joh03] Johansson, M.: On modeling, analysis and design of piecewise linear control systems. In: Proc. of the 2003 International Symposium on Circuits and Systems, ISCAS 2003, vol. 3, pp. III–646–III–649 (May 2003)

[MFTM00] Mignone, D., Ferrari-Trecate, G., Morari, M.: Stability and stabilization of piecewise affine and hybrid systems: An lmi approach. In: Proc. of the 39th IEEE Conference on Decision and Control, vol. 1, pp. 504–509 (2000)

[Min01] Miné, A.: A new numerical abstract domain based on difference-bound matrices. In: Danvy, O., Filinski, A. (eds.) PADO-II 2001. LNCS, vol. 2053, pp. 155–172. Springer, Heidelberg (2001)

[Min06] Miné, A.: The octagon abstract domain. Higher-Order and Symbolic Computation 19(1), 31–100 (2006)

[MJ81] Martin, D.H., Jacobson, D.H.: Copositive matrices and definiteness of quadratic forms subject to homogeneous linear inequality constraints. Linear Algebra and its Applications 35(0), 227–258 (1981)

[Mot51] Motzkin, T.S.: Two consequences of the transposition theorem on linear inequalities. Econometrica 19(2), 184–185 (1951)

[RG13] Roux, P., Garoche, P.-L.: Integrating policy iterations in abstract interpreters. In: Van Hung, D., Ogawa, M. (eds.) ATVA 2013. LNCS, vol. 8172, pp. 240–254. Springer, Heidelberg (2013)

[RJ00] Rantzer, A., Johansson, M.: Piecewise linear quadratic optimal control. IEEE Transactions on Automatic Control 45(4), 629–637 (2000)

[RJGF12] Roux, P., Jobredeaux, R., Garoche, P.-L., Feron, E.: A generic ellipsoid abstract domain for linear time invariant systems. In: Dang, T., Mitchell, I. (eds.) HSCC, pp. 105–114. ACM (2012)