Transactions of the Institute of Measurement and Control 32, 5 (2010) pp. 445–467

Reachability analysis of multi-affine systems*

Marius Kloetzer and Calin Belta

Division of Systems Engineering, Department of Mechanical Engineering,
Boston University, 15 Saint Mary's St., Brookline, MA 02446, USA

We present a computationally attractive technique to study the reachability of rectangular regions by trajectories of continuous multi-affine systems. The method is iterative. At each step, finer partitions and finite quotients that over-approximate the reachability properties of the initial system are produced. We exploit some convexity properties of multi-affine functions on rectangles to show that the construction of the quotient at each step requires only the evaluation of the vector field at the set of all vertices of all rectangles in the partition and finding the roots of a finite set of scalar affine functions. This methodology can be used for formal analysis of biochemical networks, aircraft and underwater vehicles, where multi-affine models are widely used.

Key words: formal verification; decidability; abstractions; nonlinear systems;

1. Introduction

The dynamics of processes found in nature or made by humans are traditionally modelled using *continuous systems*, ie, by a set of differential equations capturing the time evolution of quantities of interest. Discrete dynamics arise from integration of embedded computers and action of valves, gears, and switches in man-made systems, or occur naturally because of time scale separation. For example, in genetic networks, genes are turned 'on' and 'off' and determine the continuous behaviour of very large metabolic networks causing the phenotype of the cell. Such systems with both continuous and discrete dynamics are called *hybrid systems*. With the increasing

*Preliminary results of this work were communicated at the 9th International Workshop on Hybrid Systems: Computation and Control (HSCC), Santa Barbara, CA, 2006.

Address for correspondence: Calin Belta, Division of Systems Engineering, Department of Mechanical Engineering, Boston University, 15 Saint Mary's St., Brookline, MA 02446, USA. E-mail: cbelta@bu.edu

Figures 2–4 appear in colour online: http://tim.sagepub.com

10.1177/0142331208097838

complexity of systems such as embedded controllers and engineered genetic networks, safety is an important issue to be considered during the analysis and even the design process. The goal of safety verification is to formally prove that a system does not reach a set of ‘bad’, or ‘unsafe’ states.

Two central problems in *formal analysis* are reachability analysis and safety verification. The goal of *reachability analysis* is to construct the set of states reached by trajectories of the system originating in a given (possibly infinite dimensional) initial set. *Safety verification* is the problem of proving that a system does not have any trajectory from a given initial set to a given final (unsafe) set. For systems with finitely many states, these problems are *decidable*, ie, can be solved by a computer in a finite number of steps, since they reduce to searches on finite graphs. For continuous and hybrid systems, these problems are very difficult (in general undecidable) because of the uncountability of the state space.

One way to solve formal analysis problems for continuous and hybrid systems is to construct the set of states reached by the system, or an over-approximation of this set, by working directly in the continuous state space. Such methods are called *direct* and are not the subject of this paper. Our work can be included into the group of *indirect* methods, where a reachability problem for a continuous or hybrid system is mapped to a reachability problem for a finite state discrete system through *discrete abstractions*. The main idea in discrete abstractions is to iteratively partition the infinite dimensional continuous state space and produce partition quotients whose trajectories include the trajectories of the continuous or hybrid system. Such a discrete system is said to *simulate* the original system. If the converse is true, ie, the continuous or hybrid system simulate the discrete quotient, the two systems are called *bisimilar*, and the two reachability problems become equivalent. Therefore, in this case, the reachability problem for a continuous or hybrid system becomes decidable.

The bisimulation relation was introduced in Park (1981) and Milner (1989), formally defined for linear systems in Pappas (2003), and for non-linear systems in a categorical context in Haghverdi *et al.* (2005). In Henzinger *et al.* (1998), it has been shown that reachability is undecidable for a very simple class of hybrid systems. Several decidable classes have been identified though by restricting the continuous behaviour of the hybrid system, as in the case of timed automata (Alur and Diu, 1994), multirate automata (Alur *et al.*, 1993; Nicolin *et al.*, 1993) and rectangular automata (Puri and Varaiya, 1994; Henzinger *et al.*, 1998), or by restricting the discrete behaviour, as in order-minimal hybrid systems (Lafferriere *et al.*, 2000). All these decidable classes are too weak to represent continuous and hybrid system models encountered in practice. Then one might be satisfied with sufficient abstractions, when a discrete quotient that simulates the original system is enough to prove a safety property. But even finding the discrete quotient is not at all trivial. Related work focuses on partitioning using linear functions of the continuous variables, as in the method of predicate abstractions (Alur *et al.*, 2002; Tiwari and Khanna, 2002), or using polynomial functions as in Ghosh *et al.* (2003) and Tiwari and Khanna (2002). However, to derive the transitions of the discrete quotient, one has to be able to either integrate the vector fields of the initial system (Alur *et al.*, 2002), or use computationally expensive decision procedures such as quantifier elimination for real closed fields and theorem proving (Tiwari and Khanna, 2002), which severely limit the dimensions of the problems that can be approached.

In this paper, we focus on formal analysis of systems with continuous multi-affine vector fields, ie, affine in each variable, defined in rectangular regions of the Euclidean space. This class of systems includes a particular type of switched systems, characterised by different dynamics in different rectangular regions of the state space, where the dynamics 'match' on the separating facets. The main idea behind this work is, as in Belta and Habets (2006), Habets and Van Schuppen (2004), and Kloetzer and Belta (2008), to exploit the specific form of the vector field and the particular shape of the region to infer reachability properties of infinite uncountable sets of states from properties verified by a finite set of states. Specifically, in Belta and Habets (2006), we proved that a multi-affine function is uniquely determined by its values at the vertices of a rectangle and its restriction to the rectangle is a convex combination of these values. We used these properties to derive sufficient conditions for the existence of a feedback controller driving all initial states of a control system with multi-affine drift and constant control directions through a desired facet of a rectangle in finite time. In this paper we use these properties of multi-affine functions on rectangles to prove safety properties of multi-affine systems by iteratively constructing finer and finer discrete quotients.

Even though the abstraction procedure in this paper falls into the more general framework of Tiwari and Khanna (2002), we show that if more structure is allowed, then reachability and safety verification questions can be answered with much less computation. Indeed, as it will become clear later, the calculation of the discrete quotient at a given iteration involves only finding the roots of scalar affine functions and evaluation of multi-affine functions at a finite number of points, as opposed to quantifier elimination for real closed fields as in Tiwari and Khanna (2002). This will allow us to approach much problems, as usually found in analysis of bio-molecular networks, where the multi-affine structure appears naturally when chemical reactions with unitary stoichiometric coefficients are modelled using mass action kinetics (Batt *et al.*, 2008). Multi-affine dynamics are also found in other systems, including the celebrated Euler's equations for angular velocity of rotation of rigid bodies, the equations of motion of translating and rotating rigid bodies with rotation parameterised by quaternions (Belta and Habets, 2006), Volterra (1926), and Lotka–Volterra equations (Lotka, 1925).

The remainder of the paper is organised as follows: In Section 2, we define the reachability and safety verification problems and introduce partitions and discrete quotients. Rectangles and multi-affine functions are reviewed in Section 3, together with some convexity properties, which are fundamental for the rest of the paper. The main result of the paper, which is an algorithm for safety verification of systems with multi-affine vector fields, is included in Section 4. Illustrative case studies are presented in Section 5. The paper concludes with final remarks and outline of future work in Section 7.

2. Continuous systems and discrete quotients

Definition 1 (Continuous system). We represent a continuous dynamical system as a pair

$$CS = (X, f) \tag{1}$$

where $X \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$ is its continuous state space and f is a continuous vector field on X. In other words, the state $x \in X$ of system (1) evolves according to $\dot{x} = f(x)$.

We assume that X is a connected subset of $\mathbb{R}^n$ and introduce a *set partition* of X by defining an *abstraction map*

$$abs : X \to L \tag{2}$$

where L is a finite set of labels for all the elements in the partition. As it will be seen later in the paper, such a partition can be obtained using a finite set of polynomials and in this case L corresponds to evaluations of the polynomials over a finite set of values. Let *con* be the *concretization map* of the partition induced by *abs*:

$$con : L \to X,\ con(l) = \{x \in X | abs(x) = l\} \tag{3}$$

In other words, for $l \in L$, we use $con(l) \subseteq X$ to denote the set of all $x \in X$ in the partition element with label l. Since *abs* induces a partition and *con* is its concretisation map, we have $\bigcup_{l \in L} con(l) = X$ and $con(l) \bigcap con(l') = \emptyset$, for all $l, l' \in L$, $l \neq l'$. We use $con(l) \sim con(l')$, or simply $l \sim l'$ to denote adjacency of regions $con(l)$ and $con(l')$. For simplicity of notation, we use $con(I)$ to denote $\bigcup_{l \in I} con(l)$, where I is an arbitrary subset of L. For an arbitrary $I \subset L$, we denote by $\text{Post}(con(I))$ the set of all states in X reached by the trajectories of (1) originating in $con(I)$. The reachability problem for CS can be formulated as follows:

Problem 1 (Reachability). *For an arbitrary* $I \subset L$, *determine* $\text{Post}(con(I))$.

The safety verification problem for CS is the problem of deciding whether system (1) has trajectories between two regions in the partition induced by the map *abs*:

Problem 2 (Safety). *Given* $I, F \subset L$ *with* $I \bigcap F = \emptyset$, *determine the truth value of the following assertion*:

$$\text{Post}(con(I)) \bigcap con(F) = \emptyset \tag{4}$$

In a particular application, $con(I)$ corresponds to a set of states around initial or operating points of a system CS, while $con(F)$ might represent unsafe regions of operation.

It is obvious to see that the solution to Problem 1 immediately gives a solution to Problem 2, provided that we can calculate the intersection in Equation (4). However, in order to solve Problem 2, it is not necessary to solve Problem 1 – it is enough to construct an over-approximation of Post(*con*(*I*)) that has empty intersection with *con*(*F*). To construct over-approximations of Post(*con*(*I*)), we use *discrete quotients*:

Definition 2 (Discrete quotient). A discrete quotient of *CS* induced by the partition map '*abs*' is a finite state transition system *DS* described by the pair

$$DS = (L, T) \tag{5}$$

where *L* is the set of labels produced by the partition as in Equation (2), and $T \subseteq L \times L$ is a set of transitions satisfying the following property:

$$\begin{gathered}(l, l') \in T \text{ if } l \sim l' \textit{ and } \text{there exist } t_1, t_2 \geq 0,\\ t_1 < t_2 \text{ and a trajectory } x(t) \text{ of } CS \text{ such that}\\ x(t_1) \in con(l),\ x(t_2) \in con(l') \text{ and}\\ x(t) \in (con(l)) \cap con(l'), \text{ for all } t \in [t_1, t_2].\end{gathered} \tag{6}$$

Condition (6) means that *DS* has a transition from *l* to *l′* if *con*(*l*) and *con*(*l′*) are adjacent and *CS* has a trajectory from *con*(*l*) to *con*(*l′*). As before, for $I \subset L$, we denote by $\text{Post}(I) \subseteq L$ the set of all discrete states reached from *I* by *DS*. More formally

$$\text{Post}(I) = \bigcup_{l \in I} \text{Post}(l) \tag{7}$$

Note that we use the same operator Post for both *CS* and *DS*, with the observation that they are easily distinguished by their arguments. From (6) it follows that

$$\text{Post}(con(I)) \subseteq con(\text{Post}(I)) \tag{8}$$

Equation (8) implies that, if the transitions (6) of a discrete quotient (5) can be computed, then an over-approximation *con*(Post(*I*)) of Post(*con*(*I*)) can be easily determined by a search on the finite transition system (5), which is a decidable problem. If $\text{Post}(I) \bigcap F = \emptyset$ (which is equivalent with $con(\text{Post}(I)) \cap con(F) = \emptyset$, since *con*(*L*) is a partition of *X*), then the truth value of (4) is TRUE. Otherwise, we cannot answer Problem 2, and a less conservative discrete quotient is necessary.

There are two sources of conservativeness in the definition of *DS*. The first comes from the fact that, according to Equation (6), there might exist a transition $(l, l') \in T$ even if *CS* does not have a trajectory from *con*(*l*) to *con*(*l′*). A more correct definition of the discrete quotient should have 'if and only 'if' instead of 'if' in Equation (6). This would make *CS* and *DS* equivalent with respect to reachability of adjacent regions in one step. However, even in this case, there is a second source of conservativeness, which comes from lack of transitivity in the following sense: if $(l, l') \in T$ and $(l', l'') \in T$, which implies that l, l', l'' is a trajectory of DS, this does not imply that CS has a trajectory from $con(l)$ to $con(l')$ and to $con(l'')$, simply because it is possible that all trajectories that go from $con(l)$ to $con(l')$ escape to a region $con(l'')$, with $l''' \neq l'''$. The conservativeness is completely eliminated, ie, CS and DS are equivalent with respect to reachability properties, if and only if, in (6), the 'if' statement is replaced by 'if and only if', and all initial states in $con(l)$ flow in finite time to $con(l')$ under the dynamics of CS.

As outlined in Section 1, finding such non-conservative discrete quotients of continuous systems is an extremely hard problem. Moreover, even finding discrete quotients with 'if and only if' in Equation (6) is very difficult, since, as it will become clear later in the paper, there are situations in which $\text{Post}(con(l))$ cannot be computed exactly. In this paper, we use the relaxed Definition 2 of a discrete quotient to construct less and less conservative over-approximations $con(\text{Post}(I))$ for the solutions to Problems 1 and 2. Formally, we define a refinement of a discrete quotient as follows:

Definition 3 (Refinement). For a given continuous system CS, a discrete quotient $\overline{DS} = (\bar{L}, \bar{T})$ induced by $\overline{abs} : X \to \bar{L}$ refines a discrete quotient $DS = (L, T)$ induced by $abs : X \to L$ if $|\bar{L}| > |L|$ and the following three conditions hold:

(i) For any $l \in L$, there exists $\bar{I} \subset \bar{L}$ with $|\bar{I}| \geq 1$ so that $\overline{con}(\bar{I})$ is a partition of $con(l)$. Any $\bar{l} \in \bar{I}$ is said to refine $l \in L$, and we denote this by $\bar{l} \leq l$.

(ii) For any $\bar{l}, \bar{l}' \in \bar{L}$ with $(\bar{l}, \bar{l}') \in \bar{T}$, if there exist $l, l' \in L$, $l \neq l'$, so that $\bar{l} \leq l$ and $\bar{l}' \leq l'$, then $(l, l') \in T$.

(iii) There exist $l, l' \in L$ with $(l, l') \in T$ and $\bar{l}, \bar{l}' \in \bar{L}$ with $\bar{l} \sim \bar{l}'$, $\bar{l} \leq l$, $\bar{l}' \leq l'$, and $(\bar{l}, \bar{l}') \notin \bar{T}$.

In other words, (i) states that each region in the partition produced by abs is further partitioned by $\overline{abs}$. Note that, since $|\bar{L}| > |L|$, at least one region $con(l)$ is strictly partitioned. Condition (ii) requires that the finer quotient $\overline{DS}$ can only have transitions between states refining states connected by transitions in the coarser quotient DS and between states refining the same state of the coarser quotient DS. Conditions (i) and (ii) will guarantee that the over-approximation $con(\text{Post}(I))$ as in Equation (8) does not grow through refinement. Finally, (iii) means that there exist at least one pair of states connected in the coarser DS for which refinement determines two disconnected states in the finer description $\overline{DS}$.

An example is given in Figure 1, where an initial partition $\bigcup_{i,j=0,1,2} con(l_{ij})$ of a two-dimensional rectangle (containing its boundaries) is refined to $\bigcup_{i=0,1,2;\, j=0,\ldots,4} \overline{con}(\bar{l}_{ij})$. It is easy to see that condition (ii) of Definition 3 is satisfied, ie, no 'new' transitions are added. As it can be seen in Figure 1(b), the refinement is achieved by 'cutting' with a horizontal line where the f_1 component of the vector field becomes zero on the vertical open segment $con(l_{21})$. This leads to a partition $\overline{con}(\bar{l}_{13})$, $\overline{con}(\bar{l}_{12})$, $\overline{con}(\bar{l}_{11})$ of $con(l_{11})$ and a partition $\overline{con}(\bar{l}_{23})$, $\overline{con}(\bar{l}_{22})$, $\overline{con}(\bar{l}_{21})$ of $con(l_{21})$. In the finer quotient, it can be seen for example that there is no transition from $\bar{l}_{21}$ to $\bar{l}_{11}$ and from $\bar{l}_{13}$ to $\bar{l}_{23}$, even though the coarser quotient had transitions between l_{11} and l_{21} in both directions (condition (iii)).

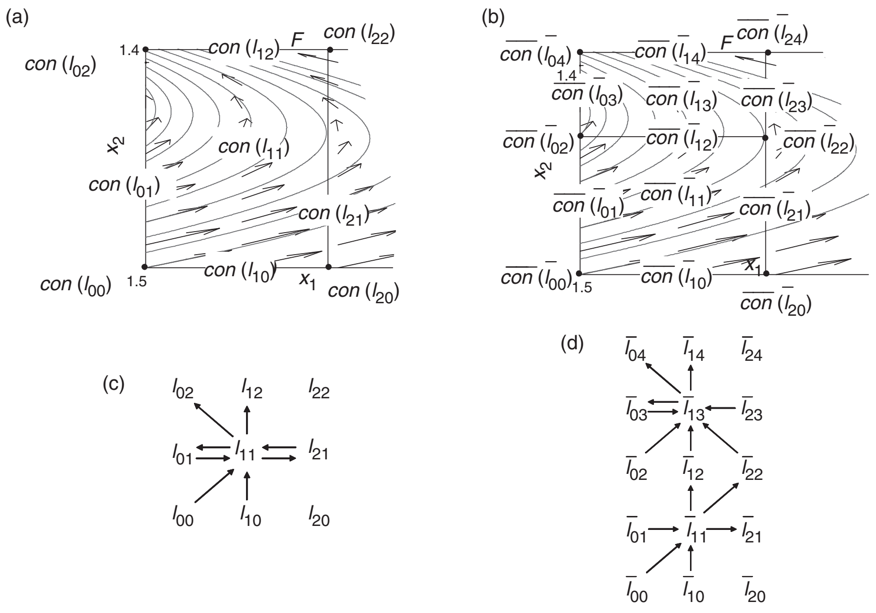

Figure 1 Discrete quotients for a vector field $f = (f_1, f_2)$, $f_1 = 2 - x_1x_2$, $f_2 = 1 + x_2 - x_1x_2$ in a rectangular region [1.5, 1.56]×[1.1, 1.42] in plane. An initial partition and the corresponding discrete quotient are shown in (a) and (c), respectively. A finer partition is shown in (b), and the corresponding discrete quotient (d) refines the initial one (c). The regions of the partition are 'open' rectangles of dimension 0 (points), 1 (open line segments), and 2 (rectangles without boundaries). The transitions of the discrete quotients correspond to 'if and only if' in Equation (6). Only the transitions among the discrete states corresponding to the closed rectangle [1.5, 1.56]×[1.1, 1.42] are shown

Condition (iii) in Definition 3 is a necessary condition for strict shrinking of the over-approximation $con(\mathrm{Post}(I))$. However, it is not sufficient. Indeed, for adjacent regions $\bar{l} \sim \bar{l}'$, if CS does not have trajectories penetrating directly from $\overline{con}(\bar{l})$ to $\overline{con}(\bar{l}')$, this does not mean that $Post(\overline{con}(\bar{l})) \bigcap \overline{con}(\bar{l}') = \emptyset$. Trajectories originating in $\overline{con}(\bar{l})$ can loop around and eventually hit $\overline{con}(\bar{l}')$.

These ideas are formalised in Proposition 1.

Proposition 1 (Reduction of conservativeness through refinement). *If* $\overline{DS} = (\bar{L}, \bar{T})$ *refines* $DS = (L, T)$, *and* $I \subset L$, $\bar{I} \subset \bar{L}$ *with the property that* $\overline{con}(\bar{I})$ *is a partition of* $con(I)$, *then we have*

$$\mathrm{Post}(con(I)) = \mathrm{Post}(\overline{con}(\bar{I})) \subseteq \overline{con}(\mathrm{Post}(\bar{I})) \subseteq con(\mathrm{Post}(I)) \qquad (9)$$

Moreover, if (iii) from Definition 3 is replaced by:

(iii)′ *There exist* $l, l' \in L$ *with* $(l, l') \in T$ *and* $\bar{l}' \in \bar{L}$ *with* $\bar{l}' \leq l'$, *and* $\bar{l}' \notin \mathrm{Post}(\bar{l})$, $\forall \bar{l} \in \bar{L}$, $\bar{l} \leq l$

and $l \in (I \cup \mathrm{Post}(I))$, *then the last inclusion relation in (9) is strict, ie, the over-approximation* $con(\mathrm{Post}(I))$ *as in Equation (8) strictly shrinks through refinement.*

Proof. The equality from (9) is an immediate consequence of the fact that $\overline{con}(\bar{I})$ and $con(I)$ cover the same region in X. The first inclusion from (9) is given by (8), so we only have to prove the last inclusion relation.

Since $\overline{DS}$ refines DS, (i) from Definition 3 implies that for any $l \in L$, $con(l) = con\left(\bigcup_{\bar{l} \in \bar{L}, \bar{l} \leq l} \bar{l}\right)$. Condition (ii) implies that $con\left(\bigcup_{\bar{l} \in \bar{L}, \bar{l} \leq l} \mathrm{Post}(\bar{l})\right) \subseteq con(\mathrm{Post}(l))$ for any $l \in L$, and using Equation (7) the last inclusion from (9) is verified. In other words, the over-approximation $con(\mathrm{Post}(I))$ as in Equation (8) does not grow through refinement, $\forall I \subset L$.

From $\bar{l}' \notin \mathrm{Post}(\bar{l})$, $\forall \bar{l} \leq l$, we have $con(\bar{l}')con\left(\bigcup_{\bar{l} \in \bar{L}, \bar{l} \leq l} \mathrm{Post}(\bar{l})\right)$. $(l, l') \in T$ implies that $con(l') \subseteq con(\mathrm{Post}(l))$, while $\bar{l}' \leq l'$ implies $con(\bar{l}') \subseteq con(l')$, so $con(\bar{l}') \subseteq con(\mathrm{Post}(l))$. We showed that (ii) implies $con\left(\bigcup_{\bar{l} \in \bar{L}, \bar{l} \leq l} \mathrm{Post}(\bar{l})\right) \subseteq con(\mathrm{Post}(l))$, so we conclude that $con\left(\bigcup_{\bar{l} \in \bar{L}, \bar{l} \leq l} \mathrm{Post}(\bar{l})\right) \subset con(\mathrm{Post}(l))$, which proves the equivalence between (iii)′ and strict shrinking of $con(\mathrm{Post}(l))$. Since (iii)′ implies (iii), (iii) is a necessary condition for constructing strictly less conservative over-approximations of $\mathrm{Post}(con(I))$. ■

Remark 1 (Simulation and bisimulation). *Any discrete quotient DS as in Definition 2 is said to simulate the initial continuous system CS. When both sources of conservativeness mentioned above are eliminated* (*ie, the 'if' statement in* (6) *is replaced by 'if and only if', and all initial states in con*(*l*) *flow in finite time to con*(*l*′) *under the dynamics of CS, for all* $l \in L$), *then CS simulates DS as well, and they are called bisimilar* (*Park, 1981; Milner, 1989*).

In this paper, we assume that X is a full dimensional 'closed' rectangle in $\mathbb{R}^n$ and the vector field f is multi-affine, ie, affine in each state component. We use iterative partitions of X into 'open' rectangles and some convexity properties of multi-affine functions on rectangles to calculate discrete quotients according to Definitions 2 and 3 and provide a solution to Problem 2 and a conservative solution to Problem 1. As it will be seen, we cannot guarantee the sufficient condition (iii)′ for strict shrinking at each step of the refinement. Instead, we satisfy the necessary condition (iii), with the 'hope' that the conservativeness is strictly reduced.

The following section gives all the necessary definitions and notation.

3. Rectangles and multi-affine functions

Two vectors $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$ and $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ with the property that $a_i < b_i$ for all $i = 1, \ldots, n$ determine a set of 3^n rectangles in $\mathbb{R}^n$:

$$\mathcal{R}(a, b) = \{R_{(l_1, \ldots, l_n)},\ l_i \in \{0, 1, 2\},\ i = 1, \ldots, n\} \tag{10}$$

where each rectangle $R_{(l_1,\ldots,l_n)}$, $l_i \in \{0,1,2\}$, $i = 1,\ldots,n$ is defined by

$$R_{(l_1,\ldots,l_n)} = \{x = (x_1,\ldots,x_n) \in \mathbb{R}^n \mid x_i = a_i \text{ if } l_i = 0,\ a_i < x_i < b_i \text{ if } l_i = 1, \\ x_i = b_i \text{ if } l_i = 2,\ i = 1,\ldots,n\} \tag{11}$$

We define the order m of a rectangle $R_{(l_1,\ldots,l_n)}$ as being the number of '1' entries in its label $(l_1,\ldots,l_n)$. The number of m-order rectangles in $\mathcal{R}(a,b)$ is $2^{n-m}n!/((n-m)!m!)$. As particular cases, there is only one n-order (full dimensional) rectangle $R_{(1,\ldots,1)}$, and 2^n zero-order rectangles, or *vertices* $R_{(l_1,\ldots,l_n)}$, $l_i \in \{0,2\}$, $i = 1,\ldots,n$. For a given rectangle $R_{(l_1,\ldots,l_n)}$, we can define

$$\mathcal{L}R_{(l_1,\ldots,l_n)} = \{R_{(l'_1,\ldots,l'_n)} \in \mathcal{R}(a,b) \mid (l'_1,\ldots,l'_n) \neq (l_1,\ldots,l_n) \wedge l'_i = l_i \text{ if } l_i \in \{0,2\}\} \tag{12}$$

The set of vertices corresponding to $R_{(l_1,\ldots,l_n)}$ is a subset of $\mathcal{L}R_{(l_1,\ldots,l_n)}$ defined by

$$\mathcal{V}R_{(l_1,\ldots,l_n)} = \{R_{(l'_1,\ldots,l'_n)} \in \mathcal{R}(a,b) \mid (l'_1,\ldots,l'_n) \neq (l_1,\ldots,l_n) \\ \wedge\ l'_i = l_i \text{ if } l_i \in \{0,2\} \ \wedge\ l'_i \in \{0,2\} \text{ if } l_i = 1\} \tag{13}$$

If the order of $R_{(l_1,\ldots,l_n)}$ is m, there are $3^m - 1$ rectangles in $\mathcal{L}R_{(l_1,\ldots,l_n)}$, all of order less than or equal to $m-1$, and 2^m vertices (zero-order rectangles) in $\mathcal{V}R_{(l_1,\ldots,l_n)}$. We call the rectangles defined by (11) *open* rectangles, with the observation that, except for $R_{(1,\ldots,1)}$, they are not open sets in $\mathbb{R}^n$. If all '$<$' in (11), if any, are replaced by '$\leq$', then $R_{(l_1,\ldots,l_n)}$ becomes *closed*, and is denoted by $\bar{R}_{(l_1,\ldots,l_n)}$. It is easy to see that $\bar{R}_{(l_1,\ldots,l_n)} = R_{(l_1,\ldots,l_n)} \cup \mathcal{L}R_{(l_1,\ldots,l_n)}$. For a closed rectangle $\bar{R}$, the sets $\mathcal{L}\bar{R}$ and $\mathcal{V}\bar{R}$ are defined as in (12) and (13) by replacing R with $\bar{R}$. It follows immediately that the sets of vertices of open and closed rectangles are identical, ie, $\mathcal{V}R = \mathcal{V}\bar{R}$. Therefore we will use $\mathcal{V}R$ for the set of vertices of $\mathcal{V}\bar{R}$.

Definition 4 (Multi-affine function) A multi-affine function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ (with $p \in \mathbb{N}$) is a polynomial in the indeterminates $x_1,\ldots,x_n$ with the property that the degree of f in any of the variables is less than or equal to 1. Stated differently, f has the form:

$$f(x_1,\ldots,x_n) = \sum_{i_1,\ldots,i_n \in \{0,1\}} c_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n} \tag{14}$$

with $c_{i_1,\ldots,i_n} \in \mathbb{R}^p$ for all $i_1,\ldots,i_n \in \{0,1\}$ and using the convention that if $i_k = 0$, then $x_k^{i_k} = 1$.

The following proposition is proved in Belta and Habets (2006):

Proposition 2. A multi-affine function is uniquely determined by its values at the vertices $\mathcal{V}R_{(1,\ldots,1)}$ of a full dimensional closed rectangle $\bar{R}_{(1,\ldots,1)}$. Its restriction to the rectangle is a convex combination of the values at the vertices and has the following form:

$$f|_{\bar{R}_{(1,\ldots,1)}}(x_1,\ldots,x_n) = \sum_{(v_1,\ldots,v_n) \in \mathcal{V}R_{(1,\ldots,1)}} \prod_{k=1}^{n} \left(\frac{x_k - a_k}{b_k - a_k}\right)^{\xi_k(v_k)} \left(\frac{b_k - x_k}{b_k - a_k}\right)^{1-\xi_k(v_k)} f(v_1,\ldots,v_n) \tag{15}$$

where $\xi_k : \{a_1, \ldots, a_n, b_1, \ldots, b_n\} \to \{0, 1\}$ *is an indicator function defined by*

$$\xi_k(a_k) = 0, \quad \xi_k(b_k) = 1, \ \ k = 1, \ldots, n \tag{16}$$

Since a multi-affine function remains multi-affine if some of its arguments are kept constant, Proposition 2 is true when a multi-affine function is restricted to a lower order closed rectangle, when Equation (15) becomes:

$$f|_{\bar{R}_{(l_1,\ldots,l_n)}}(x_1, \ldots, x_n) = \sum_{(v_1,\ldots,v_n)\in \mathcal{V}R_{(l_1,\ldots,l_n)}} \prod_{k, l_k=1} \left(\frac{x_k - a_k}{b_k - a_k}\right)^{\xi_k(v_k)} \left(\frac{b_k - x_k}{b_k - a_k}\right)^{1-\xi_k(v_k)} f(v_1, \ldots, v_n) \tag{17}$$

Note that $f|_{\bar{R}_{(l_1,\ldots,l_n)}}(x_1, \ldots, x_n)$ is obtained from $f|_{\bar{R}_{(1,\ldots,1)}}(x_1, \ldots, x_n)$ by setting $x_i = a_i$ for $l_i = 0$ and $x_i = b_i$ for $l_i = 2$, $i = 1, \ldots, n$.

Proposition 3. *If* f *is a scalar multi-affine function* ($p = 1$ *in Definition* 4) *and* $R_{(l_1,\ldots,l_n)}$ *is an open rectangle of arbitrary order, then we have*:

(a) $f(x) > 0$ *everywhere in* $R_{(l_1,\ldots,l_n)}$ *if and only if* $f(v) \geq 0$ *for all* $v \in \mathcal{V}R_{(l_1,\ldots,l_n)}$, *and there exists at least one* $v \in \mathcal{V}R_{(l_1,\ldots,l_n)}$ *for which* $f(v) > 0$.
(b) $f(x) < 0$ *everywhere in* $R_{(l_1,\ldots,l_n)}$ *if and only if* $f(v) \leq 0$ *for all* $v \in \mathcal{V}R_{(l_1,\ldots,l_n)}$, *and there exists at least one* $v \in \mathcal{V}R_{(l_1,\ldots,l_n)}$ *for which* $f(v) < 0$.
(c) $f(x) = 0$ *everywhere in* $R_{(l_1,\ldots,l_n)}$ *if and only if* $f(v) = 0$ *for all* $v \in \mathcal{V}R_{(l_1,\ldots,l_n)}$.
(d) *There exist* $x, x' \in R_{(l_1,\ldots,l_n)}$ *with* $f(x) > 0$ *and* $f(x') < 0$ *if and only if there exist* $v, v' \in \mathcal{V}R_{(l_1,\ldots,l_n)}$ *with* $f(v) > 0$ *and* $f(v') < 0$.

Proof. The restriction of f to an open rectangle is a strictly positive convex combination of the values at the vertices, ie, all the coefficients multiplying $f(v_1, \ldots, v_n)$ in the sum in Equation (17) are strictly positive. (a), (b), and (c) follow immediately from this property. Note that (c) remains true for closed rectangles, since any convex combination is enough to prove it. (d) is also immediate from continuity of f and by noting that any open neighbourhood of a vertex in $\mathcal{V}R_{(l_1,\ldots,l_n)}$ has a non-empty intersection with $R_{(l_1,\ldots,l_n)}$. ■

4. Reachability analysis of multi-affine systems

We now have all the necessary background to consider Problems 2 and 1 for a continuous system *CS* (Definition 1) , whose continuous state space is a closed rectangle in $\mathbb{R}^n$ defined by $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$ and $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$, $a_i < b_i$ for all $i = 1, \ldots, n$:

$$X = \{x = (x_1, \ldots, x_n) \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i, \ i = 1, \ldots, n\} \tag{18}$$

and whose vector field f is multi-affine as in Definition 4 (with $p = n$).

We first define a partition of X into open rectangles, which gives the states of the discrete quotient DS (Definition 2). We then define the transitions of DS and an algorithm for refinement according to Definition 3. Finally, we collect all the results in an iterative algorithm for safety verification of multi-affine systems.

4.1 The states of the discrete quotient

We assume that each axis Ox_i, $i = 1, \dots, n$ is divided into $n_i \geq 1$ intervals by the points $\theta_0^i < \theta_1^i < \cdots < \theta_{n_i}^i$, which for convenience are collected in a vector $\theta^i = (\theta_0^i, \theta_1^i, \dots, \theta_{n_i}^i)$. This induces a partition of X into $\prod_{i=1}^n (2n_i + 1)$ open rectangles. Using the same idea as in Section 3, we label the rectangles with n-uples $(l_1, \dots, l_n)$ by defining an abstraction map (2) as follows:

$$abs(x_1, \dots, x_n) = (l_1, \dots, l_n) \tag{19}$$

where, for each $i = 1, \dots, n$ and $j_i = 0, 1, \dots, n_i$,

$$\begin{aligned} l_i &= 2j_i, & \text{if} \quad & x_i = \theta_{j_i}^i \\ l_i &= 2j_i - 1, & \text{if} \quad & \theta_{j_i - 1}^i < x_i < \theta_{j_i}^i \end{aligned} \tag{20}$$

Remark 2 The connection with the work in Tiwari and Khanna (2002) can be seen as follows: the polynomials $x_i - \theta_{j_i}^i$, $j_i = 0, \dots, n_i$, $i = 1, \dots, n$ define a set of discrete variables, which generate the set L when interpreted over the set of symbols $\{pos, neg, zero\}$ *(with the obvious significance). In this representation, each discrete state $l \in L$ is a word of length $\sum_{i=1}^n n_i + n$ over the set* $\{pos, neg, zero\}$*, and the cardinality of L becomes $|L| = 3^{\sum_{i=1}^n n_i + n}$. However, in our definition* (19), $|L| = \prod_{i=1}^n (2n_i + 1)$. *The dramatic reduction in the number of discrete states comes form the fact that, in the rectangular partition, infeasible combinations of polynomial interpretations are automatically eliminated.*

As defined in Section 3, the number m of odd entries in $l = (l_1, \dots, l_n)$ is the order of the rectangle. Moreover, $con(l)$ is an open m-rectangle in X. From now on, when we refer to rectangles we mean open rectangles. If all l_i's are odd, then $con(l)$ is a (full dimensional) n-order rectangle and if all l_i's are even, then $con(l)$ is a point (vertex), or zero-order rectangle. Inspired by this observation, we define the *order* of a discrete state l as the number of its odd entries.

4.2 The transitions of the discrete quotient

Before we start constructing the set T of transitions from all discrete states $l \in L$, note that, because of the rectangular partition, it is easy to identify a subset of L where transitions are possible, so we don't have to explore the whole L in search for successors. Let

$$\mathcal{H}(l) = \{l' = (l'_1, \dots, l'_n) \in L \mid l' \neq l \wedge l'_i = l_i \, if \, l_i \, odd \wedge l'_i \in \{l_i - 1, l_i, l_i + 1\} \, if \, l_i \, even\} \tag{21}$$

and

$$\mathcal{L}(l) = \{l' = (l'_1, \ldots, l'_n) \in L \mid l' \neq l \wedge l'_i = l_i \, if \, l_i \, even \wedge l'_i \in \{l_i - 1, l_i, l_i + 1\} \, if \, l_i \, odd\} \quad (22)$$

Note that, if l is an m-order discrete state, then all the discrete states in $\mathcal{H}(l)$ are of order strictly greater than m and all the discrete states in $\mathcal{L}(l)$ are of order strictly less than m. For a m-order discrete state $l = (l_1, \ldots, l_n)$, $1 \leq l_i \leq 2n_i - 1$, the cardinality of $\mathcal{H}(l)$ and $\mathcal{L}(l)$ are $3^{n-m} - 1$ and $3^m - 1$, respectively. Given $l \in L$, it is only possible to have discrete transitions towards discrete states in $\mathcal{H}(l) \cup \mathcal{L}(l)$. For a state l with order $m \geq 1$, let $\mathcal{V}(l)$ denote the set of labels of vertices of $con(l)$. Formally,

$$\mathcal{V}(l) = \{l' = (l'_1, \ldots, l'_n) \in L \mid l' \neq l \wedge l'_i = l_i \, if \, l_i \, even \wedge l'_i \in \{l_i - 1, l_i + 1\} \, if \, l_i \, odd\} \quad (23)$$

Before adding discrete transitions to complete the discrete system DS, we assign a *signature* to each discrete state $l \in L$.

Definition 5 (Signature of a discrete state). For a discrete location $l = (l_1, \ldots, l_n) \in L$, the signature $s(l) = (s_1(l), \ldots, s_n(l))$ is a n-uple over the four-valued domain $\{po, ne, ze, in\}$ (ie, positive, negative, zero, indefinite) with the following significance, for all $i = 1, \ldots, n$:

- $s_i(l) = po$, if $f_i(x) > 0$, $\forall x \in con(l)$
- $s_i(l) = ne$, if $f_i(x) < 0$, $\forall x \in con(l)$
- $s_i(l) = ze$, if $f_i(x) = 0$, $\forall x \in con(l)$
- $s_i(l) = in$, if $\exists x \in con(l)$ so that $f_i(x) > 0$ and $\exists x \in con(l)$ so that $f_i(x) < 0$

where $f = (f_1, \ldots, f_n)$ is the vector field of CS.

The first and second cases correspond to the situation when $con(l)$ has an empty intersection with $f_i(x) = 0$. In the third case, $con(l)$ coincides with $f_i(x) = 0$ or $f_i(x) = 0$ contains $con(l)$. In the fourth, there is an intersection between $con(l)$ and $f_i(x) = 0$.

Determining the signatures for zero-order discrete states, *ie,* $l = (l_1, \ldots, l_n) \in L$ with all l_i even, is easy. Indeed, $con(l)$ is a point in X and determining the signatures reduces to evaluating the vector field f at $con(l)$ and determining its sign. Note that the symbol *in* in the signature of such a discrete state cannot appear. Based on Proposition 3, we can now formulate Algorithm 1 to determine the signature of an m-order discrete state $l = (l_1, \ldots, l_n)$, $m > 1$.

For every state $l = (l_1, \ldots, l_n) \in L$, Algorithm 2 creates a set L' such that $l \times L'$ contains transitions of DS starting from l in accordance to Definition 2. Here, we only give an informal and intuitive description of Algorithm 2.

In order to easily describe the transitions from a state with signature entries in the set $\{po, ne, ze\}$, we first introduce a map from these symbols to numbers: $eval : \{po, ne, ze\} \rightarrow \{+1, -1, 0\}$, $eval(po) = +1$, $eval(ne) = -1$, $eval(ze) = 0$. The map $ord : L \rightarrow \{0, \ldots, n\}$ is used to compute the order of a state, as defined in Section 3. Each direction i, $i = 1, \ldots, n$ is considered separately and a set L_i containing all sub-labels l'_i of states l' in which l transits is constructed. The main idea in finding elements of

Algorithm 1 Signature $s(l)$ of a discrete state l

```
for $i = 1, \dots, n$ do
  if $s_i(l') \in \{po, ze\}$ not all $ze$ for all $l' \in \mathcal{V}(l)$ then
    $s_i(l) = po$
  else if $s_i(l') \in \{ne, ze\}$ not all $ze$ for all $l' \in \mathcal{V}(l)$ then
    $s_i(l) = ne$
  else if $s_i(l') = ze$ for all $l' \in \mathcal{V}(l)$ then
    $s_i(l) = ze$
  else
    $s_i(l) = in$
  end if
end for
```

set L_i is to decide the value of l'_i based only on the value of $s_i(l)$. Roughly speaking, if $s_i(l) \in \{po, ne, ze\}$, (*ie*, $f_i(x)$ has a well-defined sign everywhere in $con(l)$ according to Definition 5), then $l'_i = l_i + eval(s_i(l))$. In this case, the added transitions correspond to Definition 2 in which the 'if' statement from Equation (6) is replaced by 'if and only if'. It is interesting to note here that our algorithm properly deals with situations in which, judged by the signature $s(l)$ of l, transitions to higher order neighbours l' are suggested, while in reality it is possible that $f(x)$ points towards $con(l')$ everywhere on $con(l)$, while the trajectories of CS only become tangent to $con(l')$ everywhere on $con(l)$ and flow to a even higher order neighbour. Each situation of this kind is signaled by 'flag' in Algorithm 2, some preliminary sets L_i, $i = 1, \dots, n$ are constructed and later they are modified in a fixpoint manner.

If $s_i(l) = in$, then by Definition 5, in general there might exist points in $con(l)$ flowing to all neighbours in direction i, and therefore we let l'_i be any of $\{l_i - 1, l_i, l_i + 1\}$. In this case, it is possible that we add transitions in DS that do not correspond to trajectories of CS, ie, Equation (6) is satisfied in general with 'if'. However, this source of conservativeness is eliminated through refinement as described below.

After finding all sets L_i, since l can have transitions to its neighbours only, set L' is found by intersecting the cartesian product of sets L_i, $i = 1, \dots, n$ with the set of neighbours of l.

4.3 Refinement

For a given partition $con(L)$ in which all entries $s_i(l)$, $i = 1, \dots, n$ in the signatures $s(l)$ of all states $l \in L$ are in the set $\{po, ne, ze\}$, $con(Post(I))$ cannot be shrunk by finer partitioning, for any $I \subset L$. More formally, if $\overline{con}(\bar{L})$ is a partition of $con(L)$ ($|\bar{L}| > |L|$) and $I \subset L$ and $\bar{I} \subset \bar{L}$ are so that $|\bar{I}| > |I|$ and $\overline{con}(\bar{I})$ is a partition of $con(I)$, then

$$\overline{con}(\text{Post}(\bar{I})) = con(\text{Post}(I)) \tag{24}$$

Therefore, it does not make sense to partition such quotients.

Algorithm 2 Transitions for a discrete state l

```
for $i = 1, \ldots, n$ do
  $L_i = \phi$
  flag($i$)=0
  if $l_i$ is odd then
    $L_i := L_i \cup \{l_i\}$
  end if
  if $s_i(l) \in \{po, ne, ze\}$ then
    $L_i := L_i \cup \{l_i + eval(s_i(l))\}$
  if $(eval(s_i(l)) = 0) \wedge (l_i$ is even$) \wedge (m \leq n-2)$ then
    flag($i$)=1
  end if
  else
    $L_i := L_i \cup \{l_i - 1, l_i, l_i + 1\}$
  end if
    $L_i := L_i \cap \{0, 1, \ldots, 2n_i\}$
  end for
    $L' := (L_1 \times L_2 \times \ldots \times L_n) \cap (\mathcal{L}(l) \cup \mathcal{H}(l))$
  $S_{prev} = 0$
  while $\sum_{i=1}^{n}$ flag($i$)$\notin\{0, S_{prev}\}$ do
    $S_{prev} = \sum_{i=1}^{n}$ flag($i$)
    for $i = 1, \ldots, n$ do
      if flag($i$)=1 then
        let $L^i_{flag} = \{l' \in L' \mid l'_i = l_i\}$
        $L_i := L_i \setminus \{l_i\}$
    for all $l' \in L^i_{flag}$ do
      if $s_i(l') \in \{po, ne, ze\}$ then
        $L_i := L_i \cup \{l_i + eval(s_i(l'))\}$
      else
        $L_i := L_i \cup \{l_i - 1, l_i, l_i + 1\}$
      end if
      if $(s_i(l') \neq ze) \vee (ord(l') > n - 2)$ then
        flag($i$)=0
      end if
    end for
  end if
    $L_i := L_i \cap \{0, 1, \ldots, 2n_i\}$
  end for
    $L' := (L_1 \times L_2 \times \ldots \times L_n) \cap (\mathcal{L}(l) \cup \mathcal{H}(l))$
  end while
```

On the contrary, if for a given partition $con(L)$ there exists a state $l \in L$ and a signature entry $s_i(l) = in$, we can show that proper partitioning produces a discrete quotient $\overline{DS} = (\bar{L}, \bar{T})$ that refines $DS = (L, T)$ in the sense of Definition 3. Therefore, 'smaller' over-approximations of the reach set can be constructed (guaranteed strictly smaller if (iii)′ in Proposition 1 holds). The explanation is as follows.

Rectangles of order 0 (vertices) always have well-defined signature entries $s_i(l)$ in all directions $i = 1, \ldots, n$. A rectangle l of order 1 from DS has indefinite signature entry $s_i(l)$ if $con(l)$ intersects the surface defined by $f_i(x) = 0$ in X. Let l_j be the only odd entry in l. Since f is multi-affine and $con(l)$ is parallel with axis Ox_j, the intersection is a point whose coordinates can be easily computed by solving a linear equation with respect to x_j. Let the solution be denoted by $\tilde{x}_j$. By splitting the current partition DS with respect to the hyperplane $x_j = \tilde{x}_j$, we obtain a new partition $\overline{DS}$. In this partition there are three states, $\bar{l}'$, $\bar{l}''$, $\bar{l}''' \in \bar{L}$, that refine state l from the previous partition ($\overline{con}(\bar{l}') \cup \overline{con}(\bar{l}'') \cup \overline{con}(\bar{l}''')$ is a partition of $con(l)$). Thus, (i) from Definition 3 is satisfied by the considered state l. All states $\bar{l}'$, $\bar{l}''$, $\bar{l}'''$ have well defined signature entry of index i, and by applying Algorithm 2 to these states, the discrete transitions will exactly correspond to continuous trajectories in direction i. Because $s_i(l) = in$, transitions from l were conservative, in the sense that l could transit in any of its neighbours on direction i. This is not true for $\bar{l}'$, $\bar{l}''$, $\bar{l}'''$, so on one hand (ii) from Definition 3 is satisfied, *ie*, none of the states $\bar{l}'$, $\bar{l}''$, $\bar{l}'''$ can have transitions to regions of space that were not captured by transitions from l in the coarser quotient. On other hand, (iii) is satisfied because $\bar{l}'$, $\bar{l}''$, $\bar{l}'''$ do not have transitions to all their neighbours on direction i, as l did. Condition (iii)′ of Proposition 1 cannot be verified by considering only $\bar{l}'$, $\bar{l}''$, $\bar{l}'''$ and their neighbours, so the strict shrinking of over-approximation $con(Post(I), \forall I \subset L$ can be observed only after the finer quotient $\overline{DS}$ is constructed and $\overline{con}(Post(\bar{I}))$ is computed. Following the same reasoning, (i) and (ii) are satisfied for all states of DS and $\overline{DS}$, while (iii) is true only for states of $\overline{DS}$ with well defined signature entry for an index for which the corresponding refined states of DS had indefinite signature entry.

A finer quotient $\overline{DS}$ of DS can be found by using Algorithm 3, which computes all possible intersections in X between all surfaces $f_i = 0$, $i = 1, \ldots, n$ and all $con(l)$, where l is a state of order 1 in DS. Note that rectangles with order greater than 1 are not split even if they have an indefinite signature on a certain direction and all their neighbours of order 1 have well-defined signatures on the same direction. From the tests we performed, we observed that if X contains no common points of any two surfaces $f_i = 0$ and $f_j = 0$, $i, j = 1, \ldots, n$, $i \neq j$, then, after a finite number of iterations, Algorithm 3 will not produce new points. In this case, all surfaces $f_i = 0$, $i = 1, \ldots, n$ will eventually have non-empty intersections only with some rectangles of order 0 and of order greater than 1.

Algorithm 3 Refinement

```
for all first order states $l = (l_1, \dots, l_n) \in \mathbf{L}$ do
    let $l_k$ be the only odd entry in $l$
    for $i = 1, \dots, n$ do
      if $s_i(l) = in$ then
        solve $f_i(\theta^1_{l_1/2}, \dots, \theta^k_{new}, \dots, \theta^n_{l_n/2}) = 0$ for $\theta^k_{new}$
        add $\theta^k_{new}$ to vector $\theta^k$
      end if
    end for
  end for
  for $i = 1, \dots, n$ ] do
    sort elements of $\theta^i$ ascending
end for
```

4.4 Safety verification algorithm

We now collect all the results in this paper in the form of an iterative algorithm for providing a solution to Problem 2. Algorithm 4 starts with an initial rectangular partition determined by the sets I and F. A discrete quotient DS is constructed as described in Sections 4.1 and 4.2 and Post(I) is calculated. If Post$(I) \bigcap F = \emptyset$, then assertion (4) is true, ie, $con(F)$ cannot be reached by the continuous system initialized in $con(I)$. If Post$(I) \bigcap F \neq \emptyset$, then refinement is undertaken as described in Section 4.3. The algorithm is stopped if any of the following occurs: the safety property is satisfied, the refinement is finished, a partitioning precision is reached, or a user defined maximum number of iterations is exceeded. In the case when the algorithm is stopped and the safety property is not verified, it returns a sub-region $con(S_F)$ of $con(F)$ which is safe for CS if initialised in $con(I)$. If only an over-approximation of the solution to Problem 1 is desired, then Algorithm 4 can be run with $F = L$ ($con(F) = X$), where the initial partition L is induced by I only.

We used standard techniques from graph theory to calculate Post(I) for a discrete quotient DS. Specifically, we first assigned to every state $l \in L$ a unique number from $\{1, \dots, |L|\}$ by defining a map $node : L \to \{1, \dots, |L|\}$. We then defined an adjacency matrix $A \in \{0,1\}^{|L| \times |L|}$ with the property that $A(node(l), node(l')) = 1$ if there is a transition (in one step) from state l to l' and $A(node(l), node(l')) = 0$ otherwise, $l, l' \in L$. We find Post(l) for a state $l \in L$ by using the following property of the adjacency matrix: if $A^k(node(l), node(l')) = 1$, then state l' can be reached in k steps from state l. The maximum power k is $|L| - 1$, because there are only $|L|$ nodes in the graph, so the longest path can have $|L| - 1$ transitions. Moreover, this search algorithm can be stopped when for a certain power k we do not obtain any new states that can be reached from l. In this framework, Post(l)=$\{l' | \exists k \in \{1, \dots, |L| - 1\}$ s.t. $A^k(node(l), node(l')) = 1\}$, and Post$(I) = \bigcup_{l \in I}$ Post(l).

Algorithm 4 Safety verification and safe region construction

```
Start with vectors $\theta^i$, $i = 1, \ldots, n$ induced by $I$ and $F$
  repeat
    Construct set $L$ of labels of all elements in partition created by $\theta^i$, $i = 1, \ldots, n$
    for all $l \in L$ do
      Run Algorithm 1 to determine the signatures $s(l)$
    end for
    for all $l \in L$ do
      Run Algorithm 2 to determine the transitions from $l$
    end for
    Construct Post($I$)
    Construct safe sub-region of $F$: $S_F = F \setminus (\mathrm{Post}(I) \cap F)$
    if $S_F = F$ then
      SAFETY IS VERIFIED; exit algorithm
    else
      Run Algorithm 3 to produce a refinement
      if (no new points are added in any of vectors $\theta^i$) $\vee$ (precision is reached) $\vee$ (maximum
      number of iterations is reached) then
        SAFETY CANNOT BE DECIDED; exit algorithm
      end if
    end if
  untilTRUE
```

On the connection between the solutions to Problems 1 and 2, note that, even if the over-approximation of $con(\mathrm{Post}(I))$ is guaranteed to strictly shrink, this does not necessarily imply that the safe sub-region $con(S_F)$ of $con(F)$ strictly grows. It is guaranteed not to shrink, but it might not grow if the refinement is made in a region of X which has empty intersection with $con(F)$ and/or the rectangles which are refined are not contained in a path from I to F in DS.

5. Case studies

We have developed a user-friendly software package for Reachability Analysis of Multi-Affine Systems (RAMAS) in Matlab, which is freely downloadable from http://iasi.bu.edu/software/. The program takes as input the dimension n, the closed rectangle X, the coefficients $c_{i_1,\ldots,i_n}$ of a multi-affine vector field f as in Equation (14), and the sets $con(I)$ and $con(F)$ given in terms of unions of open sub-rectangles of arbitrary order in X. According to Algorithm 4, it returns either a positive answer if there are no trajectories of the continuous system from $con(I)$ and $con(F)$, or a subset of $con(F)$ which is guaranteed to be safe with respect to $con(I)$. Even though our tries show that the algorithm works for $n = 7$ and even $n = 10$ (for only one iteration), in this paper we focus on a planar case ($n = 2$) so we can show illustrative pictures.

We first consider a non-linear multi-affine vector field (Case Study 1) (Figure 2). We then focus on three linear systems (ie, $\dot{x} = Ax$) (Case Studies 2, 3, 4), which are of course particular cases of multi-affine systems. The qualitative phase portraits for such planar linear systems are known, and reachability properties are almost intuitive. Applying our method to such systems gives us some idea on the conservativeness of our approach.

Case Study 1 (non-linear multi-affine system): Consider $X = [1.5, 3] \times [0.4, 2]$, $f = (f_1, f_2)$ with $f_1 = 2 - x_1 x_2$, and $f_2 = 1 + x_2 - x_1 x_2$. The initial set is $con(I) = [1.5, 2.5] \times \{0.4\}$, which can be written as the union of two zero-order open rectangles $\{1.5, 0.4\}$, $\{2.5, 0.4\}$ and one first-order open rectangle $(1.5, 2.5) \times 0.4$. The final set is $con(F) = [1.5, 3] \times [0.8, 1.4]$, which in the initial partition can be seen as the union of six zero-order open rectangles, 7 first-order open rectangles, and two second-order open rectangles. In figure 2, we plot the vector field f everywhere in X and the two curves $f_1 = 0$ and $f_2 = 0$. Note that the two curves intersect inside $con(F)$. Therefore, the refinement procedure will not terminate. At each iteration, the algorithm will produce strictly shrinking over-approximations of Post($con(I)$) in X, which lead to strictly growing safe sub-regions in $con(F)$.

The results produced at different iterations are shown in Figure 3, where it can be seen that the safe region strictly increases with the number of refinement iterations.

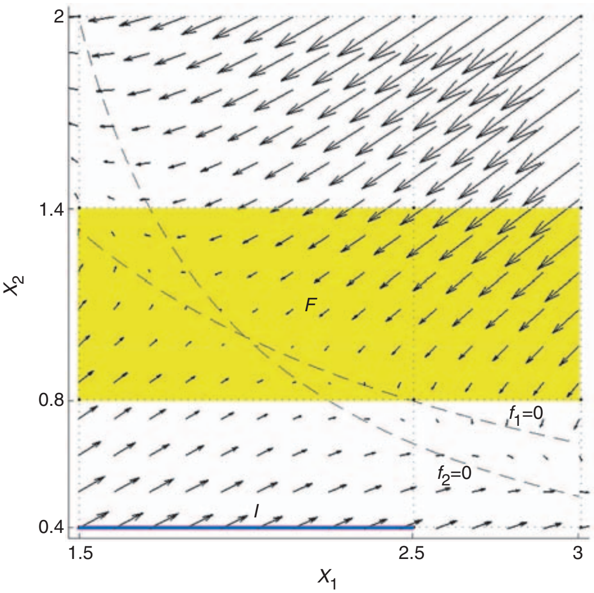

Figure 2 Case Study 1: multi-affine vector field $f = (f_1, f_2)$, $f_1 = 2 - x_1 x_2$, $f_2 = 1 + x_2 - x_1 x_2$ on X = [1.5, 3] × [0.4, 2], initial set con(I) = [1.5, 2.5] × {0.4}, final set con(F) = [1.5, 3] × [0.8, 1.4], and initial partition induced by initial and final sets

Case Study 2 (linear system, stable node): Consider the planar linear system $\dot{x} = f(x)$ with $f_1 = -1.5x_1 - 0.5x_2$, $f_2 = -0.5x_1 - 1.5x_2$ in rectangle $X = [-3, 4] \times [-3, 2]$. The origin is a (globally asymptotically) stable node for the system. The vector field is plotted in Figure 4(a), together with the lines $f_1 = 0$ and $f_2 = 0$ and the initial set $con(I) = [-2, 2] \times \{-2\}$. Figure 4(b) shows an over-approximation of the set reached from $con(I)$ (white region) and a safe set (green region). The straight dashed lines show the directions of the eigenvectors. We also plotted the trajectories starting from the extremities of $con(I)$. Since the system is linear, it is known that the closed segment $con(I)$ will remain a closed segment while flowing along the vector field. Therefore, the set reached from $con(I)$ roughly looks like the area between

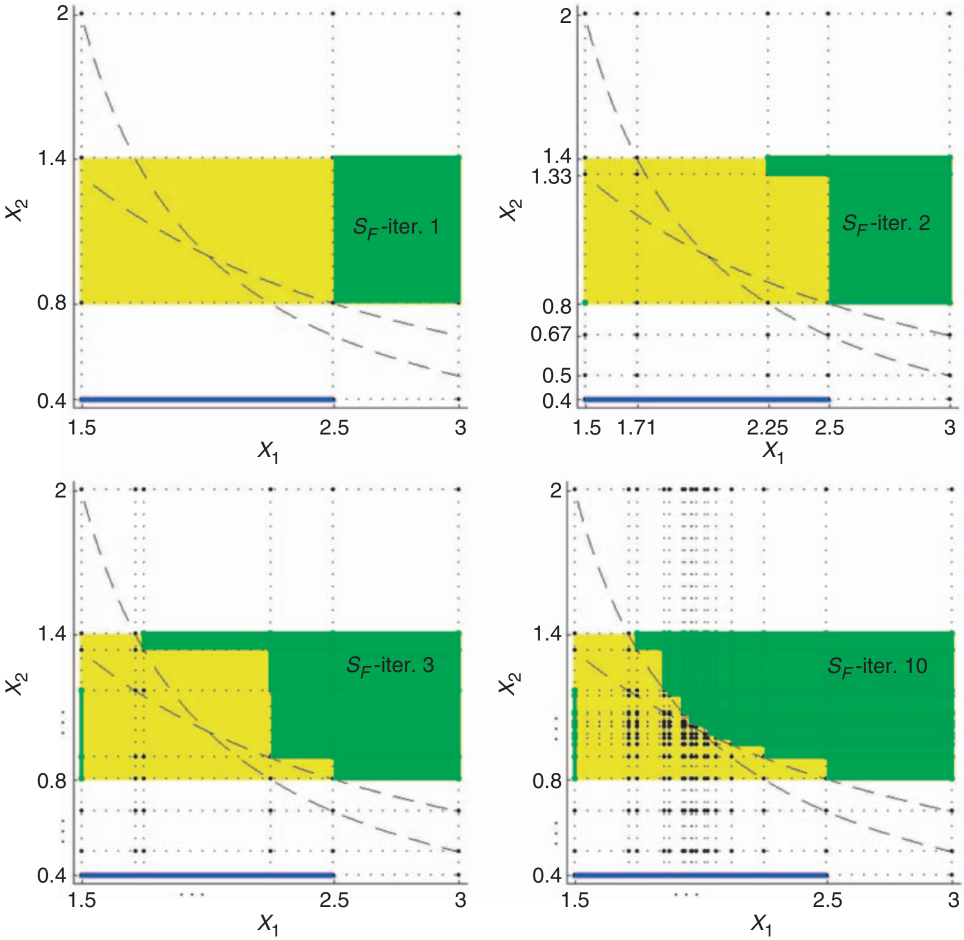

Figure 3 Case Study 1: iterations 1, 2, 3, and 10 from Algorithm 4. The growing green area (darker for black and white) represents the safe sub-region $con(S_F)$ of $con(F)$.

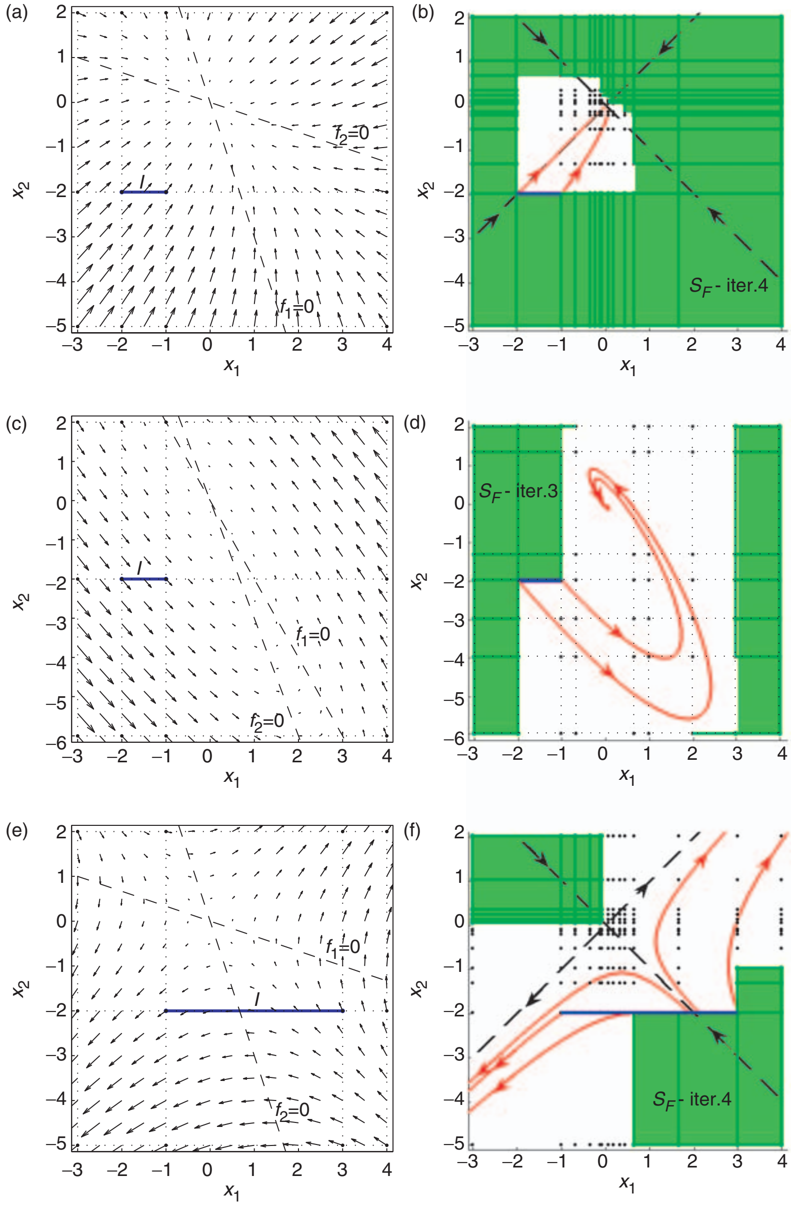

Figure 4 Reachability analysis for linear vector fields: (a), (c), and (e) show vector fields for which the origin is a stable node, stable focus, and unstable node, respectively. The equations of the straight lines are $f_1=0$ and $f_2=0$. The horizontal line segment is the initial set. The results of the reachability algorithm are shown in (b), (d), and (f), respectively. The green regions (darker for black and white) are safe sets, while the white regions are over-approximations of reachable sets

the trajectories of the extremities of $con(I)$, as shown in Figure 4(b). Our method, however, returns the white region in four iterations. More iterations will not shrink the white region dramatically – it will only remove small white chunks North-East from the origin.

Case Study 3 (linear system, stable focus): The difference between this case and Case Study 2 is that the vector field (shown in Figure 4(c)) is $f_1 = -2x_1 - x_2$, $f_2 = 3x_1 + x_2$, for which the origin is a stable focus. As it can be seen in Figure 4(d), the conservativeness of our method is more obvious in this case. Also, in this case the refinement algorithm terminates in three iterations.

Case Study 4 (linear system, unstable node): Consider the same rectangular region and the planar linear vector field $f_1 = 0.5x_1 + 1.5x_2$, $f_2 = 1.5x_1 + 0.5x_2$, for which the origin is an unstable node (saddle). The vector field is plotted in Figure 4(e), together with the initial set $con(I) = [-1, 3] \times \{-2\}$ and the two lines $f_1 = 0$ and $f_2 = 0$, which intersect at the origin. The over-approximation of $Post(con(I))$ calculated in four iterations by our method is shown as the white region in Figure 4(f), together with the eigenvectors and some illustrative trajectories. It can be seen that in this case our results are not very conservative. Note that the refinement does not terminate – it continues in a small region North-West from the origin. However, the result does not change significantly with the number of iterations.

As a conclusion to Case Studies 2, 3, and 4, our method produces conservative results when the trajectories loop around an equilibrium. This is in accordance with Proposition 1.

6. Complexity issues

Following the notation introduced at the beginning of Section 4, for a rectangle $X \in \mathbb{R}^n$, the discrete quotient DS has $L = \prod_{i=1}^{n}(2n_i + 1)$ states, where n_i is the number of intervals in the Ox_i direction, $i = 1, \ldots, n$. The total number of vertices (zero-order rectangles) is $\prod_{i=1}^{n}(n_i + 1)$. In order to find the transitions of DS, we have to compute the signature of every state $l \in L$. This can be done by running Algorithm 1 $|L|$ times, which implies that the sign of the vector field at each vertex will be evaluated more than once. This can be inefficient, since the vector field is known by the 2^n coefficients $c_{i_1,\ldots,i_n} \in \mathbb{R}^n$ from Equation (14). Another way to compute signatures would be to store the sign of the vector field at every vertex (so there would be only $\prod_{i=1}^{n}(n_i + 1)$ computations of vector fields), but this would increase the necessary storage space, which, as shown below, might be worse than the increase of computation time.

Recall that each state of DS has associated an n-uple as a label and another n-uple as a signature, beside its transitions to its neighbours. For example, if $n = 2$, $n_1 = 2$, $n_2 = 3$ (Case Study 1), at the first iteration of Algorithm 4, there are $|L| = 35$ open rectangles and we have to evaluate the vector field (defined by $n \cdot 2^n = 8$ real values) at 12 vertices.

In a three-dimensional case with $n = 3$, $n_1 = 2$, $n_2 = 3$, $n_3 = 3$, there are $|L| = 245$ states and 48 vertices, whereas for $n = 10$, $n_i = 2$, $i = 1, \ldots, 5$, $n_i = 3$, $i = 6, \ldots, 10$, $|L| = 0.5 \cdot 10^8$, the number of vertices is $2 \cdot 10^5$ and the vector field coefficients consists of $10 \cdot 2^{10}$ real values. A refinement of the initial discrete quotient DS will usually lead to a much greater number of states in $\overline{DS}$ refining DS. This number of states cannot be accurately estimated, since it depends on the specific values of the vector field coefficients. For example, in Case Study 1, the number of rectangles at the 10th iteration is 1221. However, in large dimensional cases, Algorithm 4 can be run usually for only one iteration, due to restrictions of the necessary storage space. For example, in the $n = 10$ example mentioned at the beginning of Section 5, with each real variable represented in double precision (8B), the necessary space for storing labels and signatures of rectangles (without transitions) was greater than 7GB!

On running time for the two-dimensional cases presented above, the computation took between 5–10 seconds on an 2.66G Hz IBM Pentium IV with 1 GB of RAM. For the seven-dimensional case mentioned at the beginning of this section, the first iteration took 3 minutes, the second about 20 minutes, and the third about 2 hours. For the ten-dimensional case, the first iteration took about 2 hours, and then the computer ran out of memory because of the state explosion problem in the refined discrete quotient.

7. Conclusion and future work

In this paper we developed a computationally inexpensive method for reachability analysis of multi-affine continuous systems. The method is based on rectangular partitions and iterative constructions of discrete quotients that provide an over-approximation of the reach set of the continuous system, with guaranteed decrease of conservativeness. While falling into the more general framework of Tiwari and Khanna (2002), where general polynomials are used for partition and polynomial vector fields are allowed, this paper shows that if more structure is allowed, then reachabiltity and safety verification questions can be answered with much less computation. Future work includes extensions to systems with polynomial dynamics, development of algorithms to check specifications given in terms of linear temporal logic, and applications to mathematical models found in areas such as biochemistry and control of aircraft and under-water vehicles.

Acknowledgements

This work was partially supported by NSF CAREER 0447721 and NSF 0410514 at Boston University.

References

Alur, R., Courcoubetis, C., Henzinger, T.A. and **Ho, P.H.** 1993: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Grossman, R.L., Nerode, A., Ravn, A.P. and Rischel, H., editors, *Lecture notes in computer science*, Vol. 736. Springer-Verlag, New York, 209–29.

Alur, R., Dang, T. and **Ivancic, F.** 2002: Predicate abstraction for reachability analysis of hybrid systems. *ACM Transactions on Embedded Computing Systems* 5(1), 152–99.

Alur, R. and **Dill, D.L.** 1994: A theory of timed automata. *Theoretical Computer Science* 126, 183–235.

Batt, G., Belta, C. and **Weiss, R.** 2008: Temporal logic analysis of gene networks under parameter uncertainty. *IEEE Transactions on Circuits and Systems and IEEE Transactions on Automatic Control, Joint Special Issue on Systems Biology* 53, 215–29.

Belta, C. and **Habets, L.C.G.J.M.** 2006: Control of a class of nonlinear systems on rectangles. *IEEE Transactions on Automatic Control* 51(11), 1749–59.

Ghosh, R. Tiwari, A. and **Tomlin, C.** 2003: Automated symbolic reachability analysis; with application to delta-notch signaling automata. In Maler, O. and Pnueli, A., editors, *Lecture notes in computer science*, Vol. 2623. Springer-Verlag, New York, 233–48.

Habets, L.C.G.J.M. and **van Schuppen, J.H.** 2004: A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica* 40, 21–35.

Haghverdi, E. Tabuada, P. and **Pappas, G.** 2005: Bisimulation relations for dynamical, control, and hybrid systems. *Theoretical Computer Science* 342(2–3), 229–61.

Henzinger, T.A., Kopke, P.W., Puri, A. and **Varaiya, P.** 1998: What is decidable about hybrid automata? *Journal of Computer System and Sciences* 57, 94–124.

Kloetzer, M. and **Belta, C.** 2008: A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control* 53(1), 287–97.

Lafferriere, G., Pappas, G.J. and **Sastry, S.** 2000: O-minimal hybrid systems. *Mathametics of Control, Signals, and Systems* 13(1), 1–21.

Lotka, A. 1925: *Elements of physical biology*. Dover Publications, Inc., New York.

Milner, R. 1989: *Communication and concurrency* Prentice-Hall, Englewood CliDs, NJ.

Nicolin, X., Olivero, A., Sifakis, J. and **Yovine, S.** 1993: An approach to the description and analysis of hybrid automata. In Grossman, R.L., Nerode, A., Ravn, A.P. and Rischel, H. editors, *Lecture notes in computer science*, Vol. 736. Springer-Verlag, New York, 149–78.

Pappas, G.J. 2003: Bisimilar linear systems. *Automatica* 39(12), 2035–47.

Park, D.M.R. 1981: Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Lecture notes in computer science*, Vol. 104. Springer-Verlag, London, UK, 167–83.

Puri, A. and **Varaiya, P.** 1994: Decidability of hybrid systems with rectangular differential inclusions. In David L. Dill, editor, *Lecture notes in computer science*, Vol. 818. Springer-Verlag, London, UK, 95–104.

Tiwari, A. and **Khanna, G.** 2002: Series of abstractions for hybrid automata. In Tomlin, C. and Greenstreet, M.R., editors, *Lecture notes in computer science*, Vol. 2289. Springer-Verlag, London, UK, 465–78.

Volterra, V. 1926: Fluctuations in the abundance of a species considered mathematically. *Nature* 118, 558–60.