Discrete Abstraction of Multiaffine Systems

Hui Kong[1], Ezio Bartocci[2], Sergiy Bogomolov[1], Radu Grosu[2]
Thomas A. Henzinger[1], Yu Jiang[3], and Christian Schilling[4]

[1] Institute of Science and Technology Austria, Klosterneuburg, Austria
[2] Vienna University of Technology, Vienna, Austria
[3] University of Illinois at Urbana-Champaign, IL, USA
[4] University of Freiburg, Germany

Abstract. Many biological systems can be modeled as multiaffine hybrid systems. Due to the nonlinearity of multiaffine systems, it is difficult to verify their properties of interest directly. A common strategy to tackle this problem is to construct and analyze a discrete overapproximation of the original system. However, the conservativeness of a discrete abstraction significantly determines the level of confidence we can have in the properties of the original system. In this paper, in order to reduce the conservativeness of a discrete abstraction, we propose a new method based on a sufficient and necessary decision condition for computing discrete transitions between states in the abstract system. We assume the state space partition of a multiaffine system to be based on a set of multivariate polynomials. Hence, a rectangular partition defined in terms of polynomials of the form $(x_i - c)$ is just a simple case of multivariate polynomial partition, and the new decision condition applies naturally. We analyze and demonstrate the improvement of our method over the existing methods using some examples.

Keywords: multiaffine system, hybrid system, discrete abstraction, state space partition, Gröbner basis

1 Introduction

A biological system is a complex network of biologically relevant entities. The analysis of complex biological systems can significantly benefit from the theory and techniques developed in the area of hybrid systems [15, 5, 4, 7, 6, 16, 12, 11, 22, 21]. The class of multiaffine hybrid systems [17, 9] is particularly suited to model and analyze a broad range of biological systems. However, due to the nonlinearity of multiaffine systems, it is often difficult to verify their properties of interest directly. A common strategy to tackle this problem is based on the idea of hybridization. In this setting, a given system is replaced by an abstraction where the system state space is partitioned and the original nonlinear dynamics is replaced with a simpler one in each induced partition. The resulting abstraction can either keep some approximated version of continuous dynamics [3, 2] or reason in discrete terms only [19, 20, 18]. In the following, we consider discrete abstractions of hybrid systems.

The quality of a discrete abstraction of a multiaffine system depends closely on the partition scheme of the state space and the conservativeness of discrete transitions between abstract states. A simple idea to partition the state space is to use a set of hyperplanes that are perpendicular to coordinate axes [13, 8, 23], hence the resulting regions are a set of hyperrectangles. The benefits of rectangular partition can be described as follows: 1) vertices of the hyperrectangles can be easily obtained, 2) some properties can be applied to establish the discrete transitions between abstract states (e.g., Proposition 1). However, since a rectangular partition does not take into account the feature of the vector flow, it could be inefficient. To address this problem, in [1, 30, 29, 28], a set of polynomials was used for partitioning the continuous state space. The idea is that, given a set of polynomials $\Phi = \{\varphi_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}], i = 1, \ldots, K\}$, each $\varphi_i(\boldsymbol{x})$ can partition the state space into three parts: (1) $\{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi_i(\boldsymbol{x}) < 0\}$, (2) $\{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi_i(\boldsymbol{x}) = 0\}$ and (3) $\{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi_i(\boldsymbol{x}) > 0\}$. Thus, $|\Phi|$ polynomials altogether can partition the state space into at most $3^{|\Phi|}$ parts. Both of the aforementioned partition methods have to address an important issue: how to establish discrete transitions between the abstract states (i.e. partitioned regions)? A common decision condition used by the existing methods is that, a positive first-order Lie derivative of $\varphi_i(\boldsymbol{x})$ at some point $\boldsymbol{x}_\tau$ in the hypersurface $\varphi_i(\boldsymbol{x}) = 0$ suffices to prove a trajectory being able to reach the region of $\varphi_i(\boldsymbol{x}) > 0$ from the region of $\varphi_i(\boldsymbol{x}) < 0$ and vice versa. However, this conditional test sometimes fails (i.e. the first-order Lie derivative is 0) and an overapproximating transition relation has to be built.

In this paper, similar to [29, 28], the state space partition is assumed to be based on a set Φ of multivariate polynomials. To reduce the conservativeness of a discrete abstraction of multiaffine system, we propose a necessary and sufficient condition to build the discrete transitions between the abstract states. The idea is that, given a hypersurface $H_{\varphi_i} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi_i(\boldsymbol{x}) = 0\}$ with $\varphi_i \in \Phi$, a trajectory can pass through H_{φ_i} at some $x_\tau \in H_{\varphi_i}$ if and only if there exists an odd number N such that the N'th-order Lie derivative $\mathcal{L}_{\boldsymbol{f}}^N \varphi_i$ of φ_i is not equal to 0 and all the i'th-order Lie derivative (for $1 < i < N$) of φ_i is 0 at x_τ, and fortunately, there is an upper bound for N which is computable using the Gröbner basis. More specifically, the direction of the trajectory relative to H_{φ_i} at x_τ depends on the sign of $\mathcal{L}_{\boldsymbol{f}}^N \varphi_i$: if $\mathcal{L}_{\boldsymbol{f}}^N \varphi_i > 0$, the trajectory moves from the region of $\varphi_i(\boldsymbol{x}) < 0$ to the region of $\varphi_i(\boldsymbol{x}) > 0$, otherwise, the direction reverses. For any two adjacent abstract states $\boldsymbol{u}$ and $\boldsymbol{v}$ (see Definition 4 for *adjacency*), the problem of deciding the transition relation between them is equivalent to deciding whether there exists a trajectory that passes through the intersection of multiple hypersurfaces, which can be formalized as a first-order logic formula consisting of Lie derivatives of $\varphi_i(\boldsymbol{x})$ and can be solved by an SMT solver.

The main contribution of this paper includes: 1) we propose a necessary and sufficient condition for building discrete transitions between abstract states, 2) we design an algorithm for establishing the transition relations between abstract states, 3) we analyze and demonstrate the improvement of our method over the existing methods.

The rest of the paper is organized as follows. Section 2 gives the preliminaries required for the paper. Section 3 describes the partition scheme and the mapping between the abstract states and the original state regions. Section 4 proposes the method for establishing discrete transitions between abstract states. In Section 5, we analyze and demonstrate the improvement of our method over existing methods. Finally, we conclude in Section 6.

2 Preliminaries

In this section, we recall some backgrounds we need throughout the paper. We first clarify some notation conventions. If not specified otherwise, we decorate vectors in bold face (e.g., $\boldsymbol{x}$), we use the symbol $\mathbb{K}$ for a field, $\mathbb{R}$ for the real number field, $\mathbb{C}$ for the complex number field (which is algebraically closed) and $\mathbb{N}$ for the set of natural numbers, and all the polynomials involved are multivariate polynomials. In addition, for all the polynomials $p(\boldsymbol{x})$, we denote by $\boldsymbol{x}$ the vector composed of all the variables that occur in the polynomial. $|\Psi|$ denotes the cardinality of the set Ψ.

Definition 1. *[14] A subset $I \subseteq \mathbb{K}[\boldsymbol{x}]$ is called an ideal if*

1. $0 \in I$,
2. *if $p(\boldsymbol{x}), q(\boldsymbol{x}) \in I$, then $p(\boldsymbol{x}) + q(\boldsymbol{x}) \in I$,*
3. *if $p(\boldsymbol{x}) \in I$ and $g(\boldsymbol{x}) \in \mathbb{K}[\boldsymbol{x}]$, then $p(\boldsymbol{x})g(\boldsymbol{x}) \in I$.*

Definition 2. *[14] Let $g_1, ..., g_s$ be polynomials in $\mathbb{K}[\boldsymbol{x}]$, where $\mathbb{K}$ is a field. Then we set*

$$\langle g_1, ..., g_s \rangle = \{\sum_{i=1}^{s} h_i g_i : h_1, ..., h_s \in \mathbb{K}[\boldsymbol{x}]\} \tag{1}$$

It is easy to verify that $\langle g_1, ..., g_s \rangle$ is an ideal and it is called the ideal generated by $\{g_1, ..., g_s\}$.

For the denotative convenience, we need to first present the notation of Lie derivative, which is widely used in the discipline of differential geometry. For a given polynomial $\varphi \in \mathbb{K}[\boldsymbol{x}]$ and a continuous system $\dot{\boldsymbol{x}} = \boldsymbol{f}$ (where $\boldsymbol{f} = (f_1, ..., f_n)$), the high-order Lie derivative of φ is defined as follows.

$$\mathcal{L}_{\boldsymbol{f}}^{k}\varphi \triangleq \begin{cases} \varphi, & k = 0 \\ \sum_{i=1}^{n} \frac{\partial \mathcal{L}_{\boldsymbol{f}}^{k-1}\varphi}{\partial x_i} \cdot f_i, & k \geq 1 \end{cases}$$

Essentially, the k'th-order Lie derivative of φ is the k'th derivative of φ w.r.t. time t and hence reflects the change of φ over time t. Note that we just write $\mathcal{L}_{\boldsymbol{f}}^{1}\varphi$ as $\mathcal{L}_{\boldsymbol{f}}\varphi$.

Theorem 1. *[27] (Fixed Point Theorem) Given a polynomial $\varphi \in \mathbb{K}[\boldsymbol{x}]$, if, for some $M > 0, \mathcal{L}_{\boldsymbol{f}}^{M+1}\varphi \in \langle \mathcal{L}_{\boldsymbol{f}}^{0}\varphi, ..., \mathcal{L}_{\boldsymbol{f}}^{M}\varphi \rangle,$, then $\forall k \geq M + 1 : \mathcal{L}_{\boldsymbol{f}}^{k}\varphi \in \langle \mathcal{L}_{\boldsymbol{f}}^{0}\varphi, ..., \mathcal{L}_{\boldsymbol{f}}^{M}\varphi \rangle$.*

Proposition 1. *[10] Let $f : R \to \mathbb{R}^q$ be a multiaffine function on the n-dimensional rectangle $R \subset \mathbb{R}^n$ and $\boldsymbol{x} = (x_1, ..., x_n) \in R$, suppose F_i is the lowest-dimensional face of R that contains $\boldsymbol{x}$. Then, $f(\boldsymbol{x})$ is a convex combination of the values of f at the vertices of F_i.*

Definition 3 (Multiaffine System). *A multiaffine System is a tuple $M \stackrel{def}{=} \langle X, \boldsymbol{f}, Init \rangle$, where*

1. *X is the state space of the system M,*
2. *$\boldsymbol{f}$ is a Lipschitz multiaffine polynomial vector flow function, and*
3. *Init is the initial set described by a semialgebraic set.*

A multiaffine polynomial is a polynomial for which if we fix all the variables but one, the polynomial will become a linear polynomial.

3 State Space Partition and Abstract State Mapping

In this section, we introduce the partition scheme we adopt throughout the paper and the mapping of the original states to the abstract states.

3.1 State Space Partition

We assume to use a set of multivariate polynomials to partition the state space. There are several ways available to derive the set of polynomials [30, 29, 24]: 1) take the polynomials occurring in the vector flow function, the guards and the property to be verified, 2) compute the Lie derivatives of the existing polynomials iteratively, 3) discover algebraic invariants of the system. The details of these techniques are not covered in this paper.

The idea of polynomial-based partition is as follows. Given a set of polynomials $\Phi = \{\varphi_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}], i = 1, \ldots, K\}$, each $\varphi_i(\boldsymbol{x})$ can partition the state space into three parts: (1) $\{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi_i(\boldsymbol{x}) < 0\}$, (2) $\{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi_i(\boldsymbol{x}) = 0\}$ and (3) $\{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi_i(\boldsymbol{x}) > 0\}$. Thus, $|\Phi|$ polynomials altogether can partition the state space into at most $3^{|\Phi|}$ regions, each region of the partition can be represented as $\{x \in \mathbb{R}^n \mid \bigwedge \varphi_i(x) \sim_i 0\}$ with $\sim_i \in \{>, =, <\}$. In the following, we describe how to map these regions to abstract states.

3.2 Abstract states mapping

Given a multiaffine system $C = \langle X, Init, f \rangle$, a polynomial set $\Phi = \{\varphi_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}], i = 1, \ldots, K\}$ can partition a state space into at most 3^K regions and every state $\boldsymbol{x} \in X$ can be mapped to an abstract state in $V_\Phi \in 2^{\{-1,0,1\}^K}$ using the following abstraction function $Abst : X \mapsto V_\Phi$.

$$Abst(x) = (v_1(\boldsymbol{x}), \ldots, v_K(\boldsymbol{x})), \quad v_i(\boldsymbol{x}) \triangleq \begin{cases} 1, & \varphi_i(\boldsymbol{x}) > 0 \\ 0, & \varphi_i(\boldsymbol{x}) = 0 \\ -1, & \varphi_i(\boldsymbol{x}) < 0 \end{cases} \quad i = 1, \ldots, K$$

Conversely, we have the following concretization function $Con : V_\Phi \mapsto 2^X$ that maps an abstract state to a region of the original state space.

$$Con(v) = \{x \in X \mid x \models \bigwedge_{i=1}^{K} \pi(v_i)\}, \quad \pi(v_i) \triangleq \begin{cases} \varphi_i(\boldsymbol{x}) > 0, & v_i = 1 \\ \varphi_i(\boldsymbol{x}) = 0, & v_i = 0 \quad i = 1, \dots, K \\ \varphi_i(\boldsymbol{x}) < 0, & v_i = -1 \end{cases}$$

In the abstract state space, there could exist discrete transitions only between the abstract states whose corresponding regions in the original state space are **adjacent**, which we now define formally.

Definition 4. (**Adjacency**) *Given an abstract state space $V_\Phi \in 2^{\{-1,0,1\}^K}$, two abstract states $\boldsymbol{u}, \boldsymbol{v} \in V_\Phi$ are **adjacent**, denoted by **Adj(u,v)**, if and only if they satisfy the following formula with $\sim \in \{>, <\}$*

$$\begin{aligned} \boldsymbol{dim}(\boldsymbol{u}) \neq \boldsymbol{dim}(\boldsymbol{v}) \,\wedge \\ (\boldsymbol{dim}(\boldsymbol{u}) \sim \boldsymbol{dim}(\boldsymbol{v}) \implies \forall i = 1, \dots, K : u_i = v_i \vee |u_i| \sim |v_i|) \end{aligned} \quad (2)$$

*where $\boldsymbol{dim}(\boldsymbol{w}) = \sum_{i=1}^{K} |w_i|$ is called the **dimension** of an abstract state.*

Essentially, Definition 4 means that one of two adjacent abstract states can be obtained by setting some of the nonzero components of the other state to zero. The definition is reasonable because any trajectory $\boldsymbol{x}(t)$ cannot get from the region of $\varphi_i(\boldsymbol{x}) < 0$ to the region of $\varphi_i(\boldsymbol{x}) > 0$ without crossing the hypersurface $\varphi_i(\boldsymbol{x}) = 0$.

Definition 5. (**Discrete Abstraction**) *Given a multiaffine system $C = \langle X, f, Init \rangle$ and a polynomial set $\Phi = \{\varphi_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}], i = 1, \dots, K\}$, a discrete abstraction of C w.r.t. Φ is the transition system $C_\Phi = \langle V_\Phi, T_\Phi, Init_\Phi \rangle$, where*

- *$V_\Phi \in 2^{\{-1,0,1\}^K}$ is the abstract state space;*
- *$T_\Phi \in 2^{V_\Phi \times V_\Phi}$ is the set of discrete transitions such that $(\boldsymbol{u}, \boldsymbol{v}) \in T_\Phi$ if and only if there exists a trajectory $\boldsymbol{x}(t)$ and $t_1, t_2 \in \mathbb{R}_{\geq 0}$ and $t_2 > t_1$ satisfying: 1) $\boldsymbol{x}(t_1) \in Con(\boldsymbol{u})$, 2) $\boldsymbol{x}(t_2) \in Con(\boldsymbol{v})$, 3) $\forall t \in (t_1, t_2) : \boldsymbol{x}(t) \in Con(\boldsymbol{u}) \cup Con(\boldsymbol{v})$;*
- *$Init_\Phi = \{\boldsymbol{v} \in V_\Phi \mid \exists \boldsymbol{x} \in Init : \boldsymbol{x} \in Con(\boldsymbol{v})\}$ is the initial set.*

A discrete abstraction is an overapproximation of the original system. Given a partition, to construct a precise discrete abstraction, the key point is to make the set T_Φ of discrete transitions as small as possible. The technique for this purpose is presented in the following section.

4 Establishment of Discrete Transitions

In this section, we introduce how to establish the discrete transitions between the abstract states.

4.1 A necessary and sufficient condition

Suppose we have a polynomial set $\Phi = \{\varphi_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}], i = 1, \ldots, K\}$ for the partition, according to Definition 5, for a given pair of abstract states $\boldsymbol{u}$ and $\boldsymbol{v}$. There is a discrete transition from $\boldsymbol{u}$ to $\boldsymbol{v}$ if and only if $\boldsymbol{u}$ and $\boldsymbol{v}$ are adjacent and there exists a trajectory that reaches $Con(\boldsymbol{v})$ from $Con(\boldsymbol{u})$. Assume $\boldsymbol{u} = (u_1, \cdots, u_K)$ and $\boldsymbol{v} = (v_1, \cdots, v_K)$ are adjacent and let $D_{u,v} = \{\varphi_k(\boldsymbol{x}) = 0 \mid u_k \neq v_k, \ k = 1, \ldots, K\}$, then the original problem is equivalent to deciding whether there exists a trajectory passing through the intersection of all the hypersurfaces in $D_{u,c}$. By the following proposition, we first address the issue of deciding whether there exists a trajectory passing through a single hypersurface.

Theorem 2. *A continuous system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ can pass through a hypersurface $H = \{\boldsymbol{x} \in \mathbb{R}^n \mid \varphi(\boldsymbol{x}) = 0, \varphi(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]\}$, i.e.*

$$\exists \boldsymbol{x}(t) \in \{\boldsymbol{x}(t) \mid \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}(0) \in I_0\} : \exists \tau > 0, \epsilon > 0 : \varphi(\boldsymbol{x}(\tau)) = 0$$
$$\wedge \ (\forall t_1 \in (\tau - \epsilon, \tau) : \forall t_2 \in (\tau, \tau + \epsilon) : \varphi(\boldsymbol{x}(t_1))\varphi(\boldsymbol{x}(t_2)) < 0) \tag{3}$$

iff the formula

$$\exists \boldsymbol{x}_\tau \in H : \exists N = 2k - 1 : \bigwedge_{j=1}^{N-1} \mathcal{L}_{\boldsymbol{f}}^j \varphi = 0 \wedge \mathcal{L}_{\boldsymbol{f}}^N \varphi \neq 0 \tag{4}$$

holds, where I_0 is the set of initial states and $k \in \mathbb{N}$. Moreover, if $\mathcal{L}_{\boldsymbol{f}}^N \varphi > 0$, the direction of the trajectory points from the region of $\varphi(\boldsymbol{x}) < 0$ to the region of $\varphi(\boldsymbol{x}) > 0$, and otherwise, the direction reverses.

Proof. The Taylor expansion of $\varphi(\boldsymbol{x}(t))$ at time τ is as follows.

$$\varphi(\boldsymbol{x}(t)) = \varphi(\boldsymbol{x}(\tau)) + \sum_{k=1}^{\infty} \frac{1}{n!}(\mathcal{L}_{\boldsymbol{f}}^n \varphi)(t - \tau)^n \tag{5}$$

$(4) \Rightarrow (3)$: By applying the condition (4) to the Taylor expansion (5), we can derive that $\varphi(\boldsymbol{x}(t)) = \frac{1}{N!}(\mathcal{L}_{\boldsymbol{f}}^N \varphi)(t - \tau)^N$ for some trajectory $\boldsymbol{x}(t)$ with $\boldsymbol{x}(\tau) = \boldsymbol{x}_\tau$. Since $N = 2k - 1$ is an odd number, there must exist a real $\epsilon > 0$ such that,

1. if $\mathcal{L}_{\boldsymbol{f}}^N \varphi > 0$, then $\forall t_1 \in (\tau - \epsilon, \tau) : \varphi(\boldsymbol{x}(t_1)) < 0$ and $\forall t_2 \in (\tau, \tau + \epsilon) : \varphi(\boldsymbol{x}(t_2)) > 0$, or
2. if $\mathcal{L}_{\boldsymbol{f}}^N \varphi < 0$, then $\forall t_1 \in (\tau - \epsilon, \tau) : \varphi(\boldsymbol{x}(t_1)) > 0$ and $\forall t_2 \in (\tau, \tau + \epsilon) : \varphi(\boldsymbol{x}(t_2)) < 0$.

Therefore, condition (3) holds.

$(3) \Rightarrow (4)$: We show this implication by contradiction. Let $M = \min\{j \geq 1 \mid \mathcal{L}_{\boldsymbol{f}}^j \varphi \neq 0\}$. By applying condition (3) to the Taylor expansion (5), we can derive that there exist a real $\epsilon > 0$ such that

$$\forall t \in (\tau - \epsilon, \tau + \epsilon) : \varphi(\boldsymbol{x}(t)) = \frac{1}{M!}(\mathcal{L}_{\boldsymbol{f}}^M \varphi)(t - \tau)^M + O((t - \tau)^{M+1}) \tag{6}$$

We assume the condition (4) does not hold, i.e. the following formula holds.

$$\forall \boldsymbol{x} \in H : \forall N = 2k - 1 : \bigvee_{j=1}^{N-1} \mathcal{L}_{\boldsymbol{f}}^{j} \varphi \neq 0 \vee \mathcal{L}_{\boldsymbol{f}}^{N} \varphi = 0 \tag{7}$$

If $M = +\infty$, according to the formula (6), we have $\forall t \in (\tau - \epsilon, \tau + \epsilon)$: $\varphi(\boldsymbol{x}(t)) = 0$, which contradicts the formula (3). If $M < \infty$, then,

1. if M is an even number, according to equation (6), we have $\forall t \in (\tau - \epsilon, \tau + \epsilon) \backslash \{\tau\} : \varphi(\boldsymbol{x}(t)) > 0$ when $\mathcal{L}_{\boldsymbol{f}}^{M} f_i > 0$, or $\forall t \in (\tau - \epsilon, \tau + \epsilon) \backslash \{\tau\} : \varphi(\boldsymbol{x}(t)) < 0$ when $\mathcal{L}_{\boldsymbol{f}}^{M} f_i < 0$, which contradicts the condition (3), or
2. if M is an odd number, it contradicts the condition (7).

Therefore, we have that $(3) \Rightarrow (4)$ holds. $\qquad\square$

Remark 1. The Formula (4) in Theorem 2 is a sufficient and necessary condition for deciding whether a system can pass through a hypersuface defined by a multivariate polynomial. In pratice, univariate polynomials (i.e. $x_i - c$) instead of multivariate ones are most widely used for partitioning for their simplicity, where the resulting partition is rectangular, hence, Theorem 2 applies naturally. Note that in this simplfied case,

- if $N = 1$, then Formula (4) simplifies to $\exists \boldsymbol{x}_\tau \in H : f_i(\boldsymbol{x}_\tau) \neq 0$ (where f_i is the i'th component of the vector flow function $\boldsymbol{f}$), which is used by Proposition 3 in [8]. This is the most intuitive way for a trajectory to pass through the hyperplane.
- if $N > 1$, which means $f_i(p) = 0$ and the vector field of a system is tangent to the hyperplane at $\boldsymbol{x}_\tau \in H$, the system is still capable of crossing H for having one of the odd-order Lie derivatives of $(x_i - c)$ be positive and all the other lower-order Lie derivatives vanish at $\boldsymbol{x}_\tau$.

In Theorem 2, higher-order Lie derivatives are used to characterize the relationship between a hypersurface and a vector flow. In fact, there are also other theories on hybrid systems which are based on higher-order Lie derivatives [25, 24, 26]. In [27], J. Liu et al. used higher-order Lie derivatives to describe a necessary and sufficient condition for a multivariate polynomial to be an inductive invariant for a continuous system, which needs to check all positive integers for the existence of a positive integer $N > 0$ such that the N'th-order Lie derivative is negative and all the i'th-order Lie derivative (for $i < N$) are equal to 0. In our case, we only need to check the existence of an odd number N such that the N'th-order Lie derivative is positive or negative, depending on the direction of the trajectory that we want to check at the boundary of $\{\boldsymbol{x} \in \mathbb{R} \mid \varphi(\boldsymbol{x}) \leq 0\}$ and all the other lower-order Lie derivatives are 0. Note that the $\varphi(\boldsymbol{x})$'s are not limited to univariate polynomials, that is, our partitions are not limited to rectangular regions.

One key point to apply Formula (4) is how to determine the constant N. Fortunately, there exists a computable upper bound M for N based on Gröbner

basis theory [27]. Since the continuous systems under consideration are assumed to be multiaffine, $\mathcal{L}_{\boldsymbol{f}}^j \varphi$ must be a polynomial in $\mathbb{R}[\boldsymbol{x}]$. According to Theorem 1, we have

$$N \leq \gamma = \min\{j \mid \mathcal{L}_{\boldsymbol{f}}^{j+1} \varphi \in \langle \mathcal{L}_{\boldsymbol{f}}^0 \varphi, ..., \mathcal{L}_{\boldsymbol{f}}^j \varphi \rangle\} \tag{8}$$

The principle for Formula (8) is trivial, since for every $k \geq \gamma$ there must exist some $h_0, ..., h_r \in \mathbb{R}[x]$ such that $\mathcal{L}_{\boldsymbol{f}}^k \varphi = \sum_{r=0}^{\gamma} h_r \mathcal{L}_{\boldsymbol{f}}^r \varphi$. If $N > \gamma$, $\mathcal{L}_{\boldsymbol{f}}^k \varphi$ must be 0 for all $k \geq 0$, which contradicts the fact that N satisfies Formula (4). The value of γ is computed iteratively by using the Gröbner basis. To compute the Gröbner basis, some powerful tool packages are available in popular mathematical softwares such as *Maple*. We implemented Algorithm 1 in Maple to compute γ. In Algorithm 1, R represents the high-order Lie derivative of φ, the function *GrobnerBasis* is used to compute the Gröbner basis G of $\langle \mathcal{L}_{\boldsymbol{f}}^0 \varphi, ..., \mathcal{L}_{\boldsymbol{f}}^{\gamma-1} \varphi \rangle$, and the function *NormForm* is used to compute the remainder of a polynomial w.r.t. a Gröbner basis. The iteration terminates if and when the remainder U is 0, which means $\mathcal{L}_{\boldsymbol{f}}^\gamma \varphi \in \langle G \rangle$.

Algorithm 1: Compute the constant γ for polynomial φ.

Data: $\boldsymbol{f} = [f_1, ..., f_n]$, φ
Result: γ

1   $R \leftarrow \varphi$;
2   $\gamma \leftarrow 0$;
3   $U \leftarrow R$;
4   $B \leftarrow \{R\}$;
5   **while** $U \neq 0$ **do**
6     $G \leftarrow GrobnerBasis(B)$;
7     $R \leftarrow \sum_{i=1}^n \frac{\partial R}{\partial x_i} f_i$;
8     $U \leftarrow NormForm(R, G)$;
9     $B \leftarrow B \cup \{R\}$;
10    $\gamma \leftarrow \gamma + 1$;
11 **end**

Based on Theorem 2, we further derive the following corollary for establishing a discrete transition between adjacent abstract states.

Corollary 1. *Given a continuous system $C = \langle X, \boldsymbol{f}, Init \rangle$ and a set $\Phi = \{\varphi_i(x) \in \mathbb{R}[x], i = 1, \cdots, K\}$ of real coefficient polynomials, let $C_\Phi = \langle V_\Phi, T_\Phi, Init_\Phi \rangle$ be the corresponding discrete abstraction, where V_Φ is the abstract state space and $T_\Phi \in 2^{V_\Phi \times V_\Phi}$ is the set of abstract transitions, and let $\boldsymbol{u}, \boldsymbol{v} \in V_\Phi$. Then there exists a discrete transition $\boldsymbol{e} = (\boldsymbol{u}, \boldsymbol{v}) \in T_\Phi$ if and only if $Adj(\boldsymbol{u}, \boldsymbol{v})$ and*

$$\exists \boldsymbol{x} \in Con(Min(\boldsymbol{u}, \boldsymbol{v})) : \forall i = 1, \cdots, K : u_i \neq v_i \implies$$

$$\exists N_i = 2m_i - 1 : \bigwedge_{j=1}^{N_i-1} \mathcal{L}_{\boldsymbol{f}}^j \varphi_i = 0 \bigwedge (v_i - u_i) \mathcal{L}_{\boldsymbol{f}}^{N_i} \varphi_i > 0 \tag{9}$$

where $Min(\boldsymbol{u}, \boldsymbol{v})$ returns the state of lower dimension and $m_i \in \mathbb{N}$.

Proof. By Theorem 2, we can easily prove that the corollary holds.

Remark 2. Here we give an intuitive explanation for Corollary 1. According to Definition 4, if there is a transition from $\boldsymbol{u}$ to $\boldsymbol{v}$, it must be one of the following two cases: 1) $\mathbf{dim}(\boldsymbol{u}) > \mathbf{dim}(\boldsymbol{v})$, then there exists a trajectory which reaches the intersection of the hypersurfaces $\{\varphi_i = 0 \mid u_i \neq v_i, i = 1, \cdots, K\}$, or 2) $\mathbf{dim}(\boldsymbol{u}) < \mathbf{dim}(\boldsymbol{v})$, then there exists a trajectory which escapes the intersection of the hypersurfaces. However, no matter in which case, we can decide the direction of the trajectory only by the signs of the higher-order Lie derivatives of $\{\varphi_i \mid u_i \neq v_i, i = 1, \cdots, K\}$ in the domain $Con(Min(\boldsymbol{u}, \boldsymbol{v}))$. Moreover, suppose $\mathbf{dim}(\boldsymbol{u}) > \mathbf{dim}(\boldsymbol{v})$; if there is a transition $(\boldsymbol{u}, \boldsymbol{v})$ or $(\boldsymbol{v}, \boldsymbol{u})$, it is easy to show that there is also a transition $(\boldsymbol{v}, -\boldsymbol{u})$ or $(-\boldsymbol{u}, \boldsymbol{v})$ correspondingly.

Now, we use an example to demonstrate the application of Corollary 2 to establishing discrete transitions.

Example 1. Consider the following multiaffine system.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{4(-x_2 x_3 + 12)}{13} - 2x_1 \\ 4(12 - x_1) - x_2 \\ 6(12 - x_1) - 4x_3 \end{bmatrix}$$

Let $\Phi = \{\varphi_1(\boldsymbol{x}) = x_1 - 8, \varphi_2(\boldsymbol{x}) = x_2 - 8, \varphi_3(\boldsymbol{x}) = x_3 - 8\}$ be the set of polynomials for partitioning. We aim to decide the transition relation between the abstract states $\boldsymbol{u} = (-1, -1, 1)$ and $\boldsymbol{v} = (0, 0, 0)$, which corresponds to the regions $R_{\boldsymbol{u}} = \{\boldsymbol{x} \in \mathbb{R}^3 \mid \varphi_1(\boldsymbol{x}) < 0, \varphi_2(\boldsymbol{x}) < 0, \varphi_3(\boldsymbol{x}) > 0\}$ and $R_{\boldsymbol{v}} = \{\boldsymbol{x} \in \mathbb{R}^3 \mid \varphi_i(\boldsymbol{x}) = 0, i = 1, 2, 3\}$, respectively.

By applying Algorithm 1, we can get $\gamma_1 = 4$, $\gamma_2 = \gamma_3 = 3$. Since $\mathbf{dim}(\boldsymbol{u}) = 3 > 0 = \mathbf{dim}(\boldsymbol{v})$ and $Con(\boldsymbol{v})$ contains a single point $\boldsymbol{x}_\tau = (8, 8, 8)$, by computing the Lie derivatives of $\varphi_i(\boldsymbol{x})$ at $\boldsymbol{x}_\tau$, we get

$$\mathcal{L}_{\boldsymbol{f}} \varphi_1|_{\boldsymbol{x} = \boldsymbol{x}_\tau} = 0, \mathcal{L}_{\boldsymbol{f}}^2 \varphi_1|_{\boldsymbol{x} = \boldsymbol{x}_\tau} = 0, \mathcal{L}_{\boldsymbol{f}}^3 \varphi_1|_{\boldsymbol{x} = \boldsymbol{x}_\tau} = \frac{256}{13},$$

$$\mathcal{L}_{\boldsymbol{f}} \varphi_2|_{\boldsymbol{x} = \boldsymbol{x}_\tau} = 8, \mathcal{L}_{\boldsymbol{f}} \varphi_3|_{\boldsymbol{x} = \boldsymbol{x}_\tau} = -8$$

In this case, as shown in Figure 1, the trajectory of the system is tangent to the plane $x_1 - 8 = 0$ at $\boldsymbol{x}_\tau$. Since $\mathcal{L}_{\boldsymbol{f}} \varphi_1 = 0$ at $\boldsymbol{x}_\tau$, we cannot decide the direction of the transition between $\boldsymbol{u}$ and $\boldsymbol{v}$ by considering only the first-order Lie derivative like in [10]. However, according to Corollary 1, we can decide that there is a discrete transition from $\boldsymbol{u}$ to $\boldsymbol{v}$.

4.2 Computation method

In the previous section, we have introduced a necessary and sufficient condition for deciding whether there exists a discrete transition between two abstract
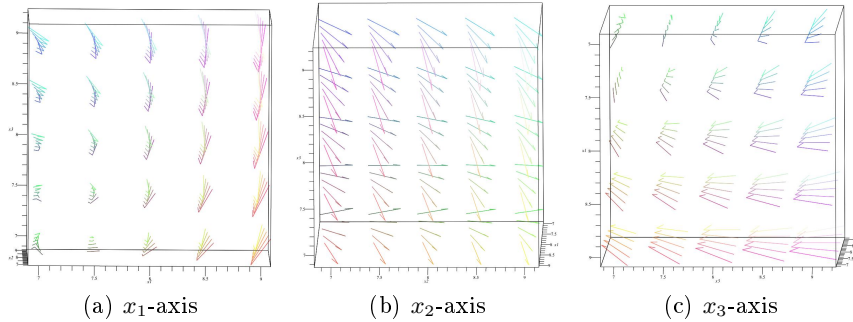
(a) x_1-axis　　　　(b) x_2-axis　　　　(c) x_3-axis

Fig. 1. Observing the vector fields of Example 1 from 3 different perspectives of view. The vector field is tangent to the plane $x_1 = 8$ at $\boldsymbol{x}_\tau = (8, 8, 8)$.

states. According to Corollary 1, the essential problem is to decide whether Formula (9) holds for two adjacent states $\boldsymbol{u}$ and $\boldsymbol{v}$, which involves deciding the satisfiability of a first-order logic formula built on a set of nonlinear polynomial equations and inequalities. To solve this problem, a number of SMT solvers are available, such as *Z3*, *SMT-RAT* etc.

Given two abstract states $\boldsymbol{u}$ and $\boldsymbol{v}$, we use Algorithm 2 to decide the transition relation between them. The idea of the algorithm is as follows. Given any two abstract states $\boldsymbol{u}$ and $\boldsymbol{v}$, if they are adjacent, we select the one with lower dimension, represented as $\boldsymbol{w} = Min(\boldsymbol{u}, \boldsymbol{v})$. As commented in Remark 2, the higher-order Lie derivatives of $\{\varphi_i(x) \mid u_i \neq v_i, i = 1, \cdots, K\}$ in the domain $Con(Min(\boldsymbol{u}, \boldsymbol{v}))$ suffice to determine the transition relation between $\boldsymbol{u}$ and $\boldsymbol{v}$. In Algorithm 2, the set $I_{\boldsymbol{u}, \boldsymbol{v}}$ collects the indices of the hypersurfaces where $\boldsymbol{u}$ differs from $\boldsymbol{v}$. Lines 5 to 8 are used to construct all the possible combinations of Lie derivatives of $\{\varphi_i \mid u_i \neq v_i, i = 1, \cdots K\}$ occurring in Formula (9). Then, we check with an SMT solver in line 10 if there exists a combination of the Lie derivatives which makes Formula (9) hold.

5　Discussion and Examples

In this section, we analyze and demonstrate the improvement of our method over the existing methods using some examples. Note that to be intuitive, we refer an abstract state to its region of concrete states in the following.

Theorem 2 presents a sufficient and necessary condition for deciding the existence of a trajectory passing through a hypersurface $\varphi(\boldsymbol{x}) = 0$ at a single point by a series of Lie derivatives of $\varphi(\boldsymbol{x})$. However, as indicated in Corollary 1, what we need to handle are mainly semialgebraic sets which are usually infinite. For example, given a rectangular partition for a 3-dimensional state space, the adjacent regions of a box consist of rectangles, edges, and vertices. To decide whether there exists a trajectory passing through an adjacent region, a general solution

---

Algorithm 2: Decide the direction of transition between $\boldsymbol{u}$ and $\boldsymbol{v}$.

Data: abstract states $\boldsymbol{u}, \boldsymbol{v}$; polynomial set $\varPhi = \{\varphi_i, i = 1, \cdots, K\}$;
the constant array $\varGamma = [\gamma_1, \cdots, \gamma_K]$ for all $\varphi_i \in \varPhi$
Result: $\boldsymbol{e}$: the transition relation

1 **if** *$\boldsymbol{u}$ and $\boldsymbol{v}$ are adjacent* **then**
2   select the lower-dimensional state $\boldsymbol{w} = Min(\boldsymbol{u}, \boldsymbol{v})$;
3 $P_w \leftarrow$ set of polynomial predicates defining $Con(\boldsymbol{w})$;
4 $I_{\boldsymbol{u},\boldsymbol{v}} \leftarrow \{i \mid u_i \neq v_i, i = 1, \cdots, K\}$;
5   **for** *each $i \in I_{\boldsymbol{u},\boldsymbol{v}}$* **do**
6 $\Psi_i \leftarrow \{\bigwedge_{j=1}^{N_i-1} \mathcal{L}_{\boldsymbol{f}}^j \varphi_i = 0 \bigwedge (v_i - u_i) \mathcal{L}_{\boldsymbol{f}}^{N_i} \varphi_i > 0 \mid 1 \leq N_i \leq \gamma_i, N_i \text{ is odd}\}$;
7   **end**
8 $\Psi \leftarrow \{(\psi_1, ..., \psi_M) \mid \psi_j \in \Psi_{i_j}, i_1 < \cdots < i_M, M = |I_{\boldsymbol{u},\boldsymbol{v}}|, i_j \in I_{\boldsymbol{u},\boldsymbol{v}}\}$;
9   **for** *each $(\psi_1, \cdots, \psi_M) \in \Psi$* **do**
10   **if** *$(\bigwedge_{p_i \in P_w} p_i) \wedge (\bigwedge_{j=1}^{M} \psi_j)$ is satisfiable by SMT solver* **then**
11 $\boldsymbol{e} \leftarrow (\boldsymbol{u}, \boldsymbol{v})$;
12   break;
13   **end**
14   **end**
15 $\boldsymbol{e} \leftarrow 0$;
16 **end**

---

is to apply an SMT solver, which is known to have a doubly-exponential complexity. Hence, there are other attempts to simplify this problem in special cases. In [8], G. Batt et al. presented a sufficient and necessary condition for deciding the existence of transitions between adjacent hyperrectangles (their definition of *adjacent* regions refers to those n-dimensional hyperrectangles having an $(n-1)$-dimensional facet in common). The condition states that for any two adjacent full-dimensional hyperrectangles R and R' (assuming R' is greater than R in x_i), there exists a transition from R to R' if and only if there exists a vertex on the shared facet F_i of R and R' satisfying $\dot{x}_i = f_i > 0$ (note that this is not true for adjacent hyperrectangles with shared facets of dimension lower than $n-1$). The condition is obvious for the sufficiency, but not for the necessity according to Theorem 2. To prove the necessity, we need to prove that there exists no trajectory from R to R' if $f_i \leq 0$ for all the vertices of F_i. This can be addressed in two cases:

- $f_i < 0$ for some vertex. According to the property of multiaffine functions described in Proposition 1, it is obvious that $f_i(x_\tau) < 0$ for all $x_\tau \in F_i \backslash \partial F_i$, where ∂F_i denotes the boundary of F_i. Therefore, all the trajectories that pass through F_i at the internal points must point from R' to R instead of the reverse.
- $f_i = 0$ for all vertices. We can easily derive from Proposition 1 that $f_i = 0$ for all the internal points of F_i. However, according to Theorem 2, we cannot conclude that there exist no trajectories from R to R' without further knowledge about the higher-order derivatives of $(x_i - c)$. Nevertheless, we

have the following proposition which asserts that all the Lie derivatives of $(x_i - c)$ at any internal points of F_i are 0 if the first Lie derivative of x_i at all the vertices of F_i are 0. Therefore, there exists no trajectory no matter in which direction, i.e. from R to R' or reversely.

Proposition 2. *Given a multiaffine system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ and a rectangular state space partition, where $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_n(\boldsymbol{x}))$, let R and R' be two n-dimensional hyperrectangles which share an $(n-1)$-dimensional facet F_i in the partition, where $F_i = \{(x_1, ..., x_n) \in \mathbb{R}^n \mid x_i = c, x_j \in [a_j, b_j], a_j \in \mathbb{R}, b_j \in \mathbb{R}, 1 \leq j \leq n, j \neq i\}$, and let $\mathbb{V}(F_i)$ denote the set of vertices of F_i. Then the following formula holds:*

$$\forall \boldsymbol{x} \in \mathbb{V}(F_i) : f_i(\boldsymbol{x}) = 0 \implies \forall \boldsymbol{x} \in F_i : \forall M \geq 0 : \mathcal{L}_{\boldsymbol{f}}^M f_i = 0 \tag{10}$$

Moreover, $x_i - c = 0$ is an invariant by the right hand side of Formula (10).

Proof. Suppose $\forall \boldsymbol{x} \in \mathbb{V}(F_i) : f_i(\boldsymbol{x}) = 0$ holds. According to Proposition 1, we can easily derive that $\forall \boldsymbol{x} \in F_i : f_i(\boldsymbol{x}) = 0$, which means that $f_i(\boldsymbol{x})$ must be of the form $f_i(\boldsymbol{x}) = (x_i - c)P_1(\boldsymbol{x})$, where $P_1(\boldsymbol{x})$ is a multiaffine function in $\mathbb{R}[x_1, ..., x_{i-1}, x_{i+1}, ..., x_n]$. To prove that $\forall \boldsymbol{x} \in F_i : \forall M \geq 0 : \mathcal{L}_{\boldsymbol{f}}^M f_i = 0$ holds, we only need to prove that every $\mathcal{L}_{\boldsymbol{f}}^M f_i$ has the form of $(x_i - c)P_M(\boldsymbol{x})$, where $P_M(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$. By induction, we assume that $\mathcal{L}_{\boldsymbol{f}}^{M-1} f_i = (x_i - c)P_{M-1}(\boldsymbol{x})$, so we only need to prove that $\mathcal{L}_{\boldsymbol{f}}^M f_i = (x_i - c)P_M(\boldsymbol{x})$. We have the following equation.

$$\begin{aligned}
\mathcal{L}_{\boldsymbol{f}}^M f_i &= \mathcal{L}_{\boldsymbol{f}}(\mathcal{L}_{\boldsymbol{f}}^{M-1} f_i) = (x_i - c)\mathcal{L}_{\boldsymbol{f}} P_{M-1} + P_{M-1}\mathcal{L}_{\boldsymbol{f}}(x_i - c) \\
&= (x_i - c)\mathcal{L}_{\boldsymbol{f}} P_{M-1} + P_{M-1}(x_i - c)P_1 = (x_i - c)(\mathcal{L}_{\boldsymbol{f}} P_{M-1} + P_{M-1}P_1)
\end{aligned} \tag{11}$$

With $P_M(\boldsymbol{x}) = \mathcal{L}_{\boldsymbol{f}} P_{M-1} + P_{M-1}P_1$, the above equation can be written as

$$\mathcal{L}_{\boldsymbol{f}}^M f_i = (x_i - c)P_M(\boldsymbol{x}) \tag{12}$$

Therefore, we can conclude that Formula (10) holds. Moreover, by the Taylor expansion of $x_i(t)$ at $x_i = c$ we can easily prove that $(x_i - c)$ is an invariant. $\quad\square$

Proposition 2 shows that there exists no trajectory connecting R and R' when $f_i = 0$ at all the vertices of F_i. However, this does not mean that there exists no trajectory that can reach F_i. In fact, there could exist an infinite number of trajectories which can reach F_i in the hyperplane P_{F_i} containing F_i. In fact, P_{F_i} forms an invariant of the system state space, i.e. any trajectory starting from P_{F_i} will never escape from P_{F_i}. To construct a precise over-approximation for the original system, these trajectories are non-negligible. However, these trajectories cannot be handled by the abstraction method in [8] because the authors only consider the transitions between adjacent full-dimensional hyperrectangles but ignore the transitions between the lower-dimensional hyperrectangles. In the following, we present an example to demonstrate the case.
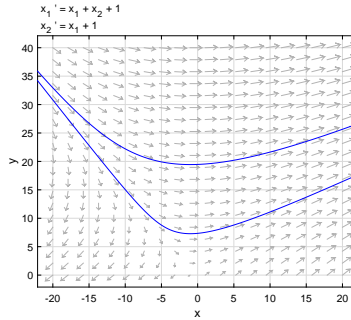
Fig. 2. Section of the vector field of Example 2 on the plane $x_3 = 1$.

Example 2. Consider the following multiaffine system.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} x_2 x_3 + x_1 + 1 \\ x_3 + x_1 + 1 \\ (x_3 - 1)(x_1 x_2 + 1) \end{bmatrix} \tag{13}$$

Let $Init = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid (x_1, x_2, x_3) \in [-10, 15] \times [15, 20] \times [0, 2]\}$ be the initial set and $Uns = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid (x_1, x_2, x_3) \in [15, 20] \times [20, 25] \times [1, 1]\}$ be the unsafe set.

Suppose we have a rectangular partition for the system in Example 2 that contains two boxes R and R' which share the facet F_i, where $\mathbb{V}(F_i) = \{(0, 0, 1), (1, 0, 1), (0, 1, 1), (1, 1, 1)\}$. We can easily verify that $\mathcal{L}_{\boldsymbol{f}}^M f_3(\boldsymbol{x}) = 0$ on the plane $x_3 = 1$ for all $M \geq 0$, which means that all the pairs of boxes besides (R, R') that share the plane $x_3 = 1$ cannot reach one another. However, according to Proposition 2, $x_3 = 1$ is an invariant and there could exist plenty of trajectories on the plane $x_3 = 1$ for which only the variables x_1 and x_2 change over time while x_3 stays unchanged. The section of the vector field at the plane $x_3 = 1$ is shown in Figure 2, and we can see that the system is unsafe. However, by using the abstraction method in [8], the transitions in the plane $x_3 = 1$ are ignored and hence the resulting transition system could be verified to be safe.

In [23], M. Kloetzer and C. Belta constructed an over-approximation for the original system by taking into account all the transitions between the hyperrectangles of different dimensions from 0 to n. In order to deal with the case of trajectories being tangent to the shared facet of neighboring hyperrectangles (i.e $f_i(\boldsymbol{x}) = 0$), the authors decide the direction of the trajectory by the direction of the vector flow in the neighboring hyperrectangles. This strategy works well only when the direction of the vector flow in the neighboring region is definite. Otherwise, the procedure fails and bidirectional transitions between two hyperrectangles have to be added in order to get an over-approximation. We present two examples to demonstrate how this can be handled by our method.

Example 3. Consider the following 3-dimensional multiaffine system.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} x_1 - 1 \\ x_2 + 1 \\ x_1 x_2 + x_1 + x_2 \end{bmatrix} \tag{14}$$

Let $\{-1, 0, 1\}^3$ be the set of grid points of a rectangular partition. What is the transition starting from the origin $(0, 0, 0)$?

For the system in Example 3, the flow vector at $(0, 0, 0)$ is $(-1, 1, 0)$. Since $\dot{x}_3 = 0$, to decide the target hyperrectangle of $(0, 0, 0)$ according to the algorithm in [23], the authors have to decide the direction of the vector flow in $H_0 = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid -1 < x_1 < 0, 0 < x_2 < 1, x_3 = 0\}$. Unfortunately, the direction of the vector flow is indefinite in H_0, and so they have to add two transitions: $(0, 0, 0) \to H_1$ and $H_0 \to (0, 0, 0)$, which is apparently not reasonable because there could exist only one trajectory passing through the origin. However, by using Corollary 1, we can easily decide that the trajectory enters the region of $\{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid -1 < x_1 < 0, 0 < x_2 < 1, -1 < x_3 < 0\}$ by $\mathcal{L}_{\boldsymbol{f}} x_1 = -1$, $\mathcal{L}_{\boldsymbol{f}} x_2 = 1$, $\mathcal{L}_{\boldsymbol{f}} x_3 = \mathcal{L}_{\boldsymbol{f}}^2 x_3 = 0$ and $\mathcal{L}_{\boldsymbol{f}}^3 x_3 = -2$.

Example 4. Consider the following 2-dimensional multiaffine system.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} (1 - x_1)x_2 \\ x_1 + x_2 + 1 \end{bmatrix} \tag{15}$$

Let $\{0, 1, 2\} \times \{0, 1, 2\}$ be the set of grid points of a rectangular partition. What are the transitions starting from the region $H_1 = \{(x_1, x_2) \in \mathbb{R}^2 \mid 0 < x_1 < 1, 0 < x_2 < 1\}$?

For the system in Example 4, it is easy to verify that $\dot{x}_1 > 0$ and $\dot{x}_2 > 0$ for all $(x_1, x_2) \in H_1$. Let $H_2 = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 = 1, 0 < x_2 < 1\}$, $H_3 = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 = x_2 = 1\}$ and $H_4 = \{(x_1, x_2) \in \mathbb{R}^2 \mid 0 < x_1 < 1, x_2 = 1\}$. According to the algorithm in [23], there should be the following transitions in the abstraction: $H_1 \to H_1$, $H_1 \to H_2$, $H_1 \to H_3$ and $H_1 \to H_4$. According to Corollary 1, however, there could not exist transitions from H_1 to H_2 and H_3 due to the fact that $L_f^m(x_1 - 1) = 0$ for all $m > 0$ in $L_1 = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 = 1\}$. In fact, L_1 is an invariant, which means that any trajectory reaching L_1 must start from and always stay in L_1.

Recall that Proposition 2 concludes that no trajectory can pass through an $(n - 1)$-dimensional facet F_i in an n-dimensional space if $f_i(\boldsymbol{v}_j) = 0$ for every vertex $\boldsymbol{v}_j$ of F_i, where f_i is the i'th component of the vector flow $\boldsymbol{f}$. However, this is not true for a facet F_i of lower dimension than $n - 1$. In other words, there could exist trajectories that can pass though an edge in the direction of x_i even if $f_i(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ in F_i. Let us demonstrate this case using the following example.

Example 5. Consider the following 4-dimensional multiaffine system.

$$
\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \\ f_4(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} -x_4 x_2 + x_4 x_3 + 2x_2 x_3 + x_1 - x_2 - 3x_3 + 1 \\ x_3 \\ x_1 \\ x_1 + x_2 \end{bmatrix} \tag{16}
$$

Let $(1, 1, 1, 3)$ and $(1, 1, 1, 4)$ be the grid points of a rectangular partition, $R_1 = \{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 \mid x_1 < 1, x_2 = 1, x_3 = 1, 3 < x_4 < 4\}$, $R_2 = \{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 \mid x_1 = x_2 = x_3 = 1, 3 < x_4 < 4\}$ and $R_3 = \{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 \mid x_1 > 1, x_2 = 1, x_3 = 1, 3 < x_4 < 4\}$. We know that both R_1 and R_3 are adjacent to R_2. To establish the transitions between them, we need to check the signs of the Lie derivatives of $x_1 - 1$ in R_2. Let $L_1 = \langle x_1 - 1, x_2 - 1, x_3 - 1 \rangle$. We can verify that $\mathcal{L}_{\boldsymbol{f}}(x_1 - 1), \mathcal{L}_{\boldsymbol{f}}^2(x_1 - 1) \in L_1$ and the remainder of $\mathcal{L}_{\boldsymbol{f}}^3(x_1 - 1)$ w.r.t. L_1 is $(5 - x_4)$, which means that in R_2 both $\mathcal{L}_{\boldsymbol{f}}(x_1 - 1)$ and $\mathcal{L}_{\boldsymbol{f}}^2(x_1 - 1)$ are identical to 0 and $\mathcal{L}_{\boldsymbol{f}}^3(x_1 - 1) > 0$. Hence, according to Corollary 1, there are transitions $R_1 \to R_2$ and $R_2 \to R_3$. Obviously, this is not decidable only by first-order Lie derivatives as in [8, 23].

6 Conclusion

In this paper, in order to reduce the conservativeness of the discrete abstraction, we proposed a new method based on a sufficient and necessary decision condition for establishing the discrete transitions between the abstract states in the abstract system. The partition of the state space of a multiaffine system is assumed to be based on a set of multivariate polynomials. A rectangular partition is just a simple case of a multivariate polynomial partition and the new decision condition applies naturally. Examples show the improvement of our method over the existing methods.

Acknowledgement

References

1. R. Alur, T. Dang, and F. Ivančić. Progress on reachability analysis of hybrid systems using predicate abstraction. In *HSCC*, pages 4–19. Springer, 2003.
2. E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Hybrid Systems: Computation and Control*, volume 2623 of *LNCS*, pages 20–35. Springer, 2003.

3. S. Bak, S. Bogomolov, T. A. Henzinger, T. T. Johnson, and P. Prakash. Scalable static hybridization methods for analysis of nonlinear systems. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC 2016, Vienna, Austria, April 12-14, 2016*, pages 155–164, 2016.
4. E. Bartocci, F. Corradini, M. R. Di Berardini, E. Entcheva, S. A. Smolka, and R. Grosu. Modeling and simulation of cardiac tissue using hybrid I/O automata. *Theoretical Computer Science*, 410(33):3149–3165, 2009.
5. E. Bartocci, F. Corradini, E. Entcheva, R. Grosu, and S. A. Smolka. Cellexcite: an efficient simulation environment for excitable cells. *BMC Bioinformatics*, 9:S–2, 2008.
6. E. Bartocci and P. Lió. Computational modeling, formal analysis and tools for systems biology. *PLOS Computational Biology*, 12(1):e1004591, 2016.
7. E. Bartocci, P. Lió, E. Merelli, and N. Paoletti. Multiple verification in complex biological systems: The bone remodelling case study. *Trans. Computational Systems Biology*, 14:53–76, 2012.
8. G. Batt, C. Belta, and R. Weiss. Temporal logic analysis of gene networks under parameter uncertainty. *Transactions on Automatic Control*, 53(Special Issue):215–229, 2008.
9. G. Batt, H. De Jong, M. Page, and J. Geiselmann. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, 44(4):982–989, 2008.
10. C. Belta and L. C. Habets. Controlling a class of nonlinear systems on rectangles. *Transactions on Automatic Control*, 51(11):1749–1759, 2006.
11. S. Bogomolov, A. Donzé, G. Frehse, R. Grosu, T. T. Johnson, H. Ladan, A. Podelski, and M. Wehrle. Guided search for hybrid systems based on coarse-grained space abstractions. *International Journal on Software Tools for Technology Transfer*, pages 1–19, 2015.
12. S. Bogomolov, G. Frehse, M. Greitschus, R. Grosu, C. S. Pasareanu, A. Podelski, and T. Strump. Assume-guarantee abstraction refinement meets hybrid systems. In *HVC 2014*, pages 116–131. Springer.
13. S. Bogomolov, C. Schilling, E. Bartocci, G. Batt, H. Kong, and R. Grosu. Abstraction-based parameter synthesis for multiaffine systems. In *Haifa Verification Conference*, pages 19–35. Springer, 2015.
14. D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2007.
15. T. Dang, C. Le Guernic, and O. Maler. Computing reachable states for nonlinear biological models. In *International Conference on Computational Methods in Systems Biology*, pages 126–141. Springer, 2009.
16. T. Dreossi and T. Dang. Parameter synthesis for polynomial biological models. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 233–242. ACM, 2014.
17. R. Grosu, G. Batt, F. H. Fenton, J. Glimm, C. Le Guernic, S. A. Smolka, and E. Bartocci. From cardiac cells to genetic regulatory networks. In *International Conference on Computer Aided Verification*, pages 396–411. Springer, 2011.
18. L. Habets, P. J. Collins, and J. H. van Schuppen. Reachability and control synthesis for piecewise-affine hybrid systems on simplices. *IEEE Transactions on Automatic Control*, 51(6):938–948, 2006.
19. L. Habets and J. H. Van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40(1):21–35, 2004.

20. L. C. G. J. M. Habets and J. Van Schuppen. A control problem for affine dynamical systems on a full-dimensional simplex. *Report-Probability, networks and algorithms*, (17):1–19, 2000.
21. Y. Jiang, H. Zhang, Z. Li, Y. Deng, X. Song, M. Gu, and J. Sun. Design and optimization of multiclocked embedded systems using formal techniques. *IEEE Transactions on Industrial Electronics*, 62(2):1270–1278, 2015.
22. Y. Jiang, H. Zhang, H. Zhang, H. Liu, X. Song, M. Gu, and J. Sun. Design of mixed synchronous/asynchronous systems with multiple clocks. *IEEE Transactions on Parallel and Distributed Systems*, 26(8):2220–2232, 2015.
23. M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. In *HSCC*, pages 348–362. Springer, 2006.
24. H. Kong, S. Bogomolov, C. Schilling, Y. Jiang, and T. A. Henzinger. Invariant clusters for hybrid systems. *arXiv preprint arXiv:1605.01450*, 2016.
25. H. Kong, F. He, X. Song, W. N. Hung, and M. Gu. Exponential-condition-based barrier certificate generation for safety verification of hybrid systems. In *CAV*, pages 242–257. Springer, 2013.
26. H. Kong, X. Song, D. Han, M. Gu, and J. Sun. A new barrier certificate for safety verification of hybrid systems. *The Computer Journal*, 57(7):1033–1045, 2014.
27. J. Liu, N. Zhan, and H. Zhao. Computing semi-algebraic invariants for polynomial dynamical systems. In *International Conference on Embedded Software*, pages 97–106. ACM, 2011.
28. A. Sogokon, K. Ghorbal, P. B. Jackson, and A. Platzer. A method for invariant generation for polynomial continuous systems. In *International Conference on Verification, Model Checking, and Abstract Interpretation*, pages 268–288. Springer, 2016.
29. A. Tiwari. Abstractions for hybrid systems. *Formal Methods in System Design*, 32(1):57–83, 2008.
30. A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. In *HSCC*, pages 465–478. Springer, 2002.