

Automatically Discovering Relaxed Lyapunov Functions for Polynomial Dynamical Systems

Jiang Liu · Naijun Zhan · Hengjun Zhao

Received: 13 December 2011 / Revised: 26 June 2012 / Accepted: 19 October 2012 / Published online: 29 November 2012
© Springer Basel 2012

Abstract The notion of Lyapunov function plays a key role in the design and verification of dynamical systems, as well as hybrid and cyber-physical systems. In this paper, to analyze the asymptotic stability of a dynamical system, we generalize standard Lyapunov functions to *relaxed Lyapunov functions* (RLFs), by considering higher order Lie derivatives. Furthermore, we present a method for automatically discovering polynomial RLFs for polynomial dynamical systems (PDSs). Our method is relatively complete in the sense that it is able to discover all polynomial RLFs with a given predefined template for any PDS. Therefore it can also generate all polynomial RLFs for the PDS by enumerating all polynomial templates.

Keywords Polynomial dynamical system · Asymptotic stability · Lyapunov function · Higher order Lie derivative

Mathematics Subject Classification (2010) Primary 93D20; Secondary 13F20

This work is supported in part by the projects NSFC-91118007, NSFC-60970031, NSFC-60736017, NSFC-61202131, YIGH-405070, cstc2011ggB40027, cstc2012ggB40004 and the PEARL project (041/2007/A3) from FDCT (Macau).

J. Liu (✉)

Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences,
Building B, HON KWOW Center No. 85 Jinyu Avenue, North Zone,
Chongqing 401122, People's Republic of China
e-mail: liujiang@cigit.ac.cn

N. Zhan

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences,
No. 4 South Fourth Street, Zhong Guan Cun,
Beijing 100190, People's Republic of China
e-mail: znj@ios.ac.cn

H. Zhao

State key Laboratory of Computer Science, Institute of Software,
University of Chinese Academy of Science, Chinese Academy of Sciences,
No. 4 South Fourth Street, Zhong Guan Cun,
Beijing 100190, People's Republic of China
e-mail: zhaohj@ios.ac.cn

1 Introduction

The notion of Lyapunov function plays a very important role in the design and verification of dynamical systems, in particular, in performance analysis, stability analysis and controller synthesis of complex dynamical and controlled systems [12, 15, 30]. In recent years, people realized that the notion is quite helpful for safety verification of hybrid and cyber-physical systems as well [31].

However, the following two issues hinder the application of Lyapunov functions in practice. Firstly, as shown by *LaSalle's Invariance Principle* [15], it is actually not necessary to require the first-order Lie derivative of a Lyapunov function to be strictly negative in order to guarantee asymptotic stability, and therefore this limits the applicability of the method. Secondly, in general there is no effective way so far to find Lyapunov functions, although many methods have been proposed by different researchers using expertise of their fields.

To address the above two issues, in this paper, we first generalize the standard concept of Lyapunov function to *relaxed Lyapunov function* (RLF). Compared with the conventional definition of Lyapunov function, the first non-zero higher order Lie derivative of an RLF is required to be negative, rather than its first-order Lie derivative. Such a relaxation extends the set of admissible functions that can be used to certify asymptotic stability. Based on RLFs, we give a new criterion for deciding asymptotic stability. Note that the same terminology “relaxed Lyapunov function” was used in [27] with a different definition.

Another contribution of this paper is that we present a relatively complete method for automatically discovering polynomial RLFs for polynomial dynamical systems (PDSs). The basic idea is to predefine a parametric polynomial as a template of RLF first, and then utilize the Lie derivatives of the template at different orders to generate constraints on the parameters, and finally solve the resulting constraints using computer algebra tools like DISCOVERER [36], QEPCAD [3] and Redlog [8]. Here, relative completeness means that our method is able to discover all polynomial RLFs with a given predefined template for a given PDS. Therefore it can also generate all polynomial RLFs for the PDS by enumerating all polynomial templates. Notice that the enumeration of templates can be automatically managed with respect to polynomial degrees. That is, for each $d \in \mathbb{N}$, we predefine a template polynomial with degree d and compute the corresponding constraint. Such an enumeration process does not terminate. However, in practice we can stop when reaching an assumed upper bound on the degree of templates.

Related Work The idea of applying higher order Lie derivatives to analyze asymptotic stability is not new. For example, in [4, 20] the authors resorted to certain linear combinations of higher order Lie derivatives with non-negative coefficients such that the combination satisfies certain conditions. This method is comparable with the framework of vector Lyapunov functions method [19, 21]. The authors of [2] presented a close work to [20] which constructs standard Lyapunov functions using sums-of-squares (SOS) optimization and semidefinite programming (SDP) tools. Our method is essentially different from theirs because an RLF only requires its first non-zero higher order Lie derivative to be negative. The work [13] indeed proposed a very similar condition to ours: first-order Lie derivative negative semidefinite and some conditions on higher order Lie derivatives. In [9] there is also a follow-up work to [13]. Anyway, in these two works, the Lyapunov functions allow the first non-zero higher order Lie derivative to be occasionally positive.

In the literature, there is a lot of work on constructing Lyapunov functions. For instance, in [1, 11, 16], methods for constructing common quadratic Lyapunov functions for linear systems were proposed, which were generalized in [29] and [34] for nonlinear systems wherein the generated Lyapunov functions are not necessarily quadratic. Another useful technique is the linear matrix inequality (LMI) method introduced in [14] and [25], which utilized the results of numerical optimization for discovering piecewise quadratic Lyapunov functions. Based on SOS and SDP [24], a method for constructing piecewise high-degree polynomial and piecewise non-polynomial Lyapunov functions was proposed in [23] and [26]. The SOS and SDP based method was also used in [32] to search for control Lyapunov functions for polynomial systems. In [28], the authors proposed a new method for computing Lyapunov functions for PDSs by solving semi-algebraic constraints using the tool DISCOVERER [36]. Approaches for constructing Lyapunov functions beyond polynomials using radial basis functions were proposed in [10].

Our method has the following features compared to the related work. Firstly, it generates relaxed Lyapunov functions rather than conventional Lyapunov functions. Secondly, it is able to discover all polynomial RLFs by enumerating all polynomial templates for any PDS, whereas other methods can only produce Lyapunov functions of special forms. Thirdly, the LMI method and SOS method are numerical, while our method is symbolic, which means it could provide a mathematically rigorous framework for the stability analysis of PDSs.

Structure The rest of this paper is organized as follows. For a self-contained presentation, Sect. 2 gives the theoretical foundations based on which our method is developed. Section 3 shows a new criterion for asymptotic stability using the notion of relaxed Lyapunov functions. In Sect. 4 we present a sound and relatively complete method and a corresponding algorithm for automatically discovering polynomial RLFs for PDSs. The method is illustrated using an example in Sect. 5. Finally, we conclude this paper and discuss possible future work in Sect. 6.

2 Theoretical Foundations

To make the paper self-contained, we present the fundamental concepts and results in polynomial ideal theory and Lyapunov stability, which could be skipped if the reader is familiar with them.

2.1 Polynomial Ideal Theory

Let \mathbb{K} be an algebraic field, and $\mathbb{K}[x_1, x_2, \dots, x_n]$ denote the polynomial ring over \mathbb{K} . Customarily, let \mathbf{x} denote the n -tuple (x_1, \dots, x_n) . Then $\mathbb{K}[x_1, x_2, \dots, x_n]$ can be written as $\mathbb{K}[\mathbf{x}]$ for short, and a polynomial in $\mathbb{K}[\mathbf{x}]$ can simply be written as $p(\mathbf{x})$ or p . In this paper, \mathbb{K} will be taken as the rational number field \mathbb{Q} or the real number field \mathbb{R} according to different contexts, and \mathbf{x} takes values from the n -dimensional Euclidean space \mathbb{R}^n .

In our method we will use polynomials with undetermined coefficients, called parametric polynomials or *templates*. Such polynomials can be denoted by $p(\mathbf{u}, \mathbf{x})$, where $\mathbf{u} = (u_1, u_2, \dots, u_t)$ is a t -tuple of parameters. Given $\mathbf{u}_0 \in \mathbb{R}^t$, we call the polynomial $p_{\mathbf{u}_0}(\mathbf{x})$ resulted by substituting \mathbf{u}_0 for \mathbf{u} in $p(\mathbf{u}, \mathbf{x})$ an *instantiation* of $p(\mathbf{u}, \mathbf{x})$. In the sequel, all parametric polynomials $p(\mathbf{u}, \mathbf{x})$ are assumed to be in $\mathbb{Q}[\mathbf{u}, \mathbf{x}]$, while their instantiations $p_{\mathbf{u}_0}(\mathbf{x})$ are polynomials in $\mathbb{R}[\mathbf{x}]$.

We briefly introduce some fundamental results relative to polynomial ideals in the following for self-containedness. Please refer to [6] for more details.

Definition 2.1 A subset $I \subseteq \mathbb{K}[\mathbf{x}]$ is called an *ideal* iff

- (a) $0 \in I$;
- (b) If $p(\mathbf{x}), g(\mathbf{x}) \in I$, then $p(\mathbf{x}) + g(\mathbf{x}) \in I$;
- (c) If $p(\mathbf{x}) \in I$, then $p(\mathbf{x})h(\mathbf{x}) \in I$ for any $h(\mathbf{x}) \in \mathbb{K}[\mathbf{x}]$.

It is easy to check that if $p_1, \dots, p_m \in \mathbb{K}[\mathbf{x}]$, then

$$\langle p_1, \dots, p_m \rangle = \left\{ \sum_{i=1}^m p_i h_i \mid \forall 1 \leq i \leq m. h_i \in \mathbb{K}[\mathbf{x}] \right\}$$

is an ideal (actually the least ideal containing p_1, p_2, \dots, p_m). In general, we say an ideal I is *generated* by polynomials $g_1, g_2, \dots, g_k \in \mathbb{K}[\mathbf{x}]$ if $I = \langle g_1, g_2, \dots, g_k \rangle$, where all g_i for $i \in \{1, 2, \dots, k\}$ are called *generators* of I . In fact, we have

Theorem 2.2 (Hilbert Basis Theorem) *Every ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ has a finite generating set. That is, $I = \langle g_1, g_2, \dots, g_k \rangle$ for some $g_1, g_2, \dots, g_k \in \mathbb{K}[\mathbf{x}]$.*

From this result, it is easy to see that

Theorem 2.3 (Ascending chain condition) *For any ascending chain*

$$I_1 \subseteq I_2 \subseteq \cdots \subseteq I_m \subseteq \cdots$$

of ideals in polynomial ring $\mathbb{K}[\mathbf{x}]$, there must be an N such that for all $m \geq N$, $I_m = I_N$.

2.2 Dynamical Systems and Stability

We summarize some fundamental theories of dynamical systems here. For details please refer to [12, 15, 30].

2.2.1 Dynamical Systems

We consider autonomous dynamical systems modeled by first-order ordinary differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ and \mathbf{f} is a vector function from \mathbb{R}^n to \mathbb{R}^n , which is also called a vector field in \mathbb{R}^n .

In this paper, we focus on special nonlinear dynamical systems whose vector fields are defined by polynomials.

Definition 2.4 (*Polynomial dynamical system*) Suppose $\mathbf{f} = (f_1, f_2, \dots, f_n)$ in (2.1). Then (2.1) is called a *polynomial dynamical system* (PDS for short) if for every $1 \leq i \leq n$, f_i is a polynomial in $\mathbb{K}[\mathbf{x}]$. Such \mathbf{f} is called a *polynomial vector field*.

For the sake of computability, in this paper, all PDSs are assumed to have vector fields defined by polynomials with rational coefficients, i.e. $\mathbf{f} \in \mathbb{Q}^n[\mathbf{x}]$.

If \mathbf{f} in (2.1) satisfies the *local Lipschitz condition*, then given $\mathbf{x}_0 \in \mathbb{R}^n$, there exists a unique solution $\mathbf{x}(t)$ of (2.1) defined on (a, b) with $a < 0 < b$ s.t.

$$\forall t \in (a, b). \frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t)) \quad \text{and} \quad \mathbf{x}(0) = \mathbf{x}_0.$$

We call $\mathbf{x}(t)$ on $[0, b)$ the *trajectory* of (2.1) starting from the initial point \mathbf{x}_0 .

Let $g(\mathbf{x})$ be a function from \mathbb{R}^n to \mathbb{R} . Suppose both g and \mathbf{f} are differentiable in \mathbf{x} at any order $n \in \mathbb{N}$ (i.e. g and \mathbf{f} are *smooth functions*). Then we can inductively define the *Lie derivatives* of g along \mathbf{f} which are functions like $L : \mathbb{R}^n \rightarrow \mathbb{R}$, as follows:

- $L_{\mathbf{f}}^0 g(\mathbf{x}) = g(\mathbf{x})$,
- $L_{\mathbf{f}}^{k+1} g(\mathbf{x}) = (\nabla L_{\mathbf{f}}^k g, \mathbf{f})$, for $k \geq 0$,

where ∇ is the *gradient* operator, i.e. $\nabla L_{\mathbf{f}}^k g \triangleq (\frac{\partial L_{\mathbf{f}}^k g}{\partial x_1}, \frac{\partial L_{\mathbf{f}}^k g}{\partial x_2}, \dots, \frac{\partial L_{\mathbf{f}}^k g}{\partial x_n})$, and (\cdot, \cdot) is the *inner product* of two vectors, i.e. $((a_1, \dots, a_n), (b_1, \dots, b_n)) = \sum_{i=1}^n a_i b_i$.

Simply, the zeroth-order Lie derivative of g along \mathbf{f} is g itself, and the first-order Lie derivative describes the slope of g along the solution trajectories of \mathbf{f} .

Polynomial functions are smooth. Thus polynomial vector field \mathbf{f} is locally Lipschitz. Besides, given a polynomial vector field \mathbf{f} and a polynomial p , the Lie derivatives of p along \mathbf{f} are well defined at all orders and are all polynomials. For a parameterized polynomial $p(\mathbf{u}, \mathbf{x})$, we can define $L_{\mathbf{f}}^k p(\mathbf{u}, \mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ by seeing \mathbf{u} as undetermined constants rather than variables. In the sequel we will implicitly employ these facts.

Example 2.5 Suppose $\mathbf{f} = (-x, y)$ and $p(x, y) = x + y^2$. Then $L_{\mathbf{f}}^0 p = x + y^2$, $L_{\mathbf{f}}^1 p = -x + 2y^2$, $L_{\mathbf{f}}^2 p = x + 4y^2$, $L_{\mathbf{f}}^3 p = -x + 8y^2$.

2.2.2 Stability

The following are classical results of stability theory in the sense of Lyapunov.

Definition 2.6 A point $\mathbf{x}_e \in \mathbb{R}^n$ is called an *equilibrium point* or *critical point* of (2.1) if $\mathbf{f}(\mathbf{x}_e) = \mathbf{0}$.

We assume $\mathbf{x}_e = \mathbf{0}$ w.l.o.g from now on.

Definition 2.7 Suppose $\mathbf{0}$ is an equilibrium point of (2.1). Then

- $\mathbf{0}$ is called *Lyapunov stable* if for any $\epsilon > 0$, there exists a $\delta > 0$ such that if $\|\mathbf{x}_0\| < \delta$,¹ then the corresponding solution $\mathbf{x}(t)$ of (2.1) starting from x_0 can be extended to infinity, and $\forall t \geq 0, \|\mathbf{x}(t)\| < \epsilon$;
- $\mathbf{0}$ is called *asymptotically stable* if it is Lyapunov stable and there exists a $\delta > 0$ such that for any $\|\mathbf{x}_0\| < \delta$, the corresponding solution $\mathbf{x}(t)$ of (2.1) starting from x_0 satisfies $\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0}$.

Lyapunov first provided a sufficient condition, using so-called *Lyapunov functions*, for the Lyapunov stability as follows.

Theorem 2.8 (Lyapunov Stability Theorem) *Suppose $\mathbf{0}$ is an equilibrium point of (2.1). If there is an open set $U \subseteq \mathbb{R}^n$ with $\mathbf{0} \in U$ and a continuous differentiable function $V : U \rightarrow \mathbb{R}$ such that*

- $V(\mathbf{0}) = 0$,
- $V(\mathbf{x}) > 0$ for all $\mathbf{x} \in U \setminus \{\mathbf{0}\}$ and
- $L_{\mathbf{f}}^1 V(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in U$,

then $\mathbf{0}$ is a stable equilibrium point of (2.1). Moreover, if condition (c) is replaced by

- $L_{\mathbf{f}}^1 V(\mathbf{x}) < 0$ for all $\mathbf{x} \in U \setminus \{\mathbf{0}\}$,

then $\mathbf{0}$ is an asymptotically stable equilibrium point of (2.1). Such V satisfying (a), (b) and (c) (or (c^)) is called a Lyapunov function.*

Basically, for asymptotic stability of autonomous dynamical systems, the Lyapunov stability theorem requires a *positive definite* function V with *negative definite* first-order Lie derivative $L_{\mathbf{f}}^1 V$ in a neighborhood of the equilibrium. If V has only negative *semi-definite* $L_{\mathbf{f}}^1 V$ but no trajectories can stay identically in the zero level set of $L_{\mathbf{f}}^1 V$, then the asymptotic stability can also be guaranteed, which is known as the Barbashin-Krasovskii-LaSalle (BKL) Principle.

Theorem 2.9 (BKL Principle) *Let V be a function satisfying conditions (a), (b) and (c) in Theorem 2.8. If the set $\mathcal{M} \triangleq \{\mathbf{x} \in U \mid L_{\mathbf{f}}^1 V(\mathbf{x}) = 0\}$ does not contain any trajectory of the system other than the trivial trajectory $\mathbf{x}(t) \equiv \mathbf{0}$, then $\mathbf{0}$ is asymptotically stable.*

Inspired by Theorem 2.9, we will define *relaxed Lyapunov functions* (RLFs for short) in the subsequent section, which guarantee the asymptotic stability of an equilibrium point.

3 Relaxed Lyapunov Function

Intuitively, a Lyapunov function requires that any trajectory starting from $\mathbf{x}_0 \in U$ cannot leave the region $\{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq V(\mathbf{x}_0)\}$. In the asymptotic stability case, the corresponding V forces any trajectory starting from $\mathbf{x}_0 \in U$ to transect the boundary $\{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) = V(\mathbf{x}_0)\}$ towards the set $\{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) < V(\mathbf{x}_0)\}$. The left picture in Fig. 1 illustrates how a Lyapunov function guarantees asymptotic stability.

¹ Given $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$ denotes the Euclidean norm of \mathbf{x} .

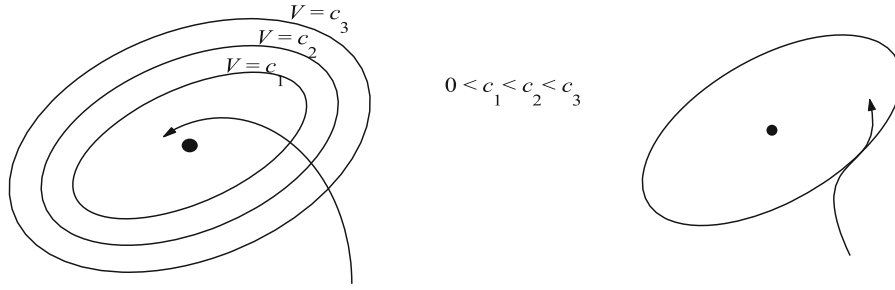


Fig. 1 Trajectories transecting Lyapunov surfaces

It is clear that $L_{\mathbf{f}}^1 V(\mathbf{x}) < 0$ is only a sufficient condition for asymptotic stability. When a point \mathbf{x} satisfies $L_{\mathbf{f}}^1 V(\mathbf{x}) = 0$, the transection requirement may still be met if the first non-zero higher order Lie derivative of V at \mathbf{x} is negative. In this case, the trajectory may be tangential to a Lyapunov surface at the cross point (see the right picture in Fig. 1). To formalize the above idea, and motivated by our recent work [17] on continuous invariant generation, we give the following definitions.

Definition 3.1 (*Pointwise Rank*) Let \mathbb{N}^+ be the set of positive natural numbers. Given a smooth function g and a smooth vector field \mathbf{f} , the *pointwise rank* of g w.r.t. \mathbf{f} is defined as the function $\gamma_{g,\mathbf{f}} : \mathbb{R}^n \rightarrow \mathbb{N}^+ \cup \{\infty\}$ given by

$$\gamma_{g,\mathbf{f}}(\mathbf{x}) = \begin{cases} \infty, & \text{if } \forall k \in \mathbb{N}^+, L_{\mathbf{f}}^k g(\mathbf{x}) = 0, \\ \min\{k \in \mathbb{N}^+ \mid L_{\mathbf{f}}^k g(\mathbf{x}) \neq 0\}, & \text{otherwise.} \end{cases}$$

Example 3.2 For $\mathbf{f} = (-x, y)$ and $p(x, y) = x + y^2$, by Example 2.5, we have $\gamma_{p,\mathbf{f}}(0, 0) = \infty$, $\gamma_{p,\mathbf{f}}(1, 1) = 1$, $\gamma_{p,\mathbf{f}}(2, 1) = 2$.

Definition 3.3 (*Transverse Set*) Given a smooth function g and a smooth vector field \mathbf{f} , the *transverse set* of g w.r.t. \mathbf{f} is defined as

$$\text{Trans}_{g,\mathbf{f}} \hat{=} \{\mathbf{x} \in \mathbb{R}^n \mid \gamma_{g,\mathbf{f}}(\mathbf{x}) < \infty \wedge L_{\mathbf{f}}^{\gamma_{g,\mathbf{f}}(\mathbf{x})} g(\mathbf{x}) < 0\}.$$

Intuitively, $\text{Trans}_{g,\mathbf{f}}$ consists of those points at which the first non-zero higher order Lie derivative of g along \mathbf{f} is negative. Now we can relax condition (c*) in Theorem 2.8 and get a new criterion for asymptotic stability.

Theorem 3.4 *Suppose $\mathbf{0}$ is an equilibrium point of a dynamical system defined by smooth vector field \mathbf{f} . If there is an open set $U \subseteq \mathbb{R}^n$ with $\mathbf{0} \in U$ and a smooth function $V : U \rightarrow \mathbb{R}$ s.t.*

- (a) $V(\mathbf{0}) = 0$,
- (b) $V(\mathbf{x}) > 0$ for all $\mathbf{x} \in U \setminus \{\mathbf{0}\}$ and
- (c) $\mathbf{x} \in \text{Trans}_{V,\mathbf{f}}$ for all $\mathbf{x} \in U \setminus \{\mathbf{0}\}$,

then $\mathbf{0}$ is an asymptotically stable equilibrium point of (2.1).

Proof First, condition (c) implies $L_{\mathbf{f}}^1 V(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in U \setminus \{\mathbf{0}\}$. In order to show the asymptotic stability of $\mathbf{0}$, according to Theorem 2.9, it is sufficient to show that $\mathcal{M} \hat{=} \{\mathbf{x} \in U \mid L_{\mathbf{f}}^1 V(\mathbf{x}) = 0\}$ contains no nontrivial trajectory of the dynamical system.

If not, let $\mathbf{x}(t)$, $t \geq 0$ be such a trajectory contained in \mathcal{M} other than $\mathbf{x}(t) \equiv \mathbf{0}$. Then for all $t \geq 0$, $L_{\mathbf{f}}^1 V(\mathbf{x}(t)) = 0$ and $\mathbf{x}(t) \neq \mathbf{0}$. In particular, by (c), $\mathbf{x}_0 = \mathbf{x}(0) \in \text{Trans}_{V,\mathbf{f}}$. Then by Definition 3.3, the *Taylor Formula* of $L_{\mathbf{f}}^1 V(\mathbf{x}(t))$ at $t = 0$

$$\begin{aligned} L_{\mathbf{f}}^1 V(\mathbf{x}(t)) &= L_{\mathbf{f}}^1 V(\mathbf{x}_0) + L_{\mathbf{f}}^2 V(\mathbf{x}_0) \cdot t + \dots + L_{\mathbf{f}}^{\gamma_{V,\mathbf{f}}(\mathbf{x}_0)} V(\mathbf{x}_0) \cdot \frac{t^{\gamma_{V,\mathbf{f}}(\mathbf{x}_0)-1}}{(\gamma_{V,\mathbf{f}}(\mathbf{x}_0) - 1)!} + o(t^{\gamma_{V,\mathbf{f}}(\mathbf{x}_0)}) \\ &= L_{\mathbf{f}}^{\gamma_{V,\mathbf{f}}(\mathbf{x}_0)} V(\mathbf{x}_0) \cdot \frac{t^{\gamma_{V,\mathbf{f}}(\mathbf{x}_0)-1}}{(\gamma_{V,\mathbf{f}}(\mathbf{x}_0) - 1)!} + o(t^{\gamma_{V,\mathbf{f}}(\mathbf{x}_0)}) \end{aligned}$$

shows that there exists an $\epsilon > 0$ s.t. $\forall t \in (0, \epsilon)$. $L_{\mathbf{f}}^1 V(\mathbf{x}(t)) < 0$, which contradicts the fact $\forall t \geq 0$. $L_{\mathbf{f}}^1 V(\mathbf{x}(t)) = 0$. \square

Definition 3.5 (*Relaxed Lyapunov Function*) We refer to the function V in Theorem 3.4 as a *relaxed Lyapunov function*, denoted by RLF.

In the next section, we will explore how to automatically discover polynomial RLFs for PDSs.

4 Automatically Discovering Polynomial RLFs for PDSs

Given a PDS, the process of automatically discovering polynomial RLFs is as follows:

1. a template, i.e. a parametric polynomial $p(\mathbf{u}, \mathbf{x})$, is predefined as a candidate for RLF;
2. the conditions for $p(\mathbf{u}, \mathbf{x})$ to be an RLF are translated into an equivalent formula Φ of the decidable *first-order theory of real closed fields* [33];
3. the constraint Φ' on parameters \mathbf{u} , or equivalently a set $S_{\mathbf{u}} \subseteq \mathbb{R}^r$ consisting of all t -tuples subject to Φ' , is obtained by applying *quantifier elimination* (QE for short. See [3, 8]) to Φ , and any instantiation of \mathbf{u} by $\mathbf{u}_0 \in S_{\mathbf{u}}$ yields an RLF $p_{\mathbf{u}_0}(\mathbf{x})$.

4.1 Computation of Transverse Set

Step 2 in the above process, i.e. translation of the three conditions in Theorem 3.4, is crucial to our method. In particular, we have to show that for any polynomial $p(\mathbf{x})$ and polynomial vector field \mathbf{f} , the transverse set $\text{Trans}_{p, \mathbf{f}}$ can be represented by first-order polynomial formulas. We will achieve this by showing that there exists a uniform upper bound on $\gamma_{p, \mathbf{f}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$ with $\gamma_{p, \mathbf{f}}(\mathbf{x}) < \infty$. To this end, we first give several theorems by exploring the properties of Lie derivatives and polynomial ideals.

Theorem 4.1 (Fixed Point Theorem) *Given $p \in \mathbb{K}[\mathbf{x}]$, if $L_{\mathbf{f}}^{i+1} p \in \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle$, then for all $m > i$, $L_{\mathbf{f}}^m p \in \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle$.*

Proof We prove this fact by induction.

Induction hypothesis: $L_{\mathbf{f}}^{i+1} p \in \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle$.

Assume $L_{\mathbf{f}}^k p \in \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle$ for some $k > i$. Then there are $g_j \in \mathbb{K}[\mathbf{x}]$ s.t. $L_{\mathbf{f}}^k p = \sum_{j=1}^i g_j L_{\mathbf{f}}^j p$. By the definition of Lie derivative it follows that

$$\begin{aligned}
 L_{\mathbf{f}}^{k+1} p &= (\nabla L_{\mathbf{f}}^k p, \mathbf{f}) \\
 &= \left(\nabla \sum_{j=1}^i g_j L_{\mathbf{f}}^j p, \mathbf{f} \right) \\
 &= \left(\sum_{j=1}^i L_{\mathbf{f}}^j p \nabla g_j + \sum_{j=1}^i g_j \nabla L_{\mathbf{f}}^j p, \mathbf{f} \right) \\
 &= \sum_{j=1}^i (\nabla g_j, \mathbf{f}) L_{\mathbf{f}}^j p + \sum_{j=1}^i g_j L_{\mathbf{f}}^{j+1} p \\
 &= \sum_{j=1}^i (\nabla g_j, \mathbf{f}) L_{\mathbf{f}}^j p + \sum_{j=1}^{i-1} g_j L_{\mathbf{f}}^{j+1} p + g_i L_{\mathbf{f}}^{i+1} p.
 \end{aligned} \tag{4.1}$$

By induction hypothesis, $L_{\mathbf{f}}^{i+1} p \in \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle$. So by (4.1) we get $L_{\mathbf{f}}^{k+1} p \in \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle$. Therefore by induction the fact follows immediately. \square

Theorem 4.2 Given $p \in \mathbb{K}[\mathbf{x}]$, the number

$$N_{p,\mathbf{f}} = \min\{i \in \mathbb{N}^+ \mid L_{\mathbf{f}}^{i+1} p \in \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle\}$$

is well defined. Furthermore, $N_{p,\mathbf{f}}$ is computable if Gröbner bases are computable in $\mathbb{K}[x]$.

Proof First it is easy to show that $N_{p,\mathbf{f}}$ has an equivalent expression

$$N_{p,\mathbf{f}} = \min\{i \in \mathbb{N}^+ \mid I_{i+1} = I_i\},$$

where $I_i = \langle L_{\mathbf{f}}^1 p, \dots, L_{\mathbf{f}}^i p \rangle \subseteq \mathbb{K}[\mathbf{x}]$. Noting that

$$I_1 \subseteq I_2 \subseteq \dots \subseteq I_k \dots$$

forms an ascending chain of ideals, by Theorem 2.3, $N_{p,\mathbf{f}}$ is well-defined. Computation of $N_{p,\mathbf{f}}$ is actually an *ideal membership* problem, which can be solved by computation of *Gröbner bases* [6]. So $N_{p,\mathbf{f}}$ is computable if the field \mathbb{K} is properly chosen so that the *Gröbner bases* of ideals are computable in the ring $\mathbb{K}[\mathbf{x}]$. \square

Example 4.3 For $\mathbf{f} = (-x, y)$ and $p(x, y) = x + y^2$, by Example 2.5, we have $L_{\mathbf{f}}^2 p \notin \langle L_{\mathbf{f}}^1 p \rangle$ and $L_{\mathbf{f}}^3 p \in \langle L_{\mathbf{f}}^1 p, L_{\mathbf{f}}^2 p \rangle$. So $N_{p,\mathbf{f}} = 2$.

Given a parametric polynomial $p(\mathbf{u}, \mathbf{x}) \in \mathbb{Q}[\mathbf{u}, \mathbf{x}]$ and a polynomial vector field $\mathbf{f} \in \mathbb{Q}^n[\mathbf{x}]$, all Lie derivatives $L_{\mathbf{f}}^k p(\mathbf{u}, \mathbf{x})$ can be seen as polynomials in $\mathbb{Q}[\mathbf{u}, \mathbf{x}]$. Then using similar arguments we can give the parametric versions of Theorem 4.1 and Theorem 4.2 for $L_{\mathbf{f}}^k p(\mathbf{u}, \mathbf{x})$, just by taking all ideals in $\mathbb{Q}[\mathbf{u}, \mathbf{x}]$. Suppose $N_{p,\mathbf{f}}$ is defined as in Theorem 4.2 for $p(\mathbf{u}, \mathbf{x})$. Then we have

Theorem 4.4 (Rank Theorem) Suppose that $p \hat{=} p(\mathbf{u}, \mathbf{x})$. Then for all $\mathbf{x} \in \mathbb{R}^n$ and all $\mathbf{u}_0 \in \mathbb{R}^t$, $\gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x}) < \infty$ implies $\gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x}) \leq N_{p,\mathbf{f}}$.

Proof If the conclusion is not true, then there exist $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{u}_0 \in \mathbb{R}^t$ s.t.

$$N_{p,\mathbf{f}} < \gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x}_0) < \infty.$$

By Definition 3.1, \mathbf{x}_0 satisfies

$$L_{\mathbf{f}}^1 p_{\mathbf{u}_0}(\mathbf{x}_0) = 0 \wedge \dots \wedge L_{\mathbf{f}}^{N_{p,\mathbf{f}}} p_{\mathbf{u}_0}(\mathbf{x}_0) = 0 \wedge L_{\mathbf{f}}^{\gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x}_0)} p_{\mathbf{u}_0}(\mathbf{x}_0) \neq 0.$$

Then by Theorem 4.2 and 4.1, for all $m > N_{p,\mathbf{f}}$, we have $L_{\mathbf{f}}^m p_{\mathbf{u}_0}(\mathbf{x}_0) = 0$. In particular,

$$L_{\mathbf{f}}^{\gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x}_0)} p_{\mathbf{u}_0}(\mathbf{x}_0) = 0,$$

which contradicts $L_{\mathbf{f}}^{\gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x}_0)} p_{\mathbf{u}_0}(\mathbf{x}_0) \neq 0$. \square

In what follows, we will adopt the convention that $\bigvee_{i \in \emptyset} \eta_i = \text{false}$ and $\bigwedge_{i \in \emptyset} \eta_i = \text{true}$, where η_i is a logical formula. Now we are able to give a logical representation of transverse sets as follows.

Theorem 4.5 Given a parameterized polynomial $p \hat{=} p(\mathbf{u}, \mathbf{x})$ and a polynomial vector field \mathbf{f} , for any $\mathbf{u}_0 \in \mathbb{R}^t$ and any $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \in \text{Trans}_{p_{\mathbf{u}_0}, \mathbf{f}}$ if and only if \mathbf{u}_0 and \mathbf{x} satisfy $\varphi_{p,\mathbf{f}}$, where

$$\varphi_{p,\mathbf{f}} \hat{=} \bigvee_{i=1}^{N_{p,\mathbf{f}}} \varphi_{p,\mathbf{f}}^i, \quad \text{and} \tag{4.2}$$

$$\varphi_{p,\mathbf{f}}^i \hat{=} \left(\bigwedge_{j=1}^{i-1} L_{\mathbf{f}}^j p(\mathbf{u}, \mathbf{x}) = 0 \right) \wedge L_{\mathbf{f}}^i p(\mathbf{u}, \mathbf{x}) < 0. \tag{4.3}$$

Proof (\Rightarrow) Suppose $\mathbf{x} \in \text{Trans}_{p_{\mathbf{u}_0}, \mathbf{f}}$. By Definition 3.3, \mathbf{x} satisfies

$$L_{\mathbf{f}}^1 p_{\mathbf{u}_0}(\mathbf{x}) = 0 \wedge L_{\mathbf{f}}^2 p_{\mathbf{u}_0}(\mathbf{x}) = 0 \wedge \dots \wedge L_{\mathbf{f}}^{\gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x})} p_{\mathbf{u}_0}(\mathbf{x}) < 0. \tag{4.4}$$

By Theorem 4.4, $\gamma_{p_{\mathbf{u}_0}, \mathbf{f}}(\mathbf{x}) \leq N_{p,\mathbf{f}}$. Then it is obvious that (4.4) must be one of the disjuncts of (4.2), thus implying (4.2).

(\Leftarrow) If \mathbf{u}_0 and \mathbf{x} satisfy $\varphi_{p,\mathbf{f}}$, then they must satisfy one of the $N_{p,\mathbf{f}}$ disjuncts. By Definition 3.3 we can see that $\mathbf{x} \in \text{Trans}_{p_{\mathbf{u}_0}, \mathbf{f}}$ holds trivially. \square

4.2 A Sound and Relatively Complete Method for Generating RLFs

Based on the results established in Sect. 4.1, our main result of automatically generating polynomial RLFs for PDSs can be stated as the following theorem.

Theorem 4.6 (Main Result)

Given a PDS $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and a parametric polynomial $p \hat{=} p(\mathbf{u}, \mathbf{x})$. Let $\mathbf{u}_0 = (u_{10}, u_{20}, \dots, u_{t_0}) \in \mathbb{R}^t$. Then $p_{\mathbf{u}_0}$ is an RLF if and only if there exists $r_0 \in \mathbb{R}, r_0 > 0$ such that $(u_{10}, u_{20}, \dots, u_{t_0}, r_0)$ satisfies $\phi_{p,\mathbf{f}} \hat{=} \phi_{p,\mathbf{f}}^1 \wedge \phi_{p,\mathbf{f}}^2 \wedge \phi_{p,\mathbf{f}}^3$, where

$$\phi_{p,\mathbf{f}}^1 \hat{=} p(\mathbf{u}, \mathbf{0}) = 0, \quad (4.5)$$

$$\phi_{p,\mathbf{f}}^2 \hat{=} \forall \mathbf{x}. (\|\mathbf{x}\|^2 > 0 \wedge \|\mathbf{x}\|^2 < r^2 \rightarrow p(\mathbf{u}, \mathbf{x}) > 0), \quad (4.6)$$

$$\phi_{p,\mathbf{f}}^3 \hat{=} \forall \mathbf{x}. (\|\mathbf{x}\|^2 > 0 \wedge \|\mathbf{x}\|^2 < r^2 \rightarrow \varphi_{p,\mathbf{f}}) \quad (4.7)$$

with $\varphi_{p,\mathbf{f}}$ defined in (4.2).

Proof First, in Theorem 3.4, the existence of an open set U is equivalent to the existence of an open ball $\mathcal{B}(\mathbf{0}, r_0) \hat{=} \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| < r_0\}$. Then according to Theorem 4.5, it is easy to check that (4.5), (4.6) and (4.7) are direct translations of conditions (a), (b) and (c) in Theorem 3.4. \square

According to Theorem 4.6, we can follow the three steps at the beginning of Sect. 4 to discover polynomial RLFs for PDSs. This method is ‘‘relatively complete’’ because we can discover all possible polynomial RLFs in the form of a predefined template, and thus can find all polynomial RLFs by enumerating all polynomial templates for a given PDS.

4.3 Simplification and Implementation

To construct $\phi_{p,\mathbf{f}}$ in Theorem 4.6, we need to compute $N_{p,\mathbf{f}}$ in advance, whose complexity is very high. What is the worse, when $N_{p,\mathbf{f}}$ is a large number the resulting $\phi_{p,\mathbf{f}}$ could be a huge formula, for which QE is infeasible in practice. Regarding this, in the following we will try to reduce the complexity of RLF generation in two aspects:

- 1) we show that some of the quantifier elimination problems arising in the process of RLF generation can be reduced to so called *real root classification* (RRC for short) problems, which can be solved in a more efficient way compared to standard QE problems;
- 2) for analysis of asymptotic stability, one RLF is enough. Therefore we can search for RLF in a stepwise manner: if an RLF can be obtained by solving constraints involving merely lower order Lie derivatives, there is no need to resort to higher order ones.

We need the following three lemmas to explain the first aspect.

Lemma 4.7 Suppose \mathbf{f} is a smooth vector field, g is a smooth function defined in an open set $U \subseteq \mathbb{R}^n$, and $L_{\mathbf{f}}^1 g(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in U$. Then for any $\mathbf{x} \in U$, $\gamma_{g,\mathbf{f}}(\mathbf{x}) < \infty$ implies $\mathbf{x} \in \text{Trans}_{g,\mathbf{f}}$.

Proof Suppose there is an $\mathbf{x}_0 \in U$ such that $\gamma_{g,\mathbf{f}}(\mathbf{x}_0) < \infty$ and $L_{\mathbf{f}}^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)} g(\mathbf{x}_0) > 0$. Let $\mathbf{x}(t)$ be the trajectory of \mathbf{f} starting from \mathbf{x}_0 . Then from

$$\begin{aligned} L_{\mathbf{f}}^1 g(\mathbf{x}(t)) &= L_{\mathbf{f}}^1 g(\mathbf{x}_0) + L_{\mathbf{f}}^2 g(\mathbf{x}_0) \cdot t + \dots + L_{\mathbf{f}}^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)} g(\mathbf{x}_0) \cdot \frac{t^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)-1}}{(\gamma_{g,\mathbf{f}}(\mathbf{x}_0) - 1)!} + o(t^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)}) \\ &= L_{\mathbf{f}}^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)} g(\mathbf{x}_0) \cdot \frac{t^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)-1}}{(\gamma_{g,\mathbf{f}}(\mathbf{x}_0) - 1)!} + o(t^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)}) \end{aligned} \quad (4.8)$$

we can see that there exists an $\epsilon > 0$ such that $\forall t \in (0, \epsilon)$, $L_{\mathbf{f}}^1 g(\mathbf{x}(t)) > 0$, which contradicts $L_{\mathbf{f}}^1 g(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in U$. \square

Lemma 4.8 *Suppose \mathbf{f} is a smooth vector field, g is a smooth function defined in an open set $U \subseteq \mathbb{R}^n$, and $L_{\mathbf{f}}^1 g(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in U$. Then for any $\mathbf{x} \in U$, $\gamma_{g,\mathbf{f}}(\mathbf{x}) < \infty$ implies $\gamma_{g,\mathbf{f}}(\mathbf{x}) = 2k + 1$ for some $k \in \mathbb{N}$.*

Proof If there is an $\mathbf{x}_0 \in U$ such that $\gamma_{g,\mathbf{f}}(\mathbf{x}_0) < \infty$ and $\gamma_{g,\mathbf{f}}(\mathbf{x}_0) = 2k$ for some $k \in \mathbb{N}^+$, by Lemma 4.7 we have $L_{\mathbf{f}}^{\gamma_{g,\mathbf{f}}(\mathbf{x}_0)} g(\mathbf{x}_0) < 0$. Then again by (4.8) we can see there exists an $\epsilon > 0$ such that $\forall t \in (-\epsilon, 0)$, $L_{\mathbf{f}}^1 g(\mathbf{x}(t)) > 0$, which contradicts $L_{\mathbf{f}}^1 g(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in U$. \square

Lemma 4.9 *Suppose \mathbf{f} is a polynomial vector field and $p(\mathbf{x})$ is a polynomial, and $L_{\mathbf{f}}^1 p(\mathbf{x}) \leq 0$ for all \mathbf{x} in an open set $U \subseteq \mathbb{R}^n$. Then for any $\mathbf{x} \in U$, $\mathbf{x} \in \text{Trans}_{p,\mathbf{f}}$ if and only if \mathbf{x} is not a common root of the sequence of polynomials*

$$L_{\mathbf{f}}^1 p(\mathbf{x}), L_{\mathbf{f}}^3 p(\mathbf{x}), \dots, L_{\mathbf{f}}^{(2K_0+1)} p(\mathbf{x}),$$

where $K_0 \triangleq \lfloor \frac{N_{p,\mathbf{f}}-1}{2} \rfloor^2$ and $N_{p,\mathbf{f}}$ is defined in Theorem 4.2.

Proof (\Rightarrow) Actually K_0 has been chosen is such a way that $2K_0 + 1$ is the largest uneven number less than or equal to $N_{p,\mathbf{f}}$, i.e. $2K_0 + 1 = N_{p,\mathbf{f}}$ or $2K_0 + 1 = N_{p,\mathbf{f}} - 1$. Suppose $\mathbf{x}_0 \in \text{Trans}_{p,\mathbf{f}}$ and $L_{\mathbf{f}}^1 p(\mathbf{x}_0) = L_{\mathbf{f}}^3 p(\mathbf{x}_0) = \dots = L_{\mathbf{f}}^{(2K_0+1)} p(\mathbf{x}_0) = 0$. From Lemma 4.8 we know that $\gamma_{p,\mathbf{f}}(\mathbf{x}_0)$ is an odd number. Thus $\gamma_{p,\mathbf{f}}(\mathbf{x}_0) \geq 2K_0 + 1 + 2 > N_{p,\mathbf{f}}$, which contradicts Theorem 4.4.

(\Leftarrow) If \mathbf{x}_0 is not a common root of $L_{\mathbf{f}}^1 p(\mathbf{x}), L_{\mathbf{f}}^3 p(\mathbf{x}), \dots, L_{\mathbf{f}}^{(2K_0+1)} p(\mathbf{x})$, then $\gamma_{p,\mathbf{f}}(\mathbf{x}_0) < \infty$. By Lemma 4.7 we get $\mathbf{x}_0 \in \text{Trans}_{p,\mathbf{f}}$. \square

Now we are able to present a simplified version of Theorem 4.6 as follows.

Theorem 4.10 *Given a PDS $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and a parametric polynomial $p \triangleq p(\mathbf{u}, \mathbf{x})$. Let $\mathbf{u}_0 = (u_{10}, u_{20}, \dots, u_{t_0}) \in \mathbb{R}^l$. Then $p_{\mathbf{u}_0}$ is an RLF if and only if there exists $r_0 \in \mathbb{R}$, $r_0 > 0$ such that $(u_{10}, u_{20}, \dots, u_{t_0}, r_0)$ satisfies $\psi_{p,\mathbf{f}} \triangleq \psi_{p,\mathbf{f}}^1 \wedge \psi_{p,\mathbf{f}}^2 \wedge \psi_{p,\mathbf{f}}^3 \wedge \psi_{p,\mathbf{f}}^4$, where*

$$\psi_{p,\mathbf{f}}^1 \triangleq p(\mathbf{u}, \mathbf{0}) = 0, \tag{4.9}$$

$$\psi_{p,\mathbf{f}}^2 \triangleq \forall \mathbf{x}. (\|\mathbf{x}\|^2 > 0 \wedge \|\mathbf{x}\|^2 < r^2 \rightarrow p(\mathbf{u}, \mathbf{x}) > 0), \tag{4.10}$$

$$\psi_{p,\mathbf{f}}^3 \triangleq \forall \mathbf{x}. (\|\mathbf{x}\|^2 < r^2 \rightarrow L_{\mathbf{f}}^1 p(\mathbf{u}, \mathbf{x}) \leq 0), \tag{4.11}$$

$$\psi_{p,\mathbf{f}}^4 \triangleq \forall \mathbf{x}. (0 < \|\mathbf{x}\|^2 < r^2 \rightarrow L_{\mathbf{f}}^1 p(\mathbf{x}) \neq 0 \vee L_{\mathbf{f}}^3 p(\mathbf{x}) \neq 0 \vee \dots \vee L_{\mathbf{f}}^{(2K_0+1)} p(\mathbf{x}) \neq 0) \tag{4.12}$$

with K_0 defined in Lemma 4.9.

Proof By combining Theorem 3.4 with Lemma 4.9 we can get the results immediately. \square

In Theorem 4.10, constraints (4.9), (4.10) and (4.11) have relatively small sizes and can be solved by QE tools, while (4.12) can be handled more efficiently as an RRC problem of parametric semi-algebraic systems.

Definition 4.11 *A parametric semi-algebraic system (PSAS for short) is a conjunction of polynomial formulas of the following form:*

$$\begin{cases} p_1(\mathbf{u}, \mathbf{x}) = 0, \dots, p_r(\mathbf{u}, \mathbf{x}) = 0, \\ g_1(\mathbf{u}, \mathbf{x}) \geq 0, \dots, g_k(\mathbf{u}, \mathbf{x}) \geq 0, \\ g_{k+1}(\mathbf{u}, \mathbf{x}) > 0, \dots, g_l(\mathbf{u}, \mathbf{x}) > 0, \\ h_1(\mathbf{u}, \mathbf{x}) \neq 0, \dots, h_m(\mathbf{u}, \mathbf{x}) \neq 0, \end{cases} \tag{4.13}$$

where $r \geq 1, l \geq k \geq 0, m \geq 0$ and all p_i 's, g_i 's and h_i 's are in $\mathbb{Q}[\mathbf{u}, \mathbf{x}] \setminus \mathbb{Q}$.

² For $0 \leq r \in \mathbb{R}$, we have $\lfloor r \rfloor \in \mathbb{N}$ and $r - 1 < \lfloor r \rfloor \leq r$.

For a PSAS, the interesting problem is so-called *real root classification*, that is, to determine conditions on the parameters \mathbf{u} such that the given PSAS has certain prescribed number of distinct real solutions. Theories on real root classification of PSASs were developed in [37,38]. A computer algebra tool named DISCOVERER [36] was developed to implement these theories. See [22,35] for some applications of this technique in stability analysis of biological systems.

Given a PSAS \mathcal{P} with n indeterminates and s polynomial equations, it was argued in [5] that CAD-based QE on \mathcal{P} has complexity doubly exponential in n . In contrast, the RRC approach has complexity singly exponential in n and doubly exponential in t , where t is the dimension of the ideal generated by the s polynomials. Therefore RRC can dramatically reduce the complexity especially when t is much less than n .

For our problem, to solve (4.12) we can define a PSAS

$$\mathcal{P} \cong \begin{cases} L_{\mathbf{f}}^1 p(\mathbf{u}, \mathbf{x}) = 0, L_{\mathbf{f}}^3 p(\mathbf{u}, \mathbf{x}) = 0, \dots, L_{\mathbf{f}}^{(2K_0+1)} p(\mathbf{u}, \mathbf{x}) = 0 \\ -\|\mathbf{x}\|^2 > -r^2, \|\mathbf{x}\|^2 > 0 \end{cases}.$$

Then the command *RealRootClassification*($\mathcal{P}, 0$) in DISCOVERER returns conditions on \mathbf{u} and r such that \mathcal{P} has no solutions. In practice, \mathcal{P} can be constructed in a stepwise manner. That is, $L_{\mathbf{f}}^{(2i+1)} p(\mathbf{u}, \mathbf{x}) = 0$ for $0 \leq i \leq K_0$ can be added to \mathcal{P} one by one. Based on this idea, we present a *relaxed Lyapunov function generation algorithm* (Algorithm 1) implementing Theorem 4.10.

Algorithm 1 Relaxed Lyapunov Function Generation

```

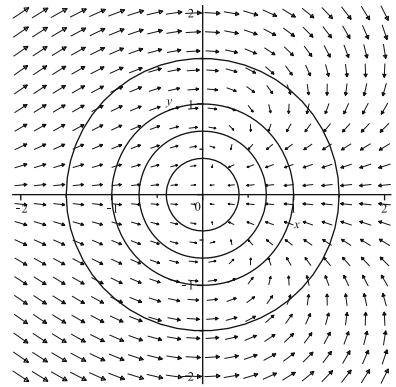
1 Input:  $\mathbf{f} \in \mathbb{Q}[x_1, \dots, x_n]^n$  with  $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ ,  $p \in \mathbb{Q}[u_1, \dots, u_t, x_1, \dots, x_n]$ 
2 Output:  $Res \subseteq \mathbb{R}^{t+1}$ 
3  $i := 1$ ;  $Res := \emptyset$ ;  $L_{\mathbf{f}}^1 p := (\nabla p, \mathbf{f})$ ;
4  $\mathcal{P} := \|\mathbf{x}\|^2 > 0 \wedge \|\mathbf{x}\|^2 < r^2$ ;
5  $Res^0 := \text{QE}(\psi_{p,\mathbf{f}}^1 \wedge \psi_{p,\mathbf{f}}^2 \wedge \psi_{p,\mathbf{f}}^3)$ ;
6 if  $Res^0 = \emptyset$  then
7   return  $\emptyset$ ;
8 else
9   repeat
10      $\mathcal{P} := \mathcal{P} \wedge L_{\mathbf{f}}^i p = 0$ ;
11      $Res := Res^0 \cap \text{RRC}(\mathcal{P}, 0)$ ;
12     if  $Res \neq \emptyset$  then
13       return  $Res$ ;
14     else
15        $L_{\mathbf{f}}^{i+1} p := (\nabla L_{\mathbf{f}}^i p, \mathbf{f})$ ;
16        $L_{\mathbf{f}}^{i+2} p := (\nabla L_{\mathbf{f}}^{i+1} p, \mathbf{f})$ ;
17        $i := i + 2$ ;
18   until  $L_{\mathbf{f}}^i p \in \langle L_{\mathbf{f}}^1 p, L_{\mathbf{f}}^2 p, \dots, L_{\mathbf{f}}^{i-1} p \rangle$ ;
19 return  $\emptyset$ ;
```

Remark 4.12 In Algorithm 1,

- $\psi_{p,\mathbf{f}}^1$, $\psi_{p,\mathbf{f}}^2$ and $\psi_{p,\mathbf{f}}^3$ in Line 5 are defined in (4.9), (4.10) and (4.11) respectively;
- QE in line 5 is done in a computer algebra tool like Redlog [8] or QEPCAD [3];
- RRC in line 11 stands for the *RealRootClassification* command in DISCOVERER;
- in Line 18 the loop test can be done by the *IdealMembership* command in MapleTM [18].

Termination of Algorithm 1 is guaranteed by Theorem 4.1 and Theorem 4.2; correctness of Algorithm 1 is guaranteed by Theorem 4.10.

Fig. 2 Vector field and Lyapunov surfaces in Example 5.1



5 Example

We illustrate our method for RLF generation using the following example.

Example 5.1 Consider the nonlinear dynamical system

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -x + y^2 \\ -xy \end{pmatrix} \quad (5.1)$$

with a unique equilibrium point $O(0, 0)$. We want to establish the asymptotic stability of O .

First, the linearization of (5.1) at O has the coefficient matrix

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$$

with eigenvalues -1 and 0 , so none of the principles of stability for linear systems can be applied. Besides, a homogeneous quadratic Lyapunov function $x^2 + axy + by^2$ for verifying asymptotic stability of (5.1) does not exist in \mathbb{R}^2 , because

$$\forall x \forall y. \left(\begin{array}{l} x^2 + y^2 > 0 \rightarrow (x^2 + axy + by^2 > 0) \\ \wedge 2x\dot{x} + ay\dot{x} + ax\dot{y} + 2by\dot{y} < 0 \end{array} \right)$$

is *false*. However, if we try to find an RLF in \mathbb{R}^2 using the simple template $p \hat{=} x^2 + ay^2$, then Algorithm 1 returns $a = 1$ in about 1 second on a desktop with a 2.66 GHz CPU and 3 GB memory. This means (5.1) has an RLF $x^2 + y^2$, and O is asymptotically stable. See Fig. 2 for an illustration.

From this example, we can see that RLFs really extend the class of functions that can be used for asymptotic stability analysis, and our method for automatically discovering RLFs can save us a lot of effort in finding conventional Lyapunov functions in some cases.

6 Conclusion

In this paper, we first generalize the notion of Lyapunov functions to *relaxed Lyapunov functions* by considering the higher order Lie derivatives of a smooth function along a smooth vector field. The main advantage of RLF is that it provides us more probability of certifying asymptotic stability. We also propose a method for automatically discovering polynomial RLFs for PDSs. Our method is relatively complete in the sense that we can generate all polynomial RLFs in the form of a predefined template, and thus can enumerate all potential polynomial RLFs

by enumerating all polynomial templates for a given PDS. We believe that our methodology could serve as a mathematically rigorous framework for the asymptotic stability analysis.

The main disadvantage of our approach is the high computational complexity: the complexity of the first-order quantifier elimination over the real closed fields is doubly exponential [7]. Although the performance of our illustrating example is good, the present approach cannot work on complex systems. Currently we are considering improving the efficiency of QE on a special class of first-order polynomial formulas, and it will be the main focus of our future work.

Acknowledgments The authors thank Professors Zhou Chaochen, Yang Lu, Xia Bican and Yu Wensheng for their helpful discussions on the topic. We also thank the anonymous referees for their valuable comments on the previous version, which helped us to improve the presentation so much.

References

1. Agrachev, A.A., Liberzon, D.: Lie-algebraic stability criteria for switched systems. *SIAM J. Control Optim.* **40**, 253–269 (2001)
2. Ahmadi, A.A., Parrilo, P.A.: On higher order derivatives of Lyapunov functions. In: American Control Conference 2011, pp. 1313–1314 (2011)
3. Brown, C.W.: QEPCAD. <http://www.usna.edu/Users/cs/qepcad/B/QEPCAD.html>
4. Butz, A.: Higher order derivatives of Lyapunov functions. *IEEE Trans. Automat. Contr.* **14**, 111–112 (1969)
5. Chen, Y., Xia, B., Yang, L., Zhan, N.: Generating polynomial invariants with DISCOVERER and QEPCAD. In: Jones, C., Liu, Z., Woodcock, J. (eds.) *Formal Methods and Hybrid Real-Time Systems*, LNCS, vol. 4700, pp. 67–82. Springer, Berlin (2007)
6. Cox, D., Little, J., O’Shea, D.: *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 2nd edn. Springer, Berlin (1997)
7. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. *J. Symb. Comput.* **5**(1-2), 29–35 (1988)
8. Dolzmann, A., Seidl, A., Sturm, T.: Redlog User Manual (Nov 2006), <http://redlog.dolzmann.de/downloads/>, edition 3.1, for Redlog Version 3.06 (Reduce 3.8)
9. Gao, B., Zhang, W.: A weaker criterion of asymptotic stability. *Comput. Math. Appl.* **56**(7), 1693–1698 (2008)
10. Giesl, P.: Construction of a local and global Lyapunov function using radial basis functions. *IMA J. Appl. Math.* **73**(5), 782–802 (2008)
11. Gurvits, L.: Stability of discrete linear inclusion. *Linear Algebra Appl.* **231**, 47–85 (1995)
12. Haddad, W.M., Chellaboina, V.: *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton University Press, New Jersey (2008)
13. Jiang, L.: On the asymptotic stability of equilibrium. *J. Math. Anal. Appl.* **301**(2), 378–383 (2005)
14. Johansson, M., Rantzer, A.: Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Automat. Contr.* **43**, 555–559 (1998)
15. Khalil, H.K.: *Nonlinear Systems*, 3rd edn. Prentice Hall, UK (2001)
16. Liberzon, D., Hespanha, J.P., Morse, A.: Stability of switched systems: a Lie-algebraic condition. *Syst. Control Lett.* **37**(3), 117–122 (1999)
17. Liu, J., Zhan, N., Zhao, H.: Computing semi-algebraic invariants for polynomial dynamical systems. In: EMSOFT’11, pp. 97–106. ACM press, New York (2011)
18. Maplesoft: Maple 14 User Manual. http://www.maplesoft.com/documentation_center/
19. Matrosov, V.M.: Method of vector Lyapunov functions of interconnected systems with distributed parameters. *Avtomatika i Telemekhanika.* **33**, 63–75 (1972) (in Russian)
20. Meigoli, V., Nikravesh, S.K.Y.: Application of higher order derivatives of Lyapunov functions in stability analysis of nonlinear homogeneous systems. In: IMECS’09 (2009). <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.3130>
21. Nersesov, S.G., Haddad, W.M.: On the stability and control of nonlinear dynamical systems via vector Lyapunov functions. In: *IEEE CDC’04*, vol. 4, pp. 4107–4112 (2004)
22. Niu, W., Wang, D.: Algebraic approaches to stability analysis of biological systems. *Math. Comput. Sci.* **1**, 507–539 (2008)
23. Papachristodoulou, A., Prajna, S.: Analysis of non-polynomial systems using the sums of squares decomposition. *Positive Polynomials in Control* **312**, 23–43 (2005)
24. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.3194> (2001)
25. Pettersson, S., Lennartson, B.: LMI for stability and robustness of hybrid systems. In: American Control Conference 1997, vol. 3, pp. 1714–1718 (1997)
26. Prajna, S., Papachristodoulou, A.: Analysis of switched and hybrid systems—beyond piecewise quadratic methods. In: American Control Conference 2003, vol. 4, pp. 2779 – 2784 (2003)
27. Ratschan, S., She, Z.: Providing a basin of attraction to a target region by computation of Lyapunov-like functions. In: *IEEE ICC’06*, pp. 1–5 (2006)

28. She, Z., Xia, B., Xiao, R.: A semi-algebraic approach for the computation of Lyapunov functions. In: CI'06. pp. 7–12 (2006)
29. Shim, H., Noh, D.J., Seo, J.H.: Common Lyapunov function for exponentially stable nonlinear systems. In: 4th SIAM Conference on Control and its Applications (1998)
30. Slotine, J.J., Li, W.: Applied Nonlinear Control. Prentice Hall, New Jersey (1991)
31. Tabuada, P.: Verification and Control of Hybrid Systems: A Symbolic Approach. Springer, Berlin (2009)
32. Tan, W., Packard, A.: Searching for control Lyapunov functions using sums of squares programming. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.69.1320>
33. Tarski, A.: A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley (1951)
34. Vu, L., Liberzon, D.: Common Lyapunov functions for families of commuting nonlinear systems. Syst. Control Lett. **54**(5), 405–416 (2005)
35. Wang, D., Xia, B.: Stability analysis of biological systems with real solution classification. In: ISSAC'05, pp. 354–361. ACM press, New York (2005)
36. Xia, B.: DISCOVERER: a tool for solving semi-algebraic systems. ACM Commun. Comput. Algebra **41**, 102–103 (2007)
37. Yang, L.: Recent advances on determining the number of real roots of parametric polynomials. J. Symb. Comput. **28**(1–2), 225–242 (1999)
38. Yang, L., Xia, B.: Real solution classifications of a class of parametric semi-algebraic systems. In: Algorithmic Algebra and Logic—Proceedings of the A3L 2005, pp. 281–289 (2005)