Complete algorithms for algebraic strongest postconditions and weakest preconditions in polynomial ODEs*

Michele Boreale
Università di Firenze
Dipartimento di Statistica, Informatica, Applicazioni (DiSIA) "G. Parenti"

Abstract

A system of polynomial ordinary differential equations (ODEs) is specified via a vector of multivariate polynomials, or vector field, F. A safety assertion $\psi \longrightarrow [F]\,\phi$ means that the trajectory of the system will lie in a subset ϕ (the postcondition) of the state-space, whenever the initial state belongs to a subset ψ (the precondition). We consider the case when ϕ and ψ are *algebraic varieties*, that is, zero sets of polynomials. In particular, polynomials specifying the postcondition can be seen as conservation laws implied by ψ. Checking the validity of algebraic safety assertions is a fundamental problem in, for instance, hybrid systems. We consider a generalized version of this problem, and offer an algorithm that, given a user specified polynomial set P and an algebraic precondition ψ, finds the largest subset of polynomials in P implied by ψ (relativized strongest postcondition). Under certain assumptions on ϕ, this algorithm can also be used to find the largest algebraic invariant included in ϕ and the weakest algebraic precondition for ϕ. Applications to continuous semialgebraic systems are also considered. The effectiveness of the proposed algorithm is demonstrated on several case studies from the literature.

Keywords: Ordinary differential equations, postconditions, preconditions, invariants, Gröbner bases.

1 Introduction

In recent years, there has been a renewed interest in computational models based on ordinary differential equations (ODEs), in such diverse fields as System Biology [2] and stochastic systems [36]. In particular, starting from [29], the field of hybrid systems has witnessed the emergence of a novel class of formal methods based on concepts from Algebraic Geometry – see e.g. [35, 30, 12] and references therein.

A system of ODEs can be seen as specifying the evolution over time, or *trajectory*, of certain variables of interest $x_1, ..., x_N$, describing for instance physical quantities. A fundamental problem in many fields is being able to prove or to disprove assertions of the following type. For each initial state in a given set $\psi \subseteq \mathbb{R}^N$ (the precondition), the resulting system's trajectory will lie in a given set $\phi \subseteq \mathbb{R}^N$ (the postcondition). This is a *safety assertion* that, using a notation akin to Platzer's Dynamic Logic, we can write as $\psi \longrightarrow [F]\,\phi$, where F is the vector field specifying the system. Evidently, safety assertions can be considered as a continuous counterpart of Hoare's triples in imperative programs – see [23].

Here we are primarily interested in the case where both ψ and ϕ are algebraic varieties, that is they are specified as zeros of (multivariate) polynomial sets, and the drifts f_i in $F = (f_1, ..., f_N)$ are polynomials themselves. Although (sets of) trajectories can rarely be represented *exactly* as algebraic varieties, these provide overapproximations that may be useful in practice. In a valid safety assertion, the polynomials

*Author's address: Michele Boreale, Università di Firenze, Dipartimento di Statistica, Informatica, Applicazioni (DiSIA) "G. Parenti", Viale Morgagni 65, I-50134 Firenze, Italy. E-mail: michele.boreale@unifi.it.

specifying the postcondition ϕ can be seen as system's *conservation laws* (for instance energy or mass conservation[1]) that are implied by the precondition ψ. Driven by the analogy with Hoare's triples, we find it natural to generalize the problem of checking the assertion $\psi \longrightarrow [F]\,\phi$ in two distinct ways. (1) Strongest postcondition: given a precondition ψ, find the smallest ϕ such that the assertion is valid; (2) weakest precondition: given a postcondition ϕ, find the largest ψ such that the assertion is valid. Problem (1) amounts to characterizing I_ψ, the set of *all* polynomials invariants (conservation laws) implied by ψ. The difficulty of (1) motivates the introduction of a relativized version of this problem: for a user specified polynomial set P, compute $P \cap I_\psi$. We call this a relativized strongest postcondition. Depending on P, computing this can be a lot easier than computing the whole I_ψ.

We offer a complete algorithm, called POST, that computes relativized strongest postconditions. In particular, this problem will be considered in the case where the set P is specified via a polynomial template. This way, for example, one can find at once all polynomial conservation laws of the system up to a given degree. As a byproduct of the POST algorithm, we also get the weakest algebraic invariant that implies all laws in $P \cap I_\psi$. The POST algorithm is based on building ascending chains of polynomial ideals: these represent, basically, more and more refined overapproximations of the (relativized) strongest postcondition. The proof of correctness and termination relies on a few concepts from Algebraic Geometry, notably Gröbner bases [9]. We will demonstrate the effectiveness of POST reporting the outcomes of a few experiments we have conducted on nontrivial systems taken from the literature, based on a Sagemath/Python implementation. Wherever possible, we will compare our results with those obtained by other authors.

Focusing on algebraic sets, as we do, does not necessarily imply that one is limited to algebraic safety properties: in fact, by considering a family of varieties depending on a set of parameters, one can often get a good approximation of a *semialgebraic* set of interest. In our case, the parameters will basically correspond to possible initial conditions of the system. In this way, we will show that safety verification of continuous systems with semialgebraic initial and unsafe sets is possible. Formally, this will require embedding the original system into a larger space, with the introduction of auxiliary variables, and relating these new variables to the original ones using a suitable precondition.

The present paper builds on our previous work on initial value problems [6] and on the conference paper [7]. We will discuss relations with these works, as well as with recent contributions from other authors dealing with invariant generation for polynomial ODES in the context of continuous and hybrid systems, notably [12] and [13].

Structure of the paper The rest of the paper is organized as follows. The necessary mathematical preliminaries, including polynomial differential equations and a few facts from Algebraic Geometry, are introduced in Section 2, while Section 3 introduces algebraic safety assertions and invariants. In Section 4, the main technical results are presented: the POST algorithm and the proof of (relative) completeness. A more algorithmic presentation of POST and computational issues connected with real radicals are discussed in Section 5. A few experiments on systems drawn from the literature are described in Section 6. An application to semialgebraic systems, together with further examples, is the subject of Section 7. Related works are reviewed and discussed in Section 8. For the sake of readability, a few technical proofs and some additional technical material have been confined to four separate Appendices (A, B, C and D).

[1]More precisely, conservation laws in our sense coincide with what are known in Physics as *first integrals of motion*, up to an additive constant.

2 Preliminaries

We review a few preliminary notions about ODEs, polynomials, Lie derivatives and Algebraic Geometry.

Polynomial ODEs Let us fix an integer $N \geq 1$ and a set of N distinct variables $x_1, ..., x_N$. We will denote by $\mathbf{x}$ the vector $(x_1, ..., x_N)$. We let $\mathbb{R}[\mathbf{x}]$ denote the set of multivariate polynomials in the variables $x_1, ..., x_N$ with coefficients in $\mathbb{R}$, and let p, q range over it. Here we regard polynomials as syntactic objects. Given an integer $d \geq 0$, by $\mathbb{R}_d[\mathbf{x}]$ we denote the set of polynomials of degree $\leq d$. As an example, $p = 2xy^2 + (1/5)wz + yz + 1$ is a polynomial of degree $\deg(p) = 3$, that is $p \in \mathbb{R}_3[x, y, z, w]$, with monomials xy^2, wz, yz and 1. Depending on the context, with a slight abuse of notation it may be convenient to let a polynomial denote the induced function $\mathbb{R}^N \to \mathbb{R}$, defined as expected: for $v \in \mathbb{R}^N$, $p(v) \in \mathbb{R}$ denotes the value obtained by evaluating p at v. In particular, x_i can be seen as denoting the projection on the i-th coordinate.

A (polynomial) *vector field* is a vector of N polynomials, $F = (f_1, ..., f_N)$, seen as a function $F : \mathbb{R}^N \to \mathbb{R}^N$. Throughout the paper, all definitions and statements refer to an arbitrarily fixed polynomial vector field F over a N-vector $\mathbf{x}$. The vector field F and an initial condition $v_0 \in \mathbb{R}^N$ together define an *initial value problem* $\mathbf{\Phi} = (F, v_0)$, often written in the following form

$$\mathbf{\Phi} : \begin{cases} \dot{\mathbf{x}}(t) & = & F(\mathbf{x}(t)) \\ \mathbf{x}(0) & = & v_0 \,. \end{cases} \tag{1}$$

The functions f_i in F are called *drifts* in this context. A *solution* to this problem is a differentiable function $\mathbf{x}(t) : D \to \mathbb{R}^N$, for some nonempty open interval $D \subseteq \mathbb{R}$ containing 0, which fulfills the above two equations, that is: $\frac{d}{dt}\mathbf{x}(t) = F(\mathbf{x}(t))$ for each $t \in D$ and $\mathbf{x}(0) = v_0$. By the Picard-Lindelöf theorem [1], there exists a nonempty open interval D containing 0, over which there is a *unique* solution, say $\mathbf{x}(t) = (x_1(t), ..., x_N(t))$, to the problem. In our case, as F is infinitely often differentiable, the solution is seen to be *analytic* in D: each $x_i(t)$ admits a Taylor series expansion in a neighborhood of 0. For definiteness, we will take the domain of definition D of $\mathbf{x}(t)$ to be the largest open interval where the Taylor expansion from 0 of each of the $x_i(t)$ converges (possibly $D = \mathbb{R}$). The resulting vector function of t, denoted $\mathbf{x}(t)$, is called the *time trajectory* of the system. Note that both the time trajectory and its domain of definition do depend in general on the initial v_0. We shall write them as $\mathbf{x}(t; v_0)$ and D_{v_0}, respectively, whenever we want to make this dependence explicit in the notation.

For any polynomial $p \in \mathbb{R}[\mathbf{x}]$, the function $p(\mathbf{x}(t)) : D \to \mathbb{R}$, obtained by composing p as a function with the time trajectory $\mathbf{x}(t)$, is analytic: we let $p(t)$ denote the extension of this function over the largest open interval of convergence (possibly coinciding with $\mathbb{R}$) of its Taylor expansion from 0. We will call $p(t)$ the *polynomial behaviour induced by p and by* the initial value problem (1). Again, fixing $N, \mathbf{x}$ and F once and for all, we shall write $p(t; v_0)$ when we want to emphasize the dependence of this function from the initial value v_0.

Lie derivatives Given a differentiable function $g : E \to \mathbb{R}$, for some open set $E \subseteq \mathbb{R}^N$, the *Lie derivative of g along F* is the function $E \to \mathbb{R}$ defined as: $\mathcal{L}_F(g) \triangleq \langle \nabla g, F \rangle = \sum_{i=1}^{N} (\frac{\partial g}{\partial x_i} \cdot f_i)$. The Lie derivative of the sum $h + g$ and product $h \cdot g$ functions obey the familiar rules

$$\mathcal{L}_F(h + g) = \mathcal{L}_F(h) + \mathcal{L}_F(g) \tag{2}$$

$$\mathcal{L}_F(h \cdot g) = h \cdot \mathcal{L}_F(g) + \mathcal{L}_F(h) \cdot g \,. \tag{3}$$

Note that $\mathcal{L}_F(x_i) = f_i$. Moreover if $p \in \mathbb{R}_d[\mathbf{x}]$ then $\mathcal{L}_F(p) \in \mathbb{R}_{d+d'}[\mathbf{x}]$, for some integer $d' \geq 0$ that depends on d and on F. This allows us to view the Lie derivative of polynomials along a polynomial field F as a purely syntactic mechanism, that is as a function $\mathcal{L}_F : \mathbb{R}[\mathbf{x}] \to \mathbb{R}[\mathbf{x}]$ that does not assume anything about the solution of (1). Informally, we can view p as a program, and taking Lie derivative of p can be interpreted as unfolding the definitions of the variables x_i's, according to the equations in (1) and to the formal rules for product and sum derivation, (2) and (3). More generally, we can define inductively $\mathcal{L}_F^{(0)}(p) \triangleq p$ and $\mathcal{L}_F^{(j+1)}(p) \triangleq \mathcal{L}_F(\mathcal{L}_F^{j}(p))$.

Example 1 The following system, borrowed from [13], will be used as a running example. Consider $N = 2$, $\mathbf{x} = (x, y)$ and the vector field $F = (y^2, xy)$. Let $p = x - y$. Examples of Lie derivatives are $\mathcal{L}_F(p) = y^2 - xy$ and $\mathcal{L}_F^{(2)}(p) = 2xy^2 - x^2y - y^3$.

The connection between Lie derivatives of p along F and the initial value problem (1) is given by the following equations, which can be readily checked. In the sequel, we let $p(v_0)$ denote the real number obtained by evaluating p at v_0:$p(t; v_0)_{|t=0} = p(v_0)$ and and $\frac{d}{dt} p(t; v_0) = (\mathcal{L}_F(p))(t; v_0)$. More generally, we have the following equation for the j-th derivative of $p(t)$ ($j = 0, 1, ...$): $\frac{d^j}{dt^j} p(t; v_0) = (\mathcal{L}_F^{(j)}(p))(t; v_0)$. In the sequel, we shall often abbreviate the syntactic Lie derivative $\mathcal{L}_F^{(j)}(p)$ as $p^{(j)}$, and shall omit the subscript $_F$ from $\mathcal{L}_F$ when clear from the context.

Algebraic Geometry preliminaries We quickly review a few notions from Algebraic Geometry that will be used throughout the paper. A comprehensive treatment of these concepts can be found for instance in Cox et al.'s excellent textbook [9]. A set of polynomials $I \subseteq \mathbb{R}[\mathbf{x}]$ is an *ideal* if: (1) $0 \in I$ and (2) $p_1, ..., p_m \in I$ and $h_1, ..., h_m \in \mathbb{R}[\mathbf{x}]$ implies $\sum_{i=1}^m h_i p_i \in I$. The ideal generated by a set $P \subseteq \mathbb{R}[\mathbf{x}]$ is $\langle\, P \,\rangle \triangleq \left\{\sum_{i=1}^m h_i p_i \;:\; m \geq 0 \text{ and } h_i \in \mathbb{R}[\mathbf{x}],\ p_i \in P \text{ for } i = 1, ..., m\right\}$. This is the smallest ideal containing P and as a consequence $\langle\, \langle\, P \,\rangle \,\rangle = \langle\, P \,\rangle$. Given an ideal I, a set P such that $I = \langle\, P \,\rangle$ is said to be a set of *generators* for I. Hilbert's basis theorem implies that: (a) any ideal $I \subseteq \mathbb{R}[\mathbf{x}]$ has a finite set of generators; (b) any infinite ascending chain of ideals $I_0 \subseteq I_1 \subseteq \cdots$ stabilizes in a finite number of steps (*ascending chain condition*). Once a total *monomial order* (e.g. lexicographic) is fixed, a multivariate version of polynomial division naturally arises – see [9] for the precise definition. A *Gröbner basis* of an ideal I (w.r.t. a fixed monomial order) is a finite set of generators G of I such that for any polynomial $p \in \mathbb{R}[\mathbf{x}]$ the *remainder* of the division of p by G, $r = p \bmod G$, enjoys following property: $p \in I$ iff $r = 0$. An alternative definition is that the leading monomial (greatest in the monomial order) of each $p \in I$ is divisible by the leading monomial of some $g \in G$. Given a Gröbner basis G of I, the *ideal membership* problem $p \in I$ can be decided[2] by just checking if $p \bmod G = 0$. Ideal inclusion $I \subseteq J$ can be decided similarly. There are algorithms (e.g. Buchberger's) that, given a finite P and a monomial order, compute a Gröbner basis G such that $\langle\, G \,\rangle = \langle\, P \,\rangle$. This computation is potentially expensive.

The geometric counterpart of polynomial sets are algebraic varieties. Given a set of polynomials $P \subseteq \mathbb{R}[\mathbf{x}]$, the set of points in $\mathbb{R}^N$ annihilating all of them

$$\mathbf{V}(P) \quad \triangleq \quad \{v \in \mathbb{R}^N : p(v) = 0 \text{ for each } p \in P\}$$

is the *algebraic*[3] *variety* represented by P. Ideals and algebraic varieties are connected as follows. For any set $A \subseteq \mathbb{R}^N$, the set of polynomials that vanish on A

$$\mathbf{I}(A) \quad \triangleq \quad \{p \in \mathbb{R}[\mathbf{x}] \;:\; p(v) = 0 \text{ for each } v \in A\}$$

is the ideal induced by A. Note that both $\mathbf{V}$ and $\mathbf{I}$ are inclusion reversing: $P \subseteq Q$ implies $\mathbf{V}(P) \supseteq \mathbf{V}(Q)$, and $A \subseteq B$ implies $\mathbf{I}(A) \supseteq \mathbf{I}(B)$. For A an algebraic variety and J an ideal, it is easy to see that $\mathbf{V}(\mathbf{I}(A)) = A$ and that $\mathbf{I}(\mathbf{V}(J)) \supseteq J$. We will have in general more than one ideal representing A.

[2] Provided the involved coefficients can be finitely represented, for instance are rational.

[3] Some authors use *affine*.

3 Algebraic safety assertions and invariants

We will be interested in *safety assertions* of the following type, where $\psi, \phi \subseteq \mathbb{R}^N$ are user specified algebraic varieties, which we call the *pre* and *postcondition*, respectively. Each of them is specified by a set of polynomials.

$$\text{Whenever } v_0 \in \psi \text{ then for each } t \in D_{v_0},\ \mathbf{x}(t; v_0) \in \phi. \tag{4}$$

The above assertion means that every trajectory starting in the precondition ψ will stay in the postcondition ϕ; hence necessarily $\psi \subseteq \phi$ for the assertion to hold. Using a notation akin to Platzer's Dynamic Logic's [23], the safety assertion (4) will be abbreviated as

$$\psi \quad \longrightarrow \quad [F]\,\phi\,. \tag{5}$$

A common technique for proving (5) is finding an algebraic variety χ such that $\psi \subseteq \chi \subseteq \phi$ and χ is an *algebraic invariant* for the vector field F, that is it satisfies $\chi \longrightarrow [F]\,\chi$. The invariance condition means that all trajectories starting in χ must remain in χ.

Let us now introduce two distinct generalizations of the problem of checking the safety assertion (5). These are the problems we will actually try to solve. In what follows, "finding" an algebraic variety means building a finite set of polynomials representing it. Also note that, for varieties, "smallest" means "strongest", and "largest" means "weakest".

Problem 1 (strongest postcondition) *Given an algebraic variety ψ, find ϕ_ψ, the smallest algebraic variety ϕ such that* (5) *is true.*

Note that ϕ_ψ always exists and is the intersection of all the varieties ϕ such that $\psi \longrightarrow [F]\,\phi$. Finding ϕ_ψ amounts to building (a basis of) an appropriate ideal I such that $\mathbf{V}(I) = \phi_\psi$. One such ideal is $I_\psi \triangleq \mathbf{I}(\phi_\psi)$. Unfortunately, computing I_ψ, or any other polynomial representation of ψ, appears to be computationally awkward. This motivates the introduction of a relativized version of the previous problem. In this version, a user specified set of polynomials P is used to tune the strength, hence precision, of the postcondition.

Problem 2 (strongest postcondition, relativized) *Given a polynomial set $P \subseteq \mathbb{R}[\mathbf{x}]$ and an algebraic variety ψ, find a finite representation of $P \cap I_\psi$.*

Of course, we have that $\mathbf{V}(P \cap I_\psi) \supseteq \mathbf{V}(I_\psi) = \phi_\psi$, which implies that $\psi \longrightarrow [F]\,\mathbf{V}(P \cap I_\psi)$. In other words, $P \cap I_\psi$ represents an overapproximation of the strongest postcondition. There is another meaningful way of generalizing the problem of checking (5).

Problem 3 (weakest precondition) *Given an algebraic variety ϕ, find ψ_ϕ, the largest algebraic variety ψ such that* (5) *is true.*

Let us now comment briefly on the relations existing between the above introduced problems. It is not difficult to see that Problem 1 and Problem 3 are both more general than the problem of checking (5) for *given ψ and ϕ*, based on the fact that one knows how to check inclusion between two varieties (see Section 2). The relativized Problem 2 too is more general than checking (5). Indeed, wanting to check the assertion $\psi \longrightarrow [F]\,\phi$, for *given* ψ and *given* $\phi = \mathbf{V}(Q)$, it is sufficient to let $P = Q$ in Problem 2 and then check if P is included in the computed $P \cap I_\psi$, that is if $P \subseteq I_\psi$.

Example 2 Let us reconsider the vector field F of Example 1. The variety $\psi = \mathbf{V}(\{p\}) = \mathbf{V}(\{x-y\})$ is the line $x = y$. Consider $\phi = \mathbf{V}(\{q\})$ where $q = x^2 - xy$. Let P be the set of all polynomials of degree ≤ 2. We can consider the following problems. (a) Decide whether $\psi \longrightarrow [F]\,\phi$; (b) find a finite representation of

$P \cap I_\psi$, that is all the conservation laws of degree at ≤ 2 that are satisfied, for each initial state in the line $x = y$ (relativized strongest postcondition); (c) find a finite representation of the largest algebraic variety ψ_ϕ such that $\psi_\phi \longrightarrow [F]\,\phi$ (weakest precondition). Note that solving (b) also yields a solution of (a).

Concerning the weakest precondition Problem 3, we note that a simple algorithm consists in collecting all algebraic conditions ensuring that the derivatives at any order of the polynomials specifying ϕ vanish. Specifically, assuming $\phi = \mathbf{V}(P)$ for a user defined, finite set $P \subseteq \mathbb{R}[\mathbf{x}]$, one considers the chain of sets $P_0 \stackrel{\triangle}{=} P$, $P_{j+1} = P_j \cup \mathcal{L}_F(P_j) = P_j \cup \{\mathcal{L}_F(p) : p \in P_j\}$, until the least m such that $\langle\ P_{m+1}\ \rangle = \langle\ P_m\ \rangle$, where the last equality can be checked via Gröbner bases computation. Then one has $\psi_\phi = \mathbf{V}(P_m)$. Termination and correctness of this algorithm can be easily derived from the results presented, for example, in [17, 12] (see also [20]). For the sake of completeness, we report a proof in Appendix A. In our experience, though, this simple algorithm tends to scale badly as the number of variables grows, so alternatives are worthwhile to consider.

In the following sections, we shall focus on Problem 2. In particular, we shall give a method, called POST, that works quite well in the case when the polynomial set P is specified by a polynomial template. Moreover, as a byproduct of this method, we will also get the weakest algebraic precondition for (and largest algebraic invariant included in) $\mathbf{V}(P \cap I_\psi)$, which can be used to address Problem 3 as well. Finally, POST will also give us a handle on the more general and difficult Problem 1.

4 The POST algorithm

Our goal is to give a method to effectively compute $P \cap I_\psi$, for user specified variety ψ and polynomial set P. Following a well-established tradition in the field of continuous and hybrid systems, we shall consider the case when the user specifies P via a polynomial *template*, which we review in the next paragraph. Throughout the section, whenever we consider a Gröbner basis over the polynomial ring $\mathbb{R}[\mathbf{a}, \mathbf{x}]$, we shall assume a lexicographic monomial ordering[4] such that $a_i > x_j$ for each i, j. This way, whenever G is a Gröbner basis of an ideal $I \subseteq \mathbb{R}[\mathbf{a}, \mathbf{x}]$, then $G \cap \mathbb{R}[\mathbf{x}]$ is a Gröbner basis of the ideal $I \cap \mathbb{R}[\mathbf{x}]$ (see [9, Ch.3,§1,Th.2]). In particular, for any finite set $G \subseteq \mathbb{R}[\mathbf{x}]$, we have that G is a Gröbner basis in $\mathbb{R}[\mathbf{a}, \mathbf{x}]$ if and only if it is in $\mathbb{R}[\mathbf{x}]$.

Polynomial templates Fix a tuple of $n \geq 1$ of distinct *parameters*, say $\mathbf{a} = (a_1, ..., a_n)$, disjoint from $\mathbf{x}$. Let $\mathbb{L}(\mathbf{a})$, ranged over by ℓ, be the set of *linear expressions* with coefficients in $\mathbb{R}$ and variables in $\mathbf{a}$; e.g. $\ell = 5a_1 + 42a_2 - 3a_3$ is one such expression[5]. A *template* [29] is a polynomial π in $\mathbb{L}(\mathbf{a})[\mathbf{x}]$, that is, a polynomial with linear expressions as coefficients. For example, the following is a template: $\pi = (5a_1 + (3/4)a_3)xy^2 + (7a_1 + (1/5)a_2)xz + (a_2 + 42a_3)$. Note that $\mathbb{L}(\mathbf{a})[\mathbf{x}] \subseteq \mathbb{R}[\mathbf{a}, \mathbf{x}]$, so, whenever convenient, we can consider a template as a polynomial in this larger ring. A *parameters valuation* is a vector

$$\lambda \quad = \quad (\mu_1, ..., \mu_n) \in \mathbb{R}^n\,.$$

Given such a λ, we will let $\ell[\lambda] \in \mathbb{R}$ denote the result of replacing each parameter a_i with μ_i, and evaluating the resulting expression; we will let $\pi[\lambda] \in \mathbb{R}[\mathbf{x}]$ denote the polynomial obtained by replacing each ℓ with $\ell[\lambda]$ in π. Given a set $S \subseteq \mathbb{R}^n$, we let $\pi[S]$ denote the set $\{\pi[\lambda] : \lambda \in S\} \subseteq \mathbb{R}[\mathbf{x}]$. The (formal) Lie derivative of π is defined as expected, once linear expressions are treated as constants; note that $\mathcal{L}(\pi)$ is still a template. It is easy to see that the following property is true: for each π and λ, one has $\mathcal{L}(\pi[\lambda]) = \mathcal{L}(\pi)[\lambda]$. This property extends as expected to the j-th Lie derivative ($j \geq 0$):

$$\mathcal{L}^{(j)}(\pi[\lambda]) \quad = \quad \mathcal{L}^{(j)}(\pi)[\lambda]\,. \tag{6}$$

[4] Any *elimination* ordering [9] for the parameters a_i could as well be considered.

[5] Note that linear expressions with a constant term, such as $2 + 5a_1 + 42a_2 - 3a_3$ are not allowed.

The algorithm Given a user specified algebraic variety ψ (the precondition) and a polynomial template π, describing $P = \pi[\mathbb{R}^n]$, our objective is to compute $P \cap I_\psi$. Let us call $p \in \mathbb{R}[\mathbf{x}]$ a *polynomial invariant* for F and v_0 if the function $p(t; v_0)$ is identically 0. A polynomial invariant expresses a law which is satisfied by the solution of the initial value problem (F, v_0), that is a conservation law. We will rely on the following two lemmas. The first one is just a reformulation of the definition of $I_\psi = \mathbf{I}(\phi_\psi)$. For the (easy) proof of the second, see e.g. [6].

Lemma 1 $I_\psi = \{p : p$ *is a polynomial invariant for each* $v_0 \in \psi\}$.

Lemma 2 *Let* $p \in \mathbb{R}[\mathbf{x}]$. *Then* p *is a polynomial invariant for the initial value* v_0 *if and only if for each* $j \geq 0$, $p^{(j)}(v_0) = 0$.

The above two lemmas suggest the following strategy to compute the set $\pi[\mathbb{R}^n] \cap I_\psi$. We should identify those parameter valuations $\lambda \in \mathbb{R}^n$, such that $\pi[\lambda]$ is a polynomial invariant for each $v_0 \in \psi$ (Lemma 1). That is, those λ's such that for each $j \geq 0$ and for each $v_0 \in \psi$, $\pi^{(j)}[\lambda](v_0) = 0$ (Lemma 2). Or, equivalently, $\pi^{(j)}[\lambda] \in \mathbf{I}(\psi)$ for each $j \geq 0$. For each $j \geq 0$, the last condition imposes certain constraints on λ, that is on the parameters of the template $\pi^{(j)}$. In order to make these constraints explicit, we shall rely on the following key lemma. A proof is reported in the Appendix B.

Lemma 3 *Let* $G \subseteq \mathbb{R}[\mathbf{x}]$ *be a Gröbner basis. Let* π *be a polynomial template and* $r = \pi \bmod G$. *Then* r *is linear in* $\mathbf{a}$. *Moreover, for each* $\lambda \in \mathbb{R}^n$, $\pi[\lambda] \bmod G = r[\lambda]$.

Fix a Gröbner basis G of $\mathbf{I}(\psi)$. By the above lemma, for a fixed j, $\pi^{(j)}[\lambda] \in \mathbf{I}(\psi)$ exactly when $r_j[\lambda] = 0$, where $r_j = \pi^{(j)} \bmod G$. By seeing r_j as a polynomial in $\mathbb{L}(\mathbf{a})[\mathbf{x}]$, the condition on λ

$$r_j[\lambda] \quad = \quad 0 \tag{7}$$

can be represented as a set of *linear* constraints on the parameters $\mathbf{a}$: indeed, a polynomial is zero exactly when all of its coefficients - in the present case, linear expressions in $\mathbf{a}$ - are zero[6]. This discussion leads to the method described below. We first give a purely mathematical description of the method, deferring the discussion of its computational aspects to Section 5.

The method can be seen as a generalization of the double chain algorithm of [6] to algebraic safety assertions. The basic idea is gradually refining the space of parameter valuations, starting from $\mathbb{R}^n$. More precisely, the algorithm builds two chains of sets: a descending chain of vector spaces, representing spaces of possible parameter valuations; and an (eventually) ascending chain of ideals, induced by those valuations. The ideal chain is used in the algorithm to detect the stabilization of the sequence. In order to state the correctness result in the most general form, fix an arbitrary ideal $I_0 \subseteq \mathbf{I}(\psi)$ and a Gröbner basis G of I_0. For each $j \geq 0$, let $r_j \triangleq \pi^{(j)} \bmod G$. For each $i \geq 0$, consider the sets

$$V_i \quad \triangleq \quad \{\lambda \in \mathbb{R}^n \,:\, r_j[\lambda] \text{ is the 0 polynomial, for } j = 0, ..., i\} \tag{8}$$

$$J_i \quad \triangleq \quad \left\langle \bigcup_{j=0}^{i} \pi^{(j)}[V_i] \right\rangle . \tag{9}$$

It is easy to check that each $V_i \subseteq \mathbb{R}^n$ is a vector space over $\mathbb{R}$ of dimension $\leq n$: this stems from the linearity in $\mathbf{a}$ of the r_j's. Now let $m \geq 0$ be the least integer such that the following conditions are *both* true:

$$V_{m+1} \quad = \quad V_m \tag{10}$$

$$J_{m+1} \quad = \quad J_m . \tag{11}$$

[6]For instance, if $\pi = (a_1 + a_2)x_1 + a_3x_2$ then $\pi[\lambda] = 0$ corresponds to the constraints $a_1 = -a_2$ and $a_3 = 0$

The algorithm returns (V_m, J_m), written $\textsc{post}_F(\psi, \pi) = (V_m, J_m)$; we shall omit the subscript $_F$ when the vector field F is clear from the context. Note that the integer m is well defined: indeed, $V_0 \supseteq V_1 \supseteq \cdots$ forms an infinite descending chain of finite-dimensional vector spaces, which must stabilize in finitely many steps. In other words, we can consider the least m' such that $V_{m'} = V_{m'+k}$ for each $k \geq 1$. Then $J_{m'} \subseteq J_{m'+1} \subseteq \cdots$ forms an infinite ascending chain of ideals, which must stabilize at some $m \geq m'$. Therefore there must be some index m such that (10) and (11) are both satisfied, and we choose the least such m.

Results Let us say that a set of polynomials J is an *invariant ideal* for the vector field F if it is an ideal and $\mathcal{L}_F(J) \triangleq \{\mathcal{L}_F(p) : p \in J\} \subseteq J$. The next theorem states the correctness and relative completeness of $\textsc{post}$. Informally, the algorithm outputs a space V such that $\pi[V] \subseteq I_\psi$, which is the largest such space if $I_0 = \mathbf{I}(\psi)$, and the smallest invariant ideal J witnessing this inclusion. We need a technical lemma, whose proof is reported in the Appendix B.

Lemma 4 *Let V_m, J_m be the sets returned by the* $\textsc{post}$ *algorithm. Then for each $j \geq 1$, one has $V_m = V_{m+j}$ and $J_m = J_{m+j}$.*

Theorem 1 (correctness and relative completeness of $\textsc{post}_F$) *Let ψ be an algebraic variety, let $I_0 \subseteq I_\psi$ be an ideal and G be a Gröbner basis of I_0. For any polynomial template π, let $\textsc{post}_F(\psi, \pi) = (V, J)$. Then*

(a) $\pi[V] \subseteq \pi[\mathbb{R}^n] \cap I_\psi$, with equality if $I_0 = \mathbf{I}(\psi)$;

(b) J is the smallest invariant ideal such that $J \supseteq \pi[V]$. Moreover, $J \subseteq I_\psi$.

Proof Let $(V, J) = (V_m, J_m)$ for some $m \geq 0$. Concerning part (a), we first note that, by virtue of Lemma 1 and Lemma 2, $\pi[\lambda] \in \pi[\mathbb{R}^n] \cap I_\psi$ if and only if for each $j \geq 0$, $(\pi[\lambda])^{(j)} = \pi^{(j)}[\lambda] \in \mathbf{I}(\psi)$ (here we have used property (6)). If $\lambda \in V_m = V_{m+1} = V_{m+2} = \cdots$ (here we are using Lemma 4), then by definition, for each $j \geq 0$, $r_j[\lambda] = (\pi^{(j)})[\lambda] \bmod G = (\pi[\lambda])^{(j)} \bmod G = 0$ (here we have used again property (6) and Lemma 3). That is, for each $j \geq 0$, $(\pi[\lambda])^{(j)} \in I_0 \subseteq \mathbf{I}(\psi)$. This implies (again by Lemma 1 and 2) that $\pi[\lambda] \in I_\psi$. Assume now that $I_0 = \mathbf{I}(\psi)$ and let λ such that $\pi[\lambda] \in \pi[\mathbb{R}^n] \cap I_\psi$, that is if for each $j \geq 0$, $(\pi[\lambda])^{(j)} = \pi^{(j)}[\lambda] \in \mathbf{I}(\psi)$. That is, being G a Gröbner basis of $\mathbf{I}(\psi)$, $\pi^{(j)}[\lambda] \bmod G = r_j[\lambda] = 0$ (the first equality here follows from Lemma 3), for each $j \geq 0$. This assertion, by definition, means that $\lambda \in V_j$ for each $j \geq 0$, hence in particular $\lambda \in V_m$.

Concerning part (b), to prove that J_m is the smallest invariant ideal including $\pi[V_m]$, it is enough to prove the following: (1) J_m is an invariant ideal, (2) $J_m \supseteq \pi[\mathbb{R}^n] \cap I_\psi$, and (3) for any invariant ideal I such that $\pi[\mathbb{R}^n] \cap I_\psi \subseteq I$, we have that $J_m \subseteq I$. We first prove (1), that J_m is an invariant ideal. Indeed, for each $\lambda \in V_m$ and each $j = 0, ..., m-1$, we have $\mathcal{L}(\pi^{(j)}[\lambda]) = \pi^{(j+1)}[\lambda] \in J_m$ by definition, while for $j = m$, since $\lambda \in V_m = V_{m+1}$, we have $\mathcal{L}(\pi^{(m)}[\lambda]) = \pi^{(m+1)}[\lambda] \in J_{m+1} = J_m$ (note that in both cases we have used property (6)). Concerning (2), note that $J_m \supseteq \pi[V_m] = \pi[\mathbb{R}^n] \cap I_\psi$, by virtue of part (a). Concerning (3), consider any invariant ideal $I \supseteq \pi[\mathbb{R}^n] \cap I_\psi$. We show by induction on $j = 0, 1, ...$ that for each $\lambda \in V_m$, $\pi^{(j)}[\lambda] \in I$; this will imply the wanted statement. Indeed, $\pi^{(0)}[V] = \pi[V] \in I_\psi \cap \pi[\mathbb{R}^n]$, as $\pi[V_m] \subseteq I_\psi$ by (a). Assuming now that $\pi^{(j)}[\lambda] \in I$, by invariance of I we have $\pi^{(j+1)}[\lambda] = \mathcal{L}(\pi^{(j)}[\lambda]) \in I$ (again, we have used here property (6)).

Finally, $J_m \subseteq I_\psi$ follows from the last statement and from the fact that I_ψ, as clearly seen from Lemma 1 and 2, is an invariant ideal. □

Example 3 We reconsider the vector field F of Example 1. Let us consider $\psi = \mathbf{V}(\{x - y\})$. A Gröbner basis of $\mathbf{I}(\psi)$ is just $G = \{x - y\}$. We let π be the complete template of degree 2 (described below). We build the chain of sets V_i, J_i, for $i = 0, 1, ...$, with the help of the computer algebra system SageMath [28]. Below, $\lambda = (v_1, ..., v_6) \in \mathbb{R}^6$ denotes a generic parameters valuation.

- $\pi = a_6xy + a_5y^2 + a_4x^2 + a_3y + a_2x + a_1$ and $r_0 = \pi \bmod G = a_4y^2 + a_5y^2 + a_6y^2 + a_2y + a_3y + a_1$. Thus $V_0 = \{v : v_4 = -v_5 - v_6,\ v_2 = -v_3,\ v_1 = 0\}$ and $J_0 = \langle\ \pi[V_0]\ \rangle$;
- $\pi^{(1)} = a_6x^2y - 2a_6xy^2 + a_6y^3 + a_3xy - a_3y^2$ and $r_1 = \pi^{(1)} \bmod G = 0$. Thus $V_1 = V_0$. Moreover $\pi^{(1)}[V_0] \subseteq J_0$, which implies $J_1 = \langle\ \pi[V_0] \cup \pi^{(1)}[V_0]\ \rangle = J_0$.

Thus $\textsc{post}(\psi, \pi) = (V_0, J_0)$. A Gröbner basis of J_0 is $G_0 = G$.

Remark 1 (result template) Given a template π and $\lambda \in \mathbb{R}^n$, checking if $\pi[\lambda] \in \pi[V]$ is equivalent to checking if $\lambda \in V$: this can be effectively done knowing a basis B of the vector space V (see Section 5). In practice, it is sometimes more convenient to represent the whole set $\pi[V]$ returned by $\textsc{post}$ compactly in terms of a *new* m-parameters result template π' such that $\pi'[\mathbb{R}^m] = \pi[V]$. For instance, in the previous example, the result template $\pi' = a_1(y^2 - x^2) + a_2(xy - x^2) + a_3(y - x)$ represents $\pi[V_0]$, in the precise sense that $\pi[V_0] = \pi'[\mathbb{R}^3]$. The result template π' can in fact be built directly from π, by propagating the linear constraints on $\mathbf{a}$ (7) as they are generated. This will be explicitly described when discussing the algorithmic presentation in Section 5.

Note that, while typically the user will be interested in $\pi[V]$, the ideal J as well may contain useful information, such as higher order, nonlinear conservation laws. The theorem below is about the meaning of J as an invariant and as a precondition. The theorem relies on Theorem 1 and on the following lemma, of independent interest, saying that invariant ideals, on the polynomial side, precisely correspond to algebraic invariants. The proofs of both the lemma and the theorem are reported in the Appendix B.

Lemma 5 *Consider a set $\chi \subseteq \mathbb{R}^N$. Then χ is an algebraic invariant for the vector field F if and only if there is an invariant ideal J for F such that $\chi = \mathbf{V}(J)$.*

Theorem 2 (weakest algebraic invariant and precondition) *For an algebraic variety ψ and a polynomial template π, let $\textsc{post}_F(\psi, \pi) = (V, J)$ and $\phi = \mathbf{V}(\pi[V])$. Then*

(a) $\mathbf{V}(J)$ *is the largest algebraic invariant included in ϕ; and*

(b) $\mathbf{V}(J)$ *is the weakest precondition of ϕ.*

We stress that Theorem 2(b) provides a means to solve Problem 3 (weakest precondition) via the $\textsc{post}$ algorithm. In fact, given ϕ, it suffices to consider *any* precondition ψ and template π such that $\textsc{post}(\psi, \pi) = (V, J)$ and $\mathbf{V}(\pi[V]) = \phi$: then $\mathbf{V}(J)$ is ϕ's weakest precondition. In particular, ψ may consists of a singleton. Also note that the theorem does not require the equality $I_0 = \mathbf{I}(\psi)$. An example of application of this technique is given below. Other examples will be discussed in Section 6 (see in particular the Kepler laws example).

Example 4 We reconsider the vector field F of Example 1. Let $\phi = \mathbf{V}(\{q\})$ be given, where $q = x^2 - xy$. We want to compute the weakest precondition ψ_ϕ via $\textsc{post}$. We choose the trivial precondition $\psi = \{(0,0)\} = \mathbf{V}(\{x, y\})$: both $x(t;\, (0,0))$ and $y(t;\, (0,0))$ are identically 0, making $\psi \rightarrow [F]\,\phi$ a valid assertion. Now choosing $\pi = a_1 \cdot q$ and running $\textsc{post}$, we obtain $\textsc{post}(\psi, \pi) = (V, J)$, where $V = \mathbb{R}$ and necessarily $\mathbf{V}(\pi[V]) = \phi$, and $J = \langle\ \{xy^2 - y^3\ ,\ x^2 - xy\}\ \rangle$. By Theorem 2(b), $\psi_\phi = \mathbf{V}(J) = \mathbf{V}(\{x - y\})$.

Finally, the following result of theoretical interest, shows that the whole ideal I_ψ as well can be characterized in terms of the $\textsc{post}$ algorithm. For any $k \geq 0$, the *complete polynomial template* of degree k over a set of variables X is $\pi \triangleq \sum_\alpha a_\alpha \alpha$, where α ranges over all monomials of degree $\leq k$ on the variables in X, and a_α ranges over distinct parameters.

Corollary 1 (characterization of I_ψ) *Let ψ be an algebraic variety. Let $k \geq 0$, π_k be the complete template of degree k over the variables in* $\mathbf{x}$ *and $(V, J) = \textsc{post}(\psi, \pi_k)$. For k large enough, $J = I_\psi$.*

Proof By Hilbert's basis theorem, there is a finite set of polynomials P such that $I_\psi = \langle\ P\ \rangle$. Therefore I_ψ is the smallest ideal containing P, and is also an invariant ideal. Now let k be the maximum degree of polynomials in P, let π_k be the complete template of degree k over all variables, and n the number of parameters in π_k. As $P \subseteq \pi_k[\mathbb{R}^n]$ and $P \subseteq I_\psi$, we have $\pi_k[\mathbb{R}^n] \cap I_\psi \supseteq P$. Now let $(V, J) = \textsc{post}(\psi, \pi_k)$. By Theorem 1(b), $J \supseteq \pi_k[V] = \pi_k[\mathbb{R}^n] \cap I_\psi \supseteq P$, hence $J \supseteq I_\psi$. On the other hand, again by Theorem 1(b), $J \subseteq I_\psi$. Therefore $J = I_\psi$. □

We leave open the problem of computing a lower bound on the degree k that is needed to recover I_ψ. We end the section with a remark on the expressive power of algebraic varieties.

Remark 2 (expressive power) Algebraic varieties can in general provide only overapproximations of sets of initial states and trajectories. However, the expressive power of algebraic varieties can often be significantly enhanced by introducing auxiliary, or *ghost* variables, in the terminology of Platzer [24]. These variables are used to express properties of interest. We have found particularly interesting the case when ghost variables are used encode *generic* initial values of the system: apparently, keeping track of such values allows for more expressive polynomial invariants. This is illustrated by the example below. We will put this technique into use in Section 6 and, in a more systematic way, in Section 7, where we shall deal with semialgebraic systems.

Example 5 Consider again the system of Example 1. With no constraints on the initial states, that is with $\psi = \mathbb{R}^2$, the strongest postcondition is quite easily seen to be the trivial $\phi = \mathbb{R}^2$, that is $I_\psi = \{0\}$. We build now a new system by introducing two new variables x_0, y_0, together with the corresponding equations $\dot{x}_0 = 0$ and $\dot{y}_0 = 0$: this means they represent (generic) constants – in effect, parameters. We consider the precondition $\psi = \mathbf{V}(\{x - x_0, y - y_0\})$, meaning that x_0 and y_0 represent the (generic) initial values of x and y, respectively. Using a complete template π of degree 2, we now get the nontrivial result $\textsc{post}(\psi, \pi) = (V, J)$ with $J = \langle\ \{x_0^2 - y_0^2 - x^2 + y^2\}\ \rangle$ and $\pi[V] = \pi'[\mathbb{R}]$, where $\pi' = a_1(x^2 - x_0^2 - y^2 + y_0^2)$. Here J represents a valid nontrivial invariant for every instantiation of x_0, y_0.

5 Computational aspects of POST

We consider here some important computational aspects of the POST algorithm. We will first derive a more algorithmic presentation of the abstract procedure introduced in Section 4; then discuss issues related to selecting an appropriate ideal $I_0 \subseteq \mathbf{I}(\psi)$ and a Gröbner G basis for it.

5.1 Algorithmic presentation

When it comes to the effective implementation of POST, a first aspect to consider is how to finitely represent the sets V_i, J_i. Each subspace V_i is spanned by a finite basis B_i: this basis is implicitly represented by the linear constraints on $\mathbf{a}$ in (8). From (9) it is then easy to check that $\bigcup_{j=0}^{i} \pi^{(j)}[B_i]$ is a basis of J_i. The termination conditions $V_i = V_{i+1}$ and $J_i = J_{i+1}$ can also be checked effectively. In particular, checking $J_i = J_{i+1}$ involves computing a Gröbner basis H of J_i, a potentially expensive operation, and checking if $\pi^{(i+1)}[B] \subseteq J_i$. Fortunately, this need not be done at each step, but only if actually $V_i = V_{i+1}$, the latter a relatively inexpensive check. This discussion leads to Algorithm 5.1; note that program blocks are defined by indentation. There, we make use of a few auxiliary variables and functions, as detailed below.

1. S is an initially empty list of polynomial templates, used to collect that successive Lie derivatives of the input template. The functions first($\cdot$) and append($\cdot$), defined on lists, have the obvious interpretation, with the proviso that first([ ]) $\triangleq 0$, the zero polynomial template. For $B \subseteq \mathbb{R}^n$, and $S = [\pi_1, ..., \pi_m]$, we let $S[B]$ denote the *set* of polynomials $\bigcup_{j=1}^{m} \pi_j[B]$.

Algorithm 1 POST

Input: F a vector field, π a n-parameters template, $G \subseteq \mathbf{I}(\psi)$ a Gröbner basis

Output: π' a template, H a Gröbner basis (s.t. $\pi'[\mathbb{R}^n] = \pi[V]$ and $J = \langle\ H\ \rangle$, where $\text{POST}_F(\psi, \pi) = (V, J)$)

$S = [\,]$
$\sigma = \emptyset$
while true do
  $r = \pi \bmod G$
  $\gamma = \text{solve}(r = 0)$
  if $\gamma = \emptyset$ **then** ▷ Check if $V_{i+1} = V_i$
    $B = \text{basis}(\sigma)$
    $H = \text{Gröbner}(S[B])$
    if $\pi[B] \subseteq \langle\ H\ \rangle$ **then** ▷ Check if $J_{i+1} = J_i$
      return $(\text{first}(S), H)$
  else
    $\pi = \gamma(\pi)$
    $S = \gamma(S)$
    $\sigma = \gamma \star \sigma$
  $S = \text{append}(S, \pi)$
  $\pi = \mathcal{L}_F(\pi)$

2. σ, γ are substitutions, encoding linear constraints existing among the parameters. Explicitly, a substitution f is a finite partial map from $\{a_1, ..., a_n\}$ (parameters) to $\mathbb{L}[\mathbf{a}]$ (linear expressions), such that no parameter in $\text{dom}(f)$ occurs in any $\ell \in \text{range}(f)$. We write $\sigma(c)$ for the result of applying σ to every parameter occurring in (the expression, set,...) c. In particular, γ accumulates all the constraints seen so far. The composition $\gamma \star \sigma$ represents the sequential application of σ then γ: for the present case[7], this substitution, seen as a set of pairs, is defined as $\gamma \star \sigma \triangleq \gamma \cup \{(a_i, \gamma(\sigma(a_i)))\ :\ a_i \in \text{dom}(\sigma)\}$.

3. $\text{basis}(\sigma)$ returns a linear basis for the subspace of $\mathbb{R}^n$ implicitly defined by the constraints in σ. Explicitly, letting $e_j \in \mathbb{R}^n$ denote the j-th canonical basis vector ($1 \leq j \leq n$), seen as a parameter valuation, we have (below 0 denotes the null vector in $\mathbb{R}^n$):

$$\text{basis}(\sigma) \quad \triangleq \quad \{(\,\sigma(a_1)[e_j], ..., \sigma(a_n)[e_j]\,)\ :\ j = 1, ..., n\} \setminus \{0\} \subseteq \mathbb{R}^n\,.$$

   As an example, if $n = 4$ and $\sigma = \{(a_1\,,\, 2a_2 - a_3)\,\}$ then $\sigma(\mathbf{a}) = (2a_2 - a_3, a_2, a_3, a_4)$ and $\text{basis}(\sigma) = \{(2, 1, 0, 0)\,,\, (-1, 0, 1, 0),\ (0, 0, 0, 1)\}$.

4. $\text{solve}(r = 0)$ returns a (minimal) substitution γ such that $\gamma(r) = 0$, the zero polynomial. We insist that $|\text{dom}(\gamma)|$ be minimal, that is eliminate as few parameters as possible. In linear algebraic terms, let $\ell_1, ..., \ell_K \in \mathbb{L}(\mathbf{a})$ be the distinct coefficients of r, where $\ell_i = \sum_{j=1}^{n} c_{ij} a_j$. Let C be the $K \times n$ coefficients matrix of the c_{ij}'s. A parameter valuation $\lambda \in \mathbb{R}^n$ makes $r[\lambda]$ the null polynomial if and only if λ^T is a solution of the linear system, in the variables $\mathbf{a}$, $C\mathbf{a}^T = 0$. We insist that γ describe the whole space $U \subseteq \mathbb{R}^n$ of solutions of this system, that is $\text{basis}(\gamma)$ spans U. As U has dimension $n - \text{rk}(C)$, this is equivalent to say that $\gamma(r) = 0$ and $|\text{dom}(\gamma)| = \text{rk}(C)$.

5. Gröbner($\cdot$), applied to a set of polynomials, returns a Gröbner basis for the ideal generated by that set.

[7] Here we use the fact that $\text{dom}(\sigma) \cap Z = \emptyset$, where Z is the set of parameters that appear in $\text{dom}(\gamma)$ or $\text{range}(\gamma)$.

The exact theoretical complexity of this algorithm is difficult to characterize, even assuming, as we do here, that the basis G s.t. $\langle G \rangle = I_0 \subseteq \mathbf{I}(\psi)$ has been precomputed. But one can at least work out some very conservative bounds, as follows. Let us denote by d the sum of the degree of π and of the maximal degree of polynomials in F, and by N the number of variables. We note that: (a) each step potentially involves the computation of a Gröbner basis, for which known algorithms have an exponential worst case time complexity upper bounded approximately by $O(D^{2^N})$, where D is the maximum degree in the input polynomial set (see [9]); (b) the maximum degree D of the derivatives $\pi^{(j)}$'s occurring in S, for $0 \leq j \leq m+1$, is bounded by $(m+1)d$. Overall, this gives a worst case time complexity of approximately $O(m^{2^N+1}d^{2^N})$. Finally, according to a result in [20], the number of steps m before stabilization of an ascending chain of ideals generated by successive Lie derivatives is upper bounded by $d^{N^{O(N^2)}}$. One should stress that these are very conservative bounds. A SageMath/Python [28] implementation strictly adhering to Algorithm 5.1 is available[8] that works reasonably well in a number of cases of practical interest; see Section 6. SageMath directly provides an efficient implementation of the main auxiliary functions, such as `Solve()` and `groebner_basis()`.

5.2 The choice of I_0 and the real radical problem

A crucial aspect in the POST algorithm is the choice of the ideal $I_0 \subseteq \mathbf{I}(\psi)$ and the computation of a Gröbner basis G for it. In the following discussion, we fix the following notation and terminology:

- $\psi = \mathbf{V}(Q)$, the variety generated by a finite (user specified) set $Q \subseteq \mathbb{R}[\mathbf{x}]$;
- $I = \langle Q \rangle$, the ideal generated by Q;
- $\mathbf{I}(\psi) = \{p : p(v) = 0 \text{ for each } v \in \psi\} \supseteq I$, the *real radical* of I.

In the statement of Theorem 1(a), equality, hence completeness, is guaranteed if $I_0 = \mathbf{I}(\psi)$ is the real radical of I; otherwise only soundness holds in general. Unfortunately, at present computing a set of generators G for a real radical appears to be, in the general case, computationally unfeasible. Below, we shall briefly discuss the state of the art concerning this problem, then a special case where this computation is feasible, and what are the alternatives in cases where it is not.

A classical algorithm for computing real radicals is due to Neuhaus [19]. This is also implemented as part of Singular's `realrad` library [10, Sect.D.4.16], accessible via SageMath [28]. The worst case asymptotic complexity of this algorithm is very high: $D^{2^{O(N^2)}}$ (exact) arithmetic operations, where D is the maximum total degree of the polynomials in the set Q. Over the years there have been improvements: let us just mention the algorithm by Lasserre et al., based on semi-definite relaxations but limited to ideals with zero dimensional varieties [16]; and the recent probabilistic method by El Din et al. [11], which lowers the asymptotic complexity to $|Q|^{O(1)}(ND)^{O(Nr2^r)}$, where r is the dimension of the variety ψ. Despite these advancements, the resulting algorithms appear to be still totally impractical but for very simple instances: [11] mentions an example with $N = 9$ variables and maximum total degree $D = 3$ which is beyond Singular's capabilities and requires 800s with their implementation.

Next, we consider a simple special case of practical interest, where it is trivial to build the real radical. This case is relevant to the auxiliary variables method mentioned in Remark 2, and will be put into systematic use when dealing with semialgebraic systems (Section 7).

Proposition 1 *Let $\mathbf{x} = (x_1, ..., x_N)$, let $1 \leq m < N$ and assume $Q = \{x_i - g_i : i = 1, ..., m\}$, where $g_i \in \mathbb{R}[x_{m+1}, ..., x_N]$. Then Q is set of generators of $\mathbf{I}(\psi)$.*

[8] `https://github.com/micheleatunifi/postconditions/blob/master/Post.py`

Proof Fix a lexicographic monomial order such that $x_i > x_j$ whenever $i \leq m$ and $j \geq m+1$. W.r.t. this order, Q is a Gröbner basis for $I = \langle\, Q \,\rangle$: indeed, take any $0 \neq p \in I$ and assume by contradiction that $\mathrm{LM}(p)$ (the leading monomial of p) is not divisible by any leading monomial x_i in Q; that is, $\mathrm{LM}(p)$ does not contain any x_i with $i \leq m$. This would imply, by definition of lex order, that p does not contain any such x_i, that is $p \in \mathbb{R}[x_{m+1}, ..., x_N]$. Then for each $v = (v_{m+1}, ..., v_N)$, we can consider $w = (g_1(v), ..., g_m(v), v_{m+1}, ..., v_N) \in \psi = \mathbf{V}(Q)$, implying $p(w) = p(v) = 0$. In conclusion, as $p(v) = 0$ for each v, p is the zero polynomial, contradicting the assumption.

Now let us check that $\mathbf{I}(\psi) = \langle\, Q \,\rangle = I$. Clearly $\mathbf{I}(\psi) \supseteq \langle\, Q \,\rangle$. On the other hand, consider any $p \in \mathbf{I}(\psi)$ and let $p = q + r$, where $r = p \bmod Q$ and $q \in I$. By the above assumptions on Q and by definition of remainder, no variable x_i with $i \leq m$ can occur in r, that is $r \in \mathbb{R}[x_{m+1}, ..., x_N]$. Now assume by contradiction $r \neq 0$, so there is $v = (v_{m+1}, ..., v_N)$ such that $r(v) = a \neq 0$. Then $w = (g_1(v), ..., g_m(v), v_{m+1}, ..., v_N) \in \psi$, hence $p(w) = 0$; yet $p(w) = q(w) + r(v) = 0 + a \neq 0$, which is a contradiction. Therefore, we can set $G = Q$ in this case. □

When computing $\mathbf{I}(\psi)$ is not feasible, there is little alternative to replacing it with some easy to compute ideal $I_0 \subseteq \mathbf{I}(\psi)$: as discussed above, this preserves soundness of the approach, although completeness is lost in general. A practical choice might be considering $\sqrt{I}$, the *complex* radical ideal of I

$$\sqrt{I} \quad \triangleq \quad \{p \in \mathbb{C}[\mathbf{x}] \;:\; p^m \in I \text{ for some } m > 0\}$$

where $\mathbb{C}$ denotes the complex field. By Hilbert's strong Nullstellensatz [9, Ch.4,§1.2,Th.6], in $\mathbb{C}[\mathbf{x}]$ we have

$$\sqrt{I} \quad = \quad \mathbf{I}(\mathbf{V}_{\mathbb{C}}(I))$$

where $\mathbf{V}_{\mathbb{C}}(I) = \{v \in \mathbb{C}^N \;:\; p(v) = 0 \text{ for each } p \in I\}$ is the complex algebraic variety induced by I. As $\mathbf{V}_{\mathbb{C}}(I) \supseteq \mathbf{V}(I)$, one has

$$\sqrt{I} \cap \mathbb{R}[\mathbf{x}] \quad \subseteq \quad \mathbf{I}(\psi)\,. \tag{12}$$

Therefore, we can set $I_0 = \sqrt{I} \cap \mathbb{R}[\mathbf{x}]$ and take as G any Gröbner basis of $\sqrt{I}$; note that, as $I \subseteq \mathbb{R}[\mathbf{x}]$, necessarily $G \subseteq \mathbb{R}[\mathbf{x}]$. The inclusion (12) is in general strict. As an example, consider $Q = \{x^2 + 1\}$, hence $\mathbf{V}(Q) = \emptyset$: then trivially $\mathbf{I}(\psi) = \mathbb{R}[\mathbf{x}]$. On the other hand, $\mathbf{V}_{\mathbb{C}}(\{x^2 + 1\}) = \{\iota, -\iota\}$ hence $\sqrt{I} \cap \mathbb{R}[\mathbf{x}] \neq \mathbb{R}[\mathbf{x}]$; for example $x \notin \sqrt{I}$.

The problem of computing a set of generators for the complex real radical of I is well understood, and there exist well-known algorithms to this purpose: in particular, those by Krick and Logar [14] and by Laplagne [15]. Although the worst-case complexity of these methods is doubly exponential in the number of variables, they often work reasonably well and a number of implementations are offered in computer algebra systems, including those in Singular's `radical` library [10, Sect.D.4.14.7]. We rely on this library in our implementation.

6 Experiments

We report below the outcomes of three experiments we have conducted, applying the post algorithm to challenging systems taken from the literature. The execution times reported below are for an implementation in Python under SageMath [28], running on a Core i5 machine[9]. Wherever possible, we compare our results with those obtained by other authors.

[9] Code and examples available at `https://github.com/micheleatunifi/postconditions/blob/master/Post.py`.

Collision avoidance We consider the two-aircraft dynamics used to study collision avoidance, discussed in many papers on hybrid systems [30, 17, 12]. The model is described by the equations below, where the variables have the following meaning: (x_1, x_2) and (y_1, y_2) represent the Cartesian coordinates of aircraft 1 and 2, respectively; (d_1, d_2) and (e_1, e_2) their velocities; applying the technique discussed in Remark 2, we also introduce the auxiliary variables (parameters, hence 0 derivative) ω_1 and ω_2, representing the angular velocities of the two aircrafts, and $x_{10}, x_{20}, y_{10}, y_{20}$, $d_{10}, d_{20}, e_{10}, e_{20}$, representing generic initial values of the corresponding variables. Overall, the system's vector field F_1 consists of 18 polynomials over as many variables (including the auxiliary ones).

$$\begin{array}{rclrclrclrcl} \dot{x}_1 &=& d_1 & \dot{y}_1 &=& e_1 & \dot{d}_1 &=& -\omega_1 d_2 & \dot{e}_1 &=& -\omega_2 e_2 \\ \dot{x}_2 &=& d_2 & \dot{y}_2 &=& e_2 & \dot{d}_2 &=& -\omega_1 d_1 & \dot{e}_2 &=& -\omega_2 e_1 \,. \end{array}$$

We consider the precondition ψ that assigns to each non constant variable the parameter corresponding to its (generic) initial value: $\psi = \mathbf{V}(\{x_1 - x_{10}, x_2 - x_{20}, ...\})$. Note that $G = \{x_1 - x_{10}, x_2 - x_{20}, ...\}$ is a set of generators for $\mathbf{I}(\psi)$, and in fact a Gröbner basis w.r.t. the lexicographic order (Proposition 1). We then consider a complete template π of degree 2 over all the system's variables: π is a linear combination of $n = 190$ monomials that uses as many parameters. We then run $\textsc{post}(\psi, \pi)$, which returns, after $m = 3$ iterations and about 16s, a pair (V, J). The vector space V corresponds to a result template with 10 parameters, $\pi' = \sum_{i=1}^{10} a_i \cdot p_i$. The instances of π' are therefore all and only the system's polynomial invariants of degree ≤ 2, starting from a fully generic precondition (Theorem 1(a)). These include all the polynomial invariants mentioned in [30, 17], and several new ones, like the following

$$-x_{10}d_{10} - x_{20}d_{20} + d_{10}x_1 + d_{20}x_2 + x_{10}d_1 - x_1d_1 + x_{20}d_2 - x_2d_2 \,.$$

Let $\phi \triangleq \mathbf{V}(\pi'[\mathbb{R}^n])$ be the variety defined by the result template π'. The invariant ideal J returned by the algorithm represents the weakest algebraic precondition $\chi \triangleq \mathbf{V}(J)$ such that $\chi \longrightarrow [F_1]\phi$: in other words, the largest algebraic precondition for which all instances of π' are polynomial invariants (Theorem 2(b)). Moreover, χ is also the weakest algebraic invariant included in ϕ (Theorem 2(a)). A Gröbner basis of J consists of 12 polynomials that represent as many conservation laws of the system (see Appendix C).

Airplanes vertical motion We consider the 6-th order longitudinal equations that capture the vertical motion (climbing, descending) of an airplane [32, Chapter 5]. The system is given by the equations below, where the variables have the following meaning: u = axial velocity, w = vertical velocity, x = range, z = altitude, q = pitch rate, θ = pitch angle; we also have two equations encoding $\cos\theta$ and $\sin\theta$. Applying the technique discussed in Remark 2, we also introduce the following auxiliary variables (parameters, hence 0 derivative): g = gravity acceleration, X/m, Z/m and M/I_{yy} whose meaning is described in [32] (see also [12, 13]); and u_0, w_0, x_0, z_0, q_0, standing for the generic initial values of the corresponding variables. Overall, the system's vector field F_2 consists of 17 polynomials over as many variables.

$$\begin{array}{rclrclrclrcl} \dot{u} &=& \frac{X}{m} - g\sin\theta - qw & \dot{z} &=& -u\sin\theta + w\cos\theta & \dot{w} &=& \frac{Z}{m} + g\cos\theta + qu & \dot{q} &=& \frac{M}{I_{yy}} \\ \dot{x} &=& u\cos\theta + w\sin\theta & \dot{\theta} &=& q & \dot{\cos\theta} &=& -q\sin\theta & \dot{\sin\theta} &=& q\cos\theta\,. \end{array}$$

In order to discover interesting polynomial invariants, we consider a complete template π of degree 2 over all the original system's plus two auxiliary variables, the latter representing the monomials qu and qw[10]. π is a linear combination of $n = 207$ monomials that uses as many parameters. We apply the approach underpinned by Theorem 2(b): we first pick up a precondition that requires $\theta = 0$ and assigns (generic) initial values to the remaining variables, $\psi_0 \triangleq \mathbf{V}(\{\theta, \sin\theta, \cos\theta - 1, u - u_0, w - w_0, x - x_0, z - z_0, q - q_0\})$. Note

[10]We could dispense with these auxiliary variables by considering a complete template of degree 3.

that $G = \{\theta, \sin\theta, ...\}$ is a set of generators for $\mathbf{I}(\psi)$, and in fact a Gröbner basis w.r.t. the lexicographic order (Proposition 1). We then run $\textsc{post}(\psi_0, \pi)$, which returns, after $m = 8$ iterations and about 26s, a pair (V, J). The vector space V corresponds to the following result template.

$$\begin{aligned}\pi' \;=\; \sum_{i=1}^{4} a_i \cdot p_i \;=\;& a_1 \cdot \left(\cos^2\theta + \sin^2\theta - 1\right) \quad + \quad a_2 \cdot \left(-\frac{1}{2}q^2 + \theta\frac{M}{I_{yy}} + \frac{1}{2}q_0^2\right) + \\ & a_3 \cdot \left(uq\cos\theta + wq\sin\theta - \frac{X}{m}\sin\theta + \frac{Z}{m}\cos\theta - x\frac{M}{I_{yy}} - -\frac{M}{I_{yy}}x_0 + u_0q_0 + \frac{Z}{m}\right) + \\ & a_4 \cdot \left(wq\cos\theta - uq\sin\theta - \theta g - \frac{X}{m}\cos\theta - \frac{Z}{m}\sin\theta - z\frac{M}{I_{yy}} - \frac{M}{I_{yy}}z_0 + w_0q_0 + \frac{X}{m}\right).\end{aligned}$$

Let $\phi \triangleq \mathbf{V}(\pi'[\mathbb{R}^n])$ be the variety defined by the result template π'. The invariant ideal J returned by the algorithm represents the weakest algebraic precondition $\chi \triangleq \mathbf{V}(J)$ such that $\chi \longrightarrow [F_2]\phi$: in other words, the largest algebraic precondition for which all instances of π' are polynomial invariants (Theorem 2(b)). Moreover, χ is also the weakest algebraic invariant included in ϕ (Theorem 2(a)). It is easily checked that $J = \langle\, \{p_1, p_2, p_3, p_4\} \,\rangle$. These findings generalize those in [12, 13]. In particular, one obtains the polynomial invariants of [12, 13] by letting $x_0 = z_0 = q_0 = 0$. By comparison, [12] reports that their method spent 1 hour to find a subset of all instances of π'. The method in [13] reportedly takes < 1s on this system, but again only finds a subset[11] of instances of π'. Moreover, it cannot infer the largest algebraic invariant implying the discovered laws, as we do.

Kepler laws We want to show how the $\textsc{post}$ algorithm automatically discovers the three Kepler's laws of planetary motion from Newton's law of gravitation. A nice and self-contained explanation of these laws can be found in [26]. Newton's laws are expressed below in a system of polar coordinates (r, θ) with the Sun at the origin. The meaning of the variables is as follows: r is the planet's distance from the origin; θ the angle from the positive horizontal semiaxis to the radius vector, measured counterclockwise; v_r and ω the planet's radial and angular velocity, respectively; $u = 1/r$ the distance reciprocal; for the purpose of expressing the invariants of interest, the system also includes equations for $\cos\theta$ and $\sin\theta$; moreover, we have constants (0 derivative variables) GM, a, e representing the product of the gravitational constant G and the Sun's mass M, the orbit's major semiaxis and its eccentricity, respectively (see below). A few more dummy constants are used to encode positivity conditions. Overall, the system's vector field F_3 consists of 15 polynomials over as many variables.

$$\begin{array}{cccc} \dot r = v_r \qquad \dot\theta = \omega & \dot v_r = -GMu^2 + r\omega^2 & \dot\omega = -2v_r\omega u & \dot u = -u^2 v_r \\ & \dot{\cos}\theta = -\omega\sin\theta & \dot{\sin}\theta = \omega\cos\theta\,. & \end{array} \tag{13}$$

Because Kepler's laws concern closed orbits[12], we first seek for a precondition ψ such that the planet's motion is an ellipse of major semiaxis a and eccentricity e. The equation of such an ellipse in polar coordinates, with one of the foci coinciding with the origin (Sun) and the horizontal axis passing through the ellipse's center, is $p_{\text{ell}} = 0$, where

$$p_{\text{ell}} \;\triangleq\; r(1 + e\cos\theta) - a(1 - e^2)\,. \tag{14}$$

We consider a suitable ψ_0 that implies a unitary circular orbit, which is an instance of p_{ell}, and apply Theorem 2(b): running $\textsc{post}(\psi_0, \pi_1)$ for a $\pi_1 = a_1 \cdot p_{\text{ell}}$, we discover, in about 43s, the largest (physically

[11] For instance, one should compare the polynomial $\psi_3 = q^2 - 2\frac{M\theta}{I_{yy}}$, which is part of the invariant cluster in [13], with the polynomial $p_2 = -\frac{1}{2}q^2 + \theta\frac{M}{I_{yy}} + \frac{1}{2}q_0^2$ in the second summand of π' above, which explicitly depends on the initial condition q_0.

[12] Note that non closed, hyperbolic or parabolic, trajectories are also possible.

meaningful) precondition ψ implying $p_{\mathrm{ell}} = 0$. In particular, for $\omega_{\mathrm{in}} \triangleq r^2\omega^2 - GM \cdot u \cdot (e+1)$, we have $\psi = \mathbf{V}(P)$ where

$$P \quad = \quad \{r - a(1-e), \theta, v_r, \omega_{\mathrm{in}}, u \cdot r - 1, \cos\theta - 1, \sin\theta\} \cup P_+ \,. \tag{15}$$

Here the set P_+ encodes positivity conditions on constants ($GM > 0, a > 0, 0 \le e < 1$) and is omitted for conciseness (further details on the computation of ψ and ψ_0 are given in Remark 3 below).

We next consider the complete polynomial template π_2 built out of monomials of degree ≤ 4 on the variables GM, a, e, r, u, dA, where $dA \triangleq \frac{1}{2}r^2\omega$ is an auxiliary variable, representing the areal velocity – that is, the first derivative of the area swept by the radius vector. We next run $\textsc{post}(\psi, \pi_2)$, which returns, after $m = 4$ iterations and about 58s, a pair (V', J'). The vector space V' corresponds to a result template $\pi_2' = a_1 \cdot (ur - 1) + a_2 \cdot (dA^2 - a \cdot GM(1-e^2)/4) + R$, where $R = \sum_{\ell=2}^{29} a_\ell \alpha_\ell$. The term $ur - 1$, that is $u = 1/r$, obtained by setting $a_1 = 1$ and the remaining parameters to 0, is another way of expressing Kepler's second law: indeed, it implies that $\mathcal{L}(dA) = -\omega r^2 u v_r + \omega r v_r = 0$, that is, that the areal velocity is constant. From Geometry, we know that the ellipse's area is $A = \pi a^2 \sqrt{1-e^2}$ (here $\pi \in \mathbb{R}$ denotes the mathematical constant). Since dA is a constant, the orbital period, expressed as a multiple of π, is $T \triangleq a^2\sqrt{1-e^2}/dA$. Therefore, the second term in π_2', obtained by setting $a_2 = 1$ and the remaining parameters to 0, can be read as saying that the square of the period, $T^2 = a^4(1-e^2)/dA^2$, is proportional to a^3, the cube of the semimajor axis: this is Kepler's third law. Any other summand of π_2' is either a multiple of $ur - 1$ or equivalent to the second term, hence it gives no further information.

Let $\phi' = \mathbf{V}(\pi_2[\lambda'])$. The invariant ideal J' returned by the algorithm represents the weakest algebraic precondition $\chi' \triangleq \mathbf{V}(J')$ such that $\chi' \longrightarrow [F_3]\phi'$: in other words, the largest algebraic precondition implying both the second and the third Kepler law (Theorem 2(b)). A Gröbner basis of the invariant ideal J' is $\{ur - 1, dA^2 - a \cdot GM(1-e^2)/4\}$, hence giving precisely the same information as π_2'.

Rather than "discovering" the laws, it is also possible to verify them directly using $\textsc{post}$, that is to check $\psi \longrightarrow [F_3]\phi_i$, with: $\phi_1 = \mathbf{V}(\{p_{\mathrm{ell}}\})$, $\phi_2 = \mathbf{V}(\{\mathcal{L}(dA)\})$ and $\phi_3 = \mathbf{V}(\{T^2 GM - 4a^3\})$. The running time for these checks is of about 45, 0.28 and 3s, respectively.

Remark 3 (on the computation of ψ_0 and ψ) Concerning the precondition ψ_0, we consider a simple unitary circular orbit, that is $p_{\mathrm{ell}} = 0$ with $GM = a = 1$ and $e = 0$. More precisely, considering as $t = 0$ to be a time when the planet is on the positive semiaxis, we let $\psi_0 = \mathbf{V}(P_0)$ with $P_0 = \{e,\ a-1,\ GM-1,\ r-1,\ \theta,\ \ v_r,\ \omega-1,\ \ u-1\}$ and use the template $\pi_1 = a_1 \cdot p_{\mathrm{ell}}$. We then run $\textsc{post}(\psi_0, \pi_1)$, which returns a pair (V, J), in $m = 8$ iterations and about 43s. By Theorem 2(b), $\chi \triangleq \mathbf{V}(J)$ is the largest algebraic precondition implying $p_{\mathrm{ell}} = 0$. A Gröbner basis of the invariant ideal J consists of 1197 polynomials. However, we want to restrict ourselves to physically meaningful initial conditions at time $t = 0$, and to closed orbits. Let J_0 denote the ideal generated by the polynomials encoding of the following conditions: $v_r = \theta = \sin\theta = 0$, $u \cdot r = \cos\theta = 1$, $r = a(1-e)$ (from $p_{\mathrm{ell}} = 0$), $dA = -r^2\omega/2$, $GM > 0$, $a > 0$ and on $0 \le e < 1$ (closed orbits). We then define $\psi \triangleq \mathbf{V}(J + J_0) = \chi \cap \mathbf{V}(J_0)$. A small set of polynomials representing ψ is obtained by computing a Gröbner basis G of $\sqrt{J + J_0}$, the complex radical of $J + J_0$. From G, via some simple manipulations, we compute the equivalent set P in (15); that is, we have $\psi = \mathbf{V}(P)$.

7 Application to continuous semialgebraic systems

We illustrate an application of the $\textsc{post}$ algorithm to the verification of continuous semialgebraic systems. The basic idea is that, once we have obtained via $\textsc{post}$ an algebraic invariant for the system at hand, we can check if this invariant, as a region of $\mathbb{R}^N$, intersects a specified *unsafe* region: if not, the system is safe. Exploiting systematically the idea of auxiliary variables (see Section 3, Remark 2), we will have the obtained invariant be dependent on a set of parameters $\mathbf{x}_0$, representing a generic initial condition for the given system.

A set $S \subseteq \mathbb{R}^N$ is *semialgebraic* if there are polynomials $g_1, ..., g_m \in \mathbb{R}[\mathbf{x}]$ such that $S = \{v \in \mathbb{R}^N : g_1(v) \geq 0, ..., g_m(v) \geq 0\}$, written $S = \mathbf{S}(\{g_1 \geq 0, ..., g_m \geq 0\})$[13]. A *continuous semialgebraic system* is a triple $SA = (F, X_0, X_U)$, composed by a a polynomial vector field F, an *initial region* $X_0 \subseteq \mathbb{R}^N$ and an *unsafe region* $X_U \subseteq \mathbb{R}^N$, both of which are semialgebraic. The system SA is *safe* if for each $v_0 \in X_0$ there is no $t \in D_{v_0}$ such that $\mathbf{x}(t; v_0) \in X_U$.

Let us now introduce some additional notation concerning auxiliary variables. Let $F = (f_1, ..., f_N)$ be a polynomial vector field, with $f_i \in \mathbb{R}[\mathbf{x}]$. Let $\mathbf{x}_0 = (x_{01}, ..., x_{0N})$ be a vector of N distinct variables, disjoint from $\mathbf{x}$: we define $\hat{\mathbf{x}} = (x_{01}, ..., x_{0N}, x_1, ..., x_N)$ and $\hat{F} = (0, ..., 0, f_1, ..., f_N)$. Note that $\hat{F}$ is a vector field $\mathbb{R}^{2N} \to \mathbb{R}^{2N}$, where the variables in $\mathbf{x}_0$ represent generic constants. For $v, w \in \mathbb{R}^N$, we will denote by (v, w) the element of $\mathbb{R}^{2N}$ obtained by concatenating v and w. Finally, we will denote by $g[\mathbf{x}_0/\mathbf{x}]$ the polynomial obtained from g by replacing each variable x_i with x_{i0}, for $i = 1, ..., N$.

The following result gives a sufficient algebraic condition for safety of a continuous semialgebraic system. Its intuitive interpretation is as follows. In $\mathbb{R}^{2N}$, let ψ be a precondition encoding that $\mathbf{x}_0$ is the initial condition for $\mathbf{x}$, and let J be an invariant ideal representing a corresponding postcondition, explicitly depending on $\mathbf{x}_0$. Hence, for any concrete instance of the initial conditions $\mathbf{x}_0$, we obtain from J a corresponding concrete postcondition. If there is no solution of the set of (in)equations representing the intersection of the initial region, of the postconditions and of the unsafe region, then the system is safe.

Theorem 3 (safety of semialgebraic systems) *Let $SA = (F, X_0, X_U)$ be a semialgebraic system, where $X_0 = \mathbf{S}(\{g_1 \geq 0, ..., g_m \geq 0\})$ and $X_U = \mathbf{S}(\{h_1 \geq 0, ..., h_n \geq 0\})$ $(g_i, h_j \in \mathbb{R}[\mathbf{x}])$. Let $\psi = \mathbf{V}(\{x_i - x_{i0} : i = 1, ..., N\}) \subseteq \mathbb{R}^{2N}$ and let $J = \langle\, \{q_1, ..., q_k\}\, \rangle \subseteq \mathbb{R}[\hat{\mathbf{x}}]$ be an invariant ideal for $\hat{F}$ such that $\mathbf{V}(J) \supseteq \psi$. Assume the following polynomial system in the variables $\hat{\mathbf{x}}$*

$$g_1[\mathbf{x}_0/\mathbf{x}] \geq 0, ..., g_m[\mathbf{x}_0/\mathbf{x}] \geq 0,\ h_1 \geq 0, ..., h_n \geq 0,\ q_1 = 0, ..., q_k = 0 \tag{16}$$

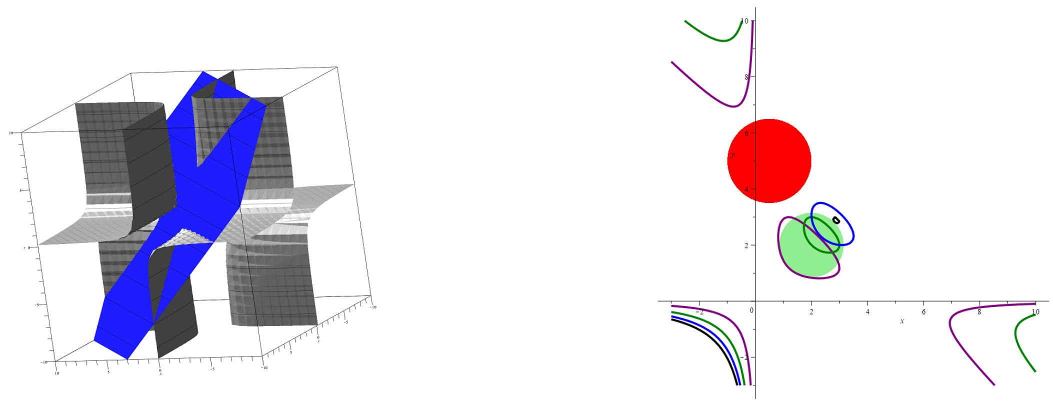
has no solution in $\mathbb{R}^{2N}$. Then SA is safe.

Proof By contradiction, assume there are $w_0 \in X_0$ and $w_1 = \mathbf{x}(t_1; w_0) \in X_U$, for some $t_1 \in D_{w_0}$. We will show that $(w_0, w_1) \in \mathbb{R}^{2N}$ is a solution of (16), thus arriving at a contradiction. Indeed, by definition $g_i[\mathbf{x}_0/\mathbf{x}](w_0, w_1) = g_i(w_0) \geq 0$ and $h_j(w_0, w_1) = h_j(w_1) \geq 0$ for each $i = 1, ..., m$ and $j = 1, ..., n$. Consider now the trajectory of $\hat{F}$ originating from (w_0, w_0), that is $\hat{\mathbf{x}}(t; (w_0, w_0))$: note that, by definition of $\hat{F}$, $\hat{\mathbf{x}}(t; (w_0, w_0)) = (w_0\,, \mathbf{x}(t; w_0))$ for each $t \in D_{w_0}$. Now, since $\mathbf{V}(J) \supseteq \psi$, we have $(w_0, w_0) \in \mathbf{V}(J)$, hence, by $\hat{F}$-invariance of J, $\hat{\mathbf{x}}(t; (w_0, w_0)) \in \mathbf{V}(J)$ for each $t \in D_{(w_0, w_0)}$ (Lemma 5). In particular, considering $t = t_1$, we have $\hat{\mathbf{x}}(t_1; (w_0, w_0)) = (w_0\,, \mathbf{x}(t_1; w_0)) = (w_0, w_1) \in \mathbf{V}(J)$. But this means $q_i(w_0, w_1) = 0$ for $i = 1, ..., k$. In conclusion, (w_0, w_1) is a solution of (16). □

There are two aspects of the previous result that are worthwhile commenting. First, checking that an algebraic system of (in)equalities like (16) is solvable is decidable, although NP-hard. One well-known and effective technique to establish insolvability is to rely on Positivstellensatz [33] and Sum-of-Squares programming: this also provides easy to verify *certificates* of insolvability. For the sake of completeness, we outline this technique in Appendix D.

Second, the procedure resulting from the theorem is of course incomplete, and its precision depends on how rich the ideal J is. An invariant ideal J satisfying the hypotheses of the theorem can be obtained by running $\textsc{post}_{\hat{F}}(\psi, \pi)$ with ψ as specified in the statement of the theorem, and any template $\pi \in \mathbb{L}(\mathbf{a})[\hat{\mathbf{x}}]$.

[13]Note that an equality $g_i = 0$ can be coded up as a pair of inequalities $g_i \geq 0$ and $-g_i \geq 0$. Similarly, a strict inequality $g_i > 0$ can be coded up using an auxiliary slack variable z as $g_i \cdot z^2 - 1 \geq 0$.



(a) (b)

Figure 1: With reference to the 3D Lotka-Volterra system in Example 6: (a) surfaces in $\mathbb{R}^3$ induced by the polynomials q_1 (gray) and q_2 (blue) in J, when instantiating (x_0, y_0, z_0) to $(1, 1, 1)$; the corresponding algebraic invariant coincides with the intersection of the two surfaces; (b) projection onto the (x, y)-plane of X_0 (green), of X_U (red) and of the four algebraic invariants obtained from J by instantiating (x_0, y_0, z_0) to four points in X_0: $(2 - c, 2 - c, 3)$, $(2, 2, 3)$, $(2, 3.15, 3)$ and $(2 + c, 2 + c, 3)$, where $c = 1.15/\sqrt{2}$.

Indeed, if $(J, V) = \textsc{post}_{\hat{F}}(\psi, \pi)$ (for some V), by Theorem 1(b) J is a $\hat{F}$-invariant ideal such that $\mathbf{V}(J) \supseteq \psi$. The last point follows because $J \subseteq I_\psi$ implies that $q(w, w) = 0$ for each $q \in J$ and $(w, w) \in \psi$. Note that this is a case where a basis for the real radical $\mathbf{I}(\psi)$ is trivial (Proposition 1), hence relative completeness holds. Therefore, by tuning the template π, one can in principle hope to obtain a J which is as precise as possible. The following example[14] illustrates this theorem.

Example 6 (3D Lotka-Volterra) Consider the 3D Lotka-Volterra system defined by $\mathbf{x} = (x, y, z)$ and the vector field $F = (xy - xz, yz - yx, zx - zy)$; see e.g. [30, 13]. Consider the semialgebraic system $SA = (F, X_0, X_U)$, where $X_0 = \mathbf{S}(\{z = 3, (x - 2)^2 + (y - 2)^2 \le 1.15^2\})$ (a disk) and $X_U = \mathbf{S}(\{(x - 1/2)^2 + (y - 5)^2 \le 1.5^2\})$ (an infinite cylinder). We wish to prove that SA is safe.

Consider the extension of F, $\hat{F}$, over the variables $\hat{\mathbf{x}} = (x_0, y_0, z_0, x, y, z)$; let $\pi \in \mathbb{L}(\mathbf{a})[\hat{\mathbf{x}}]$ be a complete template of degree 3. Running $\textsc{post}_{\hat{F}}(\psi, \pi)$ with $\psi = \mathbf{V}(\{x - x_0, y - y_0, z - z_0\})$, we get as a result (after about 40s) a pair (V, J) where $J = \langle\ \{q_1, q_2\}\ \rangle$ and

$$\begin{aligned} q_1 &= xzy - xz_0y_0 - zz_0y_0 - yz_0y_0 + z_0^2y_0 + z_0y_0^2 \\ q_2 &= x_0 + y_0 + z_0 - x - y - z\,. \end{aligned}$$

By the above discussion, J is a $\hat{F}$-invariant ideal and $\mathbf{V}(J) \supseteq \psi$. For any instantiation of x_0, y_0, z_0 with real values, J represents a 1-dimensional variety in $\mathbb{R}^3$, that is a curve, obtained as the intersection of two surfaces. See Fig. 1(a). Any trajectory starting in such a variety will remain in it.

Fig. 1(b) shows the projection onto the (x, y)-plane of X_0, of X_U and of four curves induced by J, when instantiating (x_0, y_0, z_0) with four distinct points in X_0. None of those curves intersects the unsafe region, suggesting that the system might be safe. This can in fact be proven algebraically relying on (16), which for the present case is equivalent to the following (we have eliminated the variable z_0 exploiting

[14] SageMath/Python scripts for the examples in this section available at `https://github.com/micheleatunifi/postconditions/blob/master/Post.py`.

the equation $z_0 = 3$ for X_0):

$$\begin{aligned}
-(x_0-2)^2-(y_0-2)^2+1.15^2 &\geq 0\\
-(x-1/2)^2-(y-5)^2+1.5^2 &\geq 0\\
xzy-3xy_0-3zy_0-3yy_0+9y_0+3y_0^2 &= 0\\
x_0+y_0+3-x-y-z &= 0\,.
\end{aligned}$$

This algebraic system is proven to have no solutions: we give the details, including a certificate of insolvability, in Appendix D. Therefore, by virtue of Theorem 3, SA is safe.

Theorem 3 admits the following slight generalization, that allows for a more flexible use of auxiliary variables. We report a proof in Appendix D.

Theorem 4 *Let $SA = (F, X_0, X_U)$ be a semialgebraic system, and $\hat{\mathbf{x}}$ and $\hat{F}$ be the extended variable vector and vector field, like in Theorem 3. For $r_1, ..., r_N \in \mathbb{R}[\mathbf{x}_0]$, let $\psi = \mathbf{V}(\{x_i - r_i : i = 1, ..., N\})$. Finally let $J = \langle \{q_1, ..., q_k\} \rangle \subseteq \mathbb{R}[\hat{\mathbf{x}}]$ be an invariant ideal for $\hat{F}$ such that $\mathbf{V}(J) \supseteq \psi$. Assume the following conditions hold true:*

(a) $(X_0 \times X_0) \cap Id \subseteq \psi$ (Id = identity relation over $\mathbb{R}^{2N}$);

(b) The polynomial system in the variables $\hat{\mathbf{x}}$ in (16) has no solution in $\mathbb{R}^{2N}$.

Then SA is safe.

Example 7 (coupled spring-mass system) A system consists of two identical springs of elastic constant k and length L and two identical bodies of mass m, connected in cascade: wall, spring 1, mass 1, spring 2, mass 2. This system is governed by the following equations, where x_1, x_2 and v_1, v_2 denote, respectively, the bodies' positions and velocities on the horizontal axis with the origin fixed at the wall:

$$\begin{aligned}
\dot{x}_1 &= v_1\\
\dot{v}_1 &= (k/m)(x_2-2x_1)\\
\dot{x}_2 &= v_2\\
\dot{v}_2 &= -(k/m)(x_2-x_1-L)\,.
\end{aligned}$$

Considering k/m and L as 0-derivative variables, that is constants, we let $\mathbf{x} = (k/m, L, x_1, v_2, x_2, v_2)$ and $F = (0, 0, v_1, k/m(x_2-2x_1), v_2, -k/m(x_2-x_1-L))$. Consider the system $SA = (F, X_0, X_U)$ with initial set $X_0 = \mathbf{S}(\{k/m = 1, L = 1, (x_1-1/2)^2 + (x_2-3/2)^2 \leq 1/4, v_1 = 0, v_2 = 0\})$ and unsafe set $X_U = \mathbf{S}(\{x_2 - x_1 \geq 2.17\})$. That is, we fix the value of both constants to 1 and the initial velocities to 0, and let the initial positions (x_1, x_2) of the two masses vary in a disk of radius 1/2 centered at $(1/2, 3/2)$. We then ask if the distance of the first mass from the second ever reaches or exceeds the value 2.17. Fig. 2(a), displaying the plots of $x_2(t; v_0) - x_1(t; v_0)$ for 100 random initial conditions $v_0 \in X_0$, suggests that the system might indeed be safe. We now prove this fact. Note that, despite the linearity of the system, nonlinear invariants will be essential to prove safety.

As dictated by Theorem 4, we consider an extended vector field $\hat{F}$ over the variables[15] $\hat{\mathbf{x}}$, then take a complete template π of total degree 3 over $\hat{\mathbf{x}}$ and $\psi = \mathbf{V}(\{x_1 - x_{10}, x_2 - x_{20}, v_1, v_2\})$. We obtain $(V, J) = \text{POST}_{\hat{F}}(\psi, \pi)$, which takes about 60s, for some V and $J = \langle \{q_1, q_2\} \rangle$, where

$$\begin{aligned}
q_1 = {} & 2/3(k/m)Lx_{10} + (k/m)x_{10}^2 - 4/3(k/m)x_{10}x_{20} + 1/3(k/m)x_{20}^2 - 2/3(k/m)Lx_1 - (k/m)x_1^2 + 4/3(k/m)x_1x_2 -\\
& 1/3(k/m)x_2^2 - 1/3v_1^2 + 2/3v_1v_2\\
q_2 = {} & 2/3(k/m)Lx_{10} + (k/m)Lx_{20} - 1/3(k/m)x_{10}x_{20} - 1/6(k/m)x_{20}^2 - 2/3(k/m)Lx_1 - (k/m)Lx_2 + 1/3(k/m)x_1x_2 +\\
& 1/6(k/m)x_2^2 + 1/6v_1^2 + 2/3v_1v_2 + 1/2v_2^2\,.
\end{aligned}$$

[15] In practice, it is superfluous to introduce auxiliary copies of the constants $k/m, L$. The same is true for v_1, v_2: as the their initial values is fixed by ψ, their auxiliary copies would be anyway eliminated from the final system (16).

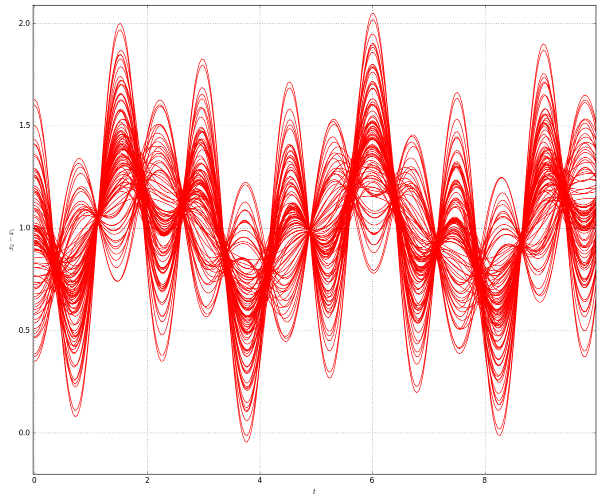


Figure 2: With reference to the coupled spring-mass system in Example 7, plots of 100 trajectories $x_2(t) - x_1(t)$ starting from random points in X_0.

One can check that, once fixed $k/m = L = 1$, for any instantiation of the parameters x_{10} and x_{20} in J, the dimension of the resulting variety is ≤ 2, implying it represents a good over-approximation the resulting system trajectory. Once k/m and L have been eliminated, the system (16) becomes

$$\begin{array}{rcl} -(x_{10}-1/2)^2-(x_{20}-3/2)^2+1/4 & \geq & 0 \\ x_2-x_1-2.17 & \geq & 0 \\ x_{10}^2-4/3x_{10}x_{20}+1/3x_{20}^2-x_1^2+4/3x_1x_2-1/3x_2^2-1/3v_1^2+2/3v_1v_2+2/3x_{10}-2/3x_1 & = & 0 \\ -1/3x_{10}x_{20}-1/6x_{20}^2+1/3x_1x_2+1/6x_2^2+1/6v_1^2+2/3v_1v_2+1/2v_2^2+2/3x_{10}+x_{20}-2/3x_1-x_2 & = & 0\,. \end{array}$$

This algebraic system is proven to have no solutions: we give some details in Appendix D. Therefore, by virtue of Theorem 4, SA is safe.

8 Conclusion, further and related work

We have provided complete algorithms to compute relativized strongest postconditions for systems of polynomial ODEs. These algorithms can be used to check safety assertions, to discover complete sets of polynomial invariants that fit a given template, and to compute largest algebraic varieties of initial conditions making given polynomial invariants true (weakest preconditions). Effectiveness of the algorithms has been demonstrated on nontrivial systems, including semialgebraic ones.

Our previous work [6] deals with initial values problems, where the precondition ψ always consists of a singleton. The method introduced there has its roots in a line of research concerning weighted automata, bisimulation and Markov chains [4, 3, 5]. Also somewhat related to the present paper is [8], where we apply the notion of invariant ideal to the construction of linear abstractions of continuous systems.

The study of the safety of hybrid systems can be shown to reduce constructively to the problem of generating invariants for their differential equations [23]. Many authors have therefore focused on the effective generation of invariants of a special type. For example, Tiwari and Khanna consider invariants generation based on *syzygies* and Gröbner basis [35]. Sankaranarayanan [30] characterizes greatest invariants in terms of a descending chains of ideals. This iteration does not always converge, thus a relaxation in terms of bounded-degree *pseudoideals* is considered: the resulting algorithm always converges, because pseudoideals form basically a descending chain of finite-dimensional vector spaces, and returns an invariant ideal, although with no guarantee of maximality [30, Th.4.1]. By contrast, the convergence of our algorithm Post is essentially based on the stabilization of ascending chains of ideals, with completeness guarantees. Matringe et al. encode invariants constraints using symbolic matrices [27].

Our work is closely related to Ghorbal and Platzer's [12], which gives a complete characterization of what it means for an algebraic set to be invariant for a polynomial ODE. It is interesting to contrast the completeness statements in [12] and of that in the present paper with one another. [12] presents a method that, given a polynomial template and an integer $N \geq 1$, determines the largest subspace of template instantiations under which a length N chain of Lie derivatives forms an invariant. Any invariant can be reduced to this form, for suitably large N. In contrast, our Theorem 1, given a template *and* an algebraic variety ψ, determines the largest subspace of the template instantiations that are polynomial invariants for trajectories starting from ψ; moreover, it determines, via J, the largest invariant variety that includes ψ. Neither of these two statements is stronger or more general than the other. From our point of view, taking the initial set explicitly into account, as we do, has some advantages. First, invariants J returned by our method can be made explicitly dependent on initial conditions, via auxiliary variables, such as x_0, z_0, w_0, u_0, q_0 in the longitudinal airplane motion. As such, these invariants can be used *directly* within semialgebraic verification methods based on Positivstellensatz: as seen in Section 7, this amounts to proving the unsatisfiability of a set of polynomial (in)equations, also involving the auxiliary variables, corresponding to the initial set, to the polynomial invariants, and to the unsafe set. Second, knowing the precondition ψ, we are in effect confining ourselves to a subset of the algebraic invariants, those that involve points in ψ: this might explain the observed gain in efficiency – practically speaking, as the worst-case complexity is left unchanged. This gain is reflected in the execution times of [12] and of our algorithm, for the examples reported in Section 6. As a more general remark, we note that the computational ingredients of [12], such as minimization of the rank of a symbolic matrix (also employed in [27]), are quite different from ours. In the future, we would like to experimentally compare these two approaches on a more systematic basis than what we have done in the present paper.

The recent work of Kong et al. [13] considers generation of invariant clusters, again based on templates. Nonlinear constraints on template parameters are resolved via symbolic computation; safety for semialgebraic systems is reduced, via Positivstellensatz, to Sum-of-Squares (SOS) programming. The resulting approach also works for semialgebraic systems. In terms of generality and effectiveness, this approach appears to considerably improve previous techniques. Kong et al.'s approach has strong similarities with our method for semialgebraic systems: in our case, clusters are generated via the introduction of auxiliary variables $\mathbf{x}_0$. Differently from our approach, though, [13] does not offer completeness guarantees in our sense: in particular, the approach only works with ideal chains that stabilize after one step, a.k.a. Darboux polynomials. On the other hand, compared to theirs, our approach appears to be slower. It would be interesting to investigate if the more general invariants J returned by our algorithm could be fruitfully employed in the approach of [13].

Ideas from Algebraic Geometry have been fruitfully applied also in Program Analysis. Relevant to our work is Müller-Olm and Seidl's [18], where an algorithm to compute all polynomial invariants up to a given degree of an imperative program is provided. Similarly to what we do, they reduce the core problem to a linear algebraic one. However, being the setting in [18] discrete rather than continuous, the techniques employed there are otherwise quite different, mainly because: (a) the construction of the ideal chain is driven by the program's operational semantics, rather than by Lie derivatives; (b) only the polynomial invariants satisfied by *all* initial program states are considered, which in a continuous setting would mostly lead to the trivial strongest postcondition. A perhaps more crucial difference is that, when computing invariants, [18] regards templates essentially as polynomials in $\mathbb{R}[\mathbf{x}, \mathbf{a}]$, rather than explicitly factoring out the parameters $\mathbf{a}$. This can make the involved computations less efficient, as known algorithms for computing Gröbner bases have a complexity which is exponential in the number of variables.

Most of the material in this paper has been extended and revised from the conference paper [7]. With respect to [7], here we include the following additional material: proofs, the discussion on the expressive power of auxiliary variables in Section 4, the algorithmic presentation and the discussion on radical ideals in Section 5, the examples about collision avoidance and Kepler laws in Section 6, the extension to semialgebraic continuous systems in Section 7, and an extended and revised discussion of related works in the present section.

Acknowledgment The author has benefited from stimulating discussions with Khalil Ghorbal.

References

[1] V.I. Arnold. *Ordinary Differential Equations*. The MIT Press, ISBN 0-262-51018-9, 1978.

[2] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. BioNet-Gen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17): 3289-3291, 2004.

[3] F. Bonchi, M.M. Bonsangue, M. Boreale, J.J.M.M. Rutten, and A. Silva. A coalgebraic perspective on linear weighted automata. *Inf. Comput.* 211: 77-105, 2012.

[4] M. Boreale. Weighted Bisimulation in Linear Algebraic Form. *CONCUR 2009*, LNCS 5710: 163-177, Springer, 2009.

[5] M. Boreale. Analysis of Probabilistic Systems via Generating Functions and Padé Approximation. *ICALP 2015* (2) 2015: 82-94, LNCS 9135, Springer, 2015. Extended version available as DiSIA working paper 2016/10, `http://local.disia.unifi.it/wp_disia/2016/wp_disia_2016_10.pdf`.

[6] M. Boreale. Algebra, coalgebra, and minimization in polynomial differential equations. In *Proc. of FoSSACS 2017*, LNCS 10203:71-87, Springer, 2017. Full version in *Logical Methods in Computer Science* 15(1), 2019.

[7] M. Boreale. Complete algorithms for algebraic strongest postconditions and weakest preconditions in polynomial ODEs. *SOFSEM 2018: Theory and Practice of Computer Science - 44th International Conference on Current Trends in Theory and Practice of Computer Science*, LNCS 10706:442-455, Springer, 2018.

[8] M. Boreale. Algorithms for exact and approximate linear abstractions of polynomial continuous systems. *HSCC 2018*: 207-216, ACM, 2018.

[9] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics, Springer, 2007.

[10] W. Decker, G.-M. Greuel, G. Pfister, H. Schönemann. SINGULAR 4-1-2 — A computer algebra system for polynomial computations. `http://www.singular.uni-kl.de`, 2019.

[11] M.S. El Din, Zhi-H. Yang, L. Zhi. On the Complexity of Computing Real Radicals of Polynomial Systems. *ISSAC 2018*: 351-358, ACM, 2018.

[12] K. Ghorbal, A. Platzer. Characterizing Algebraic Invariants by Differential Radical Invariants. *TACAS 2014*, LNCS 8413: 279-294, 2014. Extended version available from `http://reports-archive.adm.cs.cmu.edu/anon/2013/CMU-CS-13-129.pdf`.

[13] H. Kong, S. Bogomolov, Ch. Schilling, Yu Jiang, Th.A. Henzinger. Safety Verification of Nonlinear Hybrid Systems Based on Invariant Clusters. *HSCC 2017*:163-172, ACM, 2017.

[14] T. Krick T, A. Logar. An algorithm for the computation of the radical of an ideal in the ring of polynomials. *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. AAECC 1991*, LNCS 539:195-205, Springer, 1991.

[15] A.S. Laplagne. An algorithm for the computation of the radical of an ideal. *Proceedings of the 2006 international symposium on Symbolic and algebraic computation. ISSAC 2006*, pp 191-195, ACM, 2006.

[16] J-B. Lasserre, M. Laurent, B. Mourrain, P. Rostalski, Ph. Trébuchet. Moment matrices, border bases and real radical computation. *Journal of Symbolic Computation*, 51:63-85, 2013.

[17] J. Liu, N. Zhan, H. Zhao. Computing Semi-algebraic Invariants for Polynomial Dynamical Systems. *EMSOFT*: 97-106, ACM, 2011.

[18] M. Müller-Olm and H. Seidl. Computing polynomial program invariants. *Information Processing Letters* 91(5), 233-244, 2004.

[19] R. Neuhaus. Computation of real radicals of polynomial ideals - II. *Journal of Pure and Applied Algebra* 124 (1-3): 261-280, 1998.

[20] D. Novikov and S. Yakovenko. Trajectories of polynomial vector fields and ascending chains of polynomial ideals. *Annales de l'institut Fourier* 49(2): 563-609, 1999.

[21] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming* 96(2): 293-320, 2003.

[22] A. Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reasoning* 41(2), 143-189, 2008.

[23] A. Platzer. Logics of dynamical systems. *LICS 2012*: 13-24, IEEE, 2012.

[24] A. Platzer. The structure of differential invariants and differential cut elimination. *Logical Methods in Computer Science* 8(4): 1-38, 2012.

[25] S. Prajna, A. Papachristodoulou, P. Seiler, P. Parrilo. SOSTOOLS and its control applications. *Positive polynomials in control*, pp. 580- 580, 2005.

[26] G.R. Putland. A self-contained derivation of Kepler's laws from Newton's laws. `https://www.grputland.com/2013/12/self-contained-derivation-of-keplers-laws-from-newtons-laws.html`, 2013.

[27] R. Rebiha, A. V. Moura, and N. Matringe. Generating invariants for non-linear hybrid systems. *Theoretical Computer Science*, 594:180-200, 2015.

[28] SageMath, free open-source mathematics software, `http://www.sagemath.org/`.

[29] S. Sankaranarayanan, H. Sipma, and Z. Manna. Non-linear loop invariant generation using Gröbner bases. *POPL 2004*: 318-329, ACM, 2004.

[30] S. Sankaranarayanan. Automatic invariant generation for hybrid systems using ideal fixed points. *HSCC 2010*: 221-230, ACM, 2010.

[31] A. Sogokon, K. Ghorbal, P.B. Jackson and A. Platzer. A method for invariant generation for polynomial continuous systems. In *Verification, Model Checking, and Abstract Interpretation - 17th International Conference*, VMCAI 2016, LNCS 9583:268-288. Springer, 2016.

[32] R.F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.

[33] G. Stengle. A Nullstellensatz and a Positivstellensatz in Semialgebraic Geometry. *Mathematische Annalen.* 207(2): 87-97, 1974.

[34] A. Tiwari. Approximate reachability for linear systems. *HSCC 2003*: 514-525, ACM, 2003.

[35] A. Tiwari, G. Khanna. Nonlinear systems: Approximating reach sets. *HSCC 2004*:600-614, ACM, 2004.

[36] M. Tribastone, S. Gilmore, and J. Hillston. Scalable differential analysis of process algebra models. *IEEE Trans. Software Eng.*, 38(1):205-219, 2012.

A A simple algorithm for weakest preconditions

Fix a vector field F. Let $\phi = \mathbf{V}(P)$ be a user specified postcondition, with $P \subseteq \mathbb{R}[\mathbf{x}]$ a finite set of polynomials. We define inductively the sets P_j, $j \geq 0$, as follows: $P_0 \triangleq P$ and $P_{j+1} = \mathcal{L}(P_j)$. For $j \geq 0$, we let

$$I_j \triangleq \langle \ \cup_{i=0}^{j} P_i \ \rangle . \tag{17}$$

Let m the least integer such that $I_m = I_{m+1}$, which must exist as $I_0 \subseteq I_1 \subseteq \cdots$ forms an infinite ascending chains of ideals that must eventually stabilize. We let $\textsc{pre}(\phi) \triangleq I_m$. Note that the termination condition reduces to checking equality between two ideals, which can be effectively done (Section 2).

Theorem A.1 (correctness and completeness of $\textsc{pre}$) *Let ϕ be an algebraic variety and $I = \textsc{pre}(\phi)$. Then $\mathbf{V}(I) = \psi_\phi$.*

Proof Let $\chi \triangleq \mathbf{V}(I)$. It is easy to check that $I = \langle \ \{p^{(j)} \ : \ j \geq 0 \text{ and } p \in P\} \ \rangle$ and that I is an invariant ideal. By Lemma 5 then χ is an algebraic invariant of F, that is $\chi \longrightarrow [F]\chi$. Moreover, as $P \subseteq I$, $\phi \supseteq \chi$, hence $\chi \longrightarrow [F]\,\phi$. This shows that χ is a valid precondition of ϕ. We now show that it is actually the largest. Consider any ψ such that $\psi \longrightarrow [F]\,\phi$ and any $v_0 \in \psi$. This means that, for each $p \in I$, p is a polynomial invariant for v_0. That is (Lemma 2), for each $p \in I$ and $j \geq 0$, $p^{(j)}(v_0) = 0$. Therefore, $v_0 \in \mathbf{V}\Big(\{p^{(j)} \ : \ j \geq 0 \text{ and } p \in P\}\Big) = \mathbf{V}(I) = \chi$. □

B Proofs of Section 4

Proof of Lemma 3 Recall from [9] that the *multidegree* of a monomial $\alpha \in \mathbb{R}[\mathbf{a}, \mathbf{x}]$ is the tuple of the exponents of the parameters a_i ($1 \leq i \leq n$) and variables x_j ($1 \leq j \leq N$) as they occur in α: $\text{multideg}(\alpha) = (k_1, ..., k_n, k_{n+1}, ..., k_{n+N})$. Once the multidegrees are totally ordered, one lets multideg(π) be the maximum multidegree of the monomials occurring in π. Note, by our choice of a lexicographic order where $a_i > x_j$ for any i, j, we have for example, $\text{multideg}(a_i^2 x_j) > \text{multideg}(a_i x_j^k)$, whatever $k \geq 0$.

Let π be a template and $r = \pi \bmod G$. We first prove that r is a template as well, that is, parameters a_i can occur only linearly in r. By the properties of multivariate division [9, Ch.2,§3,Theorem 3], there is a $q = \sum_\ell h_\ell g_\ell$, with $h_\ell \in \mathbb{R}[\mathbf{a}, \mathbf{x}]$ and $g_\ell \in G$, such that

$$\pi \quad = \quad q + r . \tag{18}$$

Moreover, again by the same result: (a) multideg$(\pi) \geq$ multideg(q); (b) r is a linear combination of monomials, none of which is divisible by the leading term of any polynomial in G. Assume by contradiction there is in r a summand $\mu\alpha$ ($0 \neq \mu \in \mathbb{R}$), where a parameter a_i occurs in the monomial α with a degree > 1. By the linearity of π in $\mathbf{a}$ and (18), we deduce that $-\mu\alpha$ must be a summand of q (seen as a linear combination of monomials), so that the two terms can cancel each other. We deduce that multideg$(q) \geq$ multideg(α). Hence, by (a) above and transitivity, multideg$(\pi) \geq$ multideg(α). But this is impossible: indeed, by the chosen lexicographic order, we must have multideg$(\pi) <$ multideg(α), because π is linear in all parameters in $\mathbf{a}$, whereas in α the degree of a_i is ≥ 2.

Now, consider any $\lambda \in \mathbb{R}^n$. By (18) we have $\pi[\lambda] = q[\lambda] + r[\lambda]$. Clearly $q[\lambda] \in \langle \ G \ \rangle$, seen as a subset of $\mathbb{R}[\mathbf{x}]$; moreover, (b) above implies that none of the monomials in $r[\lambda]$ is divisible by the leading term of any polynomial in G. Since G is a Gröbner basis in $\mathbb{R}[\mathbf{x}]$, these two properties say that $r[\lambda]$ is the (unique) remainder of the division of $\pi[\lambda]$ by G: see e.g. [9, Ch.2,§6,Proposition 1]. In other words, $\pi[\lambda] \bmod G = r[\lambda]$. □

Proof of Lemma 4 We proceed by induction on j. The base case $j = 1$ follows from the definition of m. Assuming by induction hypothesis that $V_m = \cdots = V_{m+j}$ and that $J_m = \cdots = J_{m+j}$, we prove now that $V_m = V_{m+j+1}$ and that $J_m = J_{m+j+1}$. The key to the proof is the following fact

$$\pi^{(m+j+1)}[\lambda] \quad \in \quad J_m \quad \text{for each } \lambda \in V_m\,. \tag{19}$$

From this fact the thesis will follow, indeed:

1. $V_m = V_{m+j+1}$. To see this, observe that for each $\lambda \in V_{m+j} = V_m$ (the equality here follows from the induction hypothesis), it follows from (19) that $\pi^{(m+j+1)}[\lambda]$ can be written as a finite sum of the form $\sum_l h_l \cdot \pi^{(j_l)}[u_l]$, with $0 \le j_l \le m$ and $u_l \in V_m$. For each $0 \le j_l \le m$, $\pi^{(j_l)}[u_l] \bmod G = 0$ by assumption, from which it easily follows that also $\pi^{(m+j+1)}[\lambda] \bmod G = \left(\sum_l h_l \cdot \pi^{(j_l)}[u_l]\right) \bmod G = 0$. This shows that $\lambda \in V_{m+j+1}$ and proves that $V_{m+j+1} \supseteq V_{m+j} = V_m$. The reverse inclusion is obvious;

2. $J_m = J_{m+j+1}$. As a consequence of $V_{m+j+1} = V_{m+j} (= V_m)$ (the previous point), we can write

$$\begin{aligned} J_{m+j+1} &= \langle\ \cup_{i=1}^{m+j} \pi^{(i)}[V_{m+j}] \cup \pi^{(m+j+1)}[V_{m+j}]\ \rangle \\ &= \langle\ J_{m+j} \cup \pi^{(m+j+1)}[V_{m+j}]\ \rangle \\ &= \langle\ J_m \cup \pi^{(m+j+1)}[V_m]\ \rangle \end{aligned}$$

   where the last step follows by induction hypothesis. From (19), we have that $\pi^{(m+j+1)}[V_m] \subseteq J_m$, which implies the thesis for this case, as $\langle\ J_m\ \rangle = J_m$.

We prove now (19). Fix any $\lambda \in V_m$. First, note that $\pi^{(m+j+1)}[\lambda] = \mathcal{L}(\pi^{(m+j)}[\lambda])$ (here we are using (6)). As by induction hypothesis $\pi^{(m+j)}[V_m] = \pi^{(m+j)}[V_{m+j}] \subseteq J_{m+j} = J_m$, we have that $\pi^{(m+j)}[\lambda]$ can be written as a finite sum $\sum_l h_l \cdot \pi^{(j_l)}[u_l]$, with $0 \le j_l \le m$ and $u_l \in V_m$. Applying the rules of Lie derivatives (2), (3), we find that $\pi^{(m+j+1)}[\lambda] = \mathcal{L}(\pi^{(m+j)}[\lambda])$ equals

$$\sum_l \left(h_l \cdot \pi^{(j_l+1)}[u_l] + \mathcal{L}(h_l) \cdot \pi^{(j_l)}[u_l]\right)\,.$$

Now, for each u_l, $u_l \in V_m = V_{m+1}$, each term $\pi^{(j_l+1)}[u_l]$, with $0 \le j_l + 1 \le m + 1$, is by definition in $J_{m+1} = J_m$. This proves that $\pi^{(m+j+1)}[V] \in J_m$, as required. □

Proof of Lemma 5 First assume that χ is an algebraic invariant. Take $J = \mathbf{I}(\chi)$. This is by definition an ideal. We show that J is invariant. Indeed, take any $v_0 \in \chi$ and $p \in J$: by hypothesis, $\mathbf{x}(t; v_0) \in \chi$, hence $p(t; v_0) = 0$, for each $t \in D_{v_0}$, that is p is a polynomial invariant for v_0. But, by Lemma 2, this is equivalent to $p^{(j)}(v_0) = 0$ for each $j \ge 0$, in particular, $p^{(1)}(v_0) = 0$. Since $v_0 \in \chi$ is arbitrary, $p^{(1)} \in J$. Since $p \in J$ is arbitrary, we have that J is an invariant ideal.

Conversely, assume that $\chi = \mathbf{V}(J)$ for J an invariant ideal. Of course χ is algebraic. We show that χ is invariant, that is that $\chi \longrightarrow [F]\chi$. Indeed, take any $v_0 \in \chi$ and $p \in J$: by hypothesis, $p^{(j)} \in J$ for each $j \ge 0$, hence $p^{(j)}(v_0) = 0$ for each j. Again by Lemma 2, this means that $p(t; v_0)$ is identically 0. Since $p \in J$ is arbitrary, this means that $\mathbf{x}(t; v_0) \in \chi$ for each $t \in D_{v_0}$. Since $v_0 \in \chi$ is arbitrary, we have that χ is an invariant. □

Proof of Theorem 2 Part (a) follows directly from Theorem 1(b) and Lemma 5.

Concerning part (b), let $\chi = \psi_\phi$, the weakest precondition (algebraic variety) for $\phi = \mathbf{V}(\pi[V])$. Let $(V', J') = \text{post}(\chi, \pi)$. We first prove that $\pi[V] = \pi[V']$. Indeed, one one hand, by definition of I_χ we have that $\pi[V] \subseteq I_\chi$ and therefore: $\pi[V'] = \pi[\mathbb{R}^n] \cap I_\chi \supseteq \pi[V] \cap I_\chi = \pi[V]$, where the first equality comes from Theorem 1(a). On the other hand, $\psi \subseteq \chi$ by definition of χ, which implies $I_\chi \subseteq I_\psi$, therefore we have: $\pi[V] = \pi[\mathbb{R}^n] \cap I_\psi \supseteq \pi[\mathbb{R}^n] \cap I_\chi = \pi[V']$, where the first equality comes again from Theorem 1(a). Thus we have proved $\pi[V] = \pi[V']$. Now by Theorem 1(b) and $\pi[V] = \pi[V']$, we deduce that $J = J'$.

Now we prove that $\chi = \mathbf{V}(J)$. Since, by definition of I_χ, $\chi \longrightarrow [F]\,\mathbf{V}(I_\chi)$, we must have $\chi \subseteq \mathbf{V}(I_\chi)$; but $J = J' \subseteq I_\chi$ (again Theorem 1(b)), hence we have $\mathbf{V}(J) = \mathbf{V}(J') \supseteq \mathbf{V}(I_\chi) \supseteq \chi$, that is $\mathbf{V}(J) \supseteq \chi$. On the other hand, by part (a), $\mathbf{V}(J)$ is an algebraic invariant, that is $\mathbf{V}(J) \longrightarrow [F]\,\mathbf{V}(J)$; hence, since $\pi[V] \subseteq J$ and $\mathbf{V}(\pi[V]) \supseteq \mathbf{V}(J)$, we get $\mathbf{V}(J) \longrightarrow [F]\,\mathbf{V}(\pi[V]) = \phi$; the latter implies $\mathbf{V}(J) \subseteq \chi$, by definition of χ. In conclusion, $\chi = \mathbf{V}(J)$. $\square$

C Details for the collision avoidance example of Section 6

The following is a Gröbner basis of the invariant ideal J under the lexicographic order.

$$
\begin{aligned}
G \;=\; \{\; & (x_{10})^2 d_{20} + (x_{20})^2 d_{20} - 2x_{10}d_{20}x_1 + d_{20}x_1^2 - 2x_{20}d_{20}x_2 + d_{20}x_2^2 - 2x_{10}x_{20}d_1 + 2x_{20}x_1d_1 + 2x_{10}x_2d_1 - \\
& \quad 2x_1x_2d_1 + (x_{10})^2 d_2 - (x_{20})^2 d_2 - 2x_{10}x_1d_2 + x_1^2 d_2 + 2x_{20}x_2d_2 - x_2^2 d_2, \\
& (y_{10})^2 e_{20} + (y_{20})^2 e_{20} - 2y_{10}e_{20}y_1 + e_{20}y_1^2 - 2y_{20}e_{20}y_2 + e_{20}y_2^2 - 2y_{10}y_{20}e_1 + 2y_{20}y_1e_1 + 2y_{10}y_2e_1 - \\
& \quad 2y_1y_2e_1 + (y_{10})^2 e_2 - (y_{20})^2 e_2 - 2y_{10}y_1e_2 + y_1^2 e_2 + 2y_{20}y_2e_2 - y_2^2 e_2, \\
& \omega_1 x_{10} - \omega_1 x_1 - d_{20} + d_2, \\
& \omega_1 x_{20} - \omega_1 x_2 + d_{10} - d_1, \\
& \omega_2 y_{10} - \omega_2 y_1 - e_{20} + e_2, \\
& \omega_2 y_{20} - \omega_2 y_2 + e_{10} - e_1, \\
& x_{10}d_{10} + x_{20}d_{20} - d_{10}x_1 - d_{20}x_2 - x_{10}d_1 + x_1d_1 - x_{20}d_2 + x_2d_2, \\
& x_{20}d_{10} - x_{10}d_{20} + d_{20}x_1 - d_{10}x_2 + x_{20}d_1 - x_2d_1 - x_{10}d_2 + x_1d_2, \\
& (d_{10})^2 + (d_{20})^2 - d_1^2 - d_2^2, \\
& y_{10}e_{10} + y_{20}e_{20} - e_{10}y_1 - e_{20}y_2 - y_{10}e_1 + y_1e_1 - y_{20}e_2 + y_2e_2, \\
& y_{20}e_{10} - y_{10}e_{20} + e_{20}y_1 - e_{10}y_2 + y_{20}e_1 - y_2e_1 - y_{10}e_2 + y_1e_2, \\
& (e_{10})^2 + (e_{20})^2 - e_1^2 - e_2^2 \;\}
\end{aligned}
$$

D Proof of Theorem 4, Positivstellensatz and SOS programming in Section 7

Proof of Theorem 4 By contradiction, assume there are $w_0 \in X_0$ and $w_1 = \mathbf{x}(t_1; w_0) \in X_U$, for some $t_1 \in D_{w_0}$. We will show that $(w_0, w_1) \in \mathbb{R}^{2N}$ is a solution of (16), thus arriving at a contradiction. Indeed, by definition $g_i[\mathbf{x}_0/\mathbf{x}](w_0, w_1) = g_i(w_0) \geq 0$ and $h_j(w_0, w_1) = h_j(w_1) \geq 0$, for each $i = 1, ..., m$ and $j = 1, ..., n$. Consider now the trajectory of $\hat{F}$ originating from (w_0, w_0), that is $\hat{\mathbf{x}}(t; (w_0, w_0))$: note that, by definition of $\hat{F}$, $\hat{\mathbf{x}}(t; (w_0, w_0)) = (w_0\,,\, \mathbf{x}(t; w_0))$ for each $t \in D_{w_0}$. Now, since $\mathbf{V}(J) \supseteq \psi \supseteq (X_0 \times X_0) \cap Id$, we have $(w_0, w_0) \in \mathbf{V}(J)$, hence, by $\hat{F}$-invariance of J, $\hat{\mathbf{x}}(t; (w_0, w_0)) \in \mathbf{V}(J)$ for each $t \in D_{(w_0, w_0)}$ (Lemma 5). In particular, considering $t = t_1$, we have $\hat{\mathbf{x}}(t_1; (w_0, w_0)) = (w_0\,,\, \mathbf{x}(t_1; w_0)) = (w_0, w_1) \in \mathbf{V}(J)$. But this means $q_i(w_0, w_1) = 0$ for $i = 1, ..., k$. In conclusion, (w_0, w_1) is a solution of (16). $\square$

When working in algebraically closed fields, like $\mathbb{C}$, Hilbert's Nullstellensatz [9] implies that a system of polynomial equations P has no solution if and only if $1 \in \sqrt{\langle\, P \,\rangle}$. This gives a simple criterion to check if P is solvable. The following result, often considered as the real algebraic counterpart of Hilbert's Nullstellensatz, is due to Stengle [33]. Let us introduce the necessary terminology. In what follows, all polynomials are in $\mathbb{R}[\mathbf{x}]$ for some fixed $\mathbf{x} = (x_1, ..., x_N)$. A polynomial s is a *Sum of Squares (SOS)* if $s = \sum_j h_j^2$, for some polynomials h_j. Given a finite set of polynomials $A = \{f_1, ..., f_n\}$, the *cone* generated by A is $C(A) \triangleq \{\sum_{\sigma \subseteq \{1,...,n\}} s_\sigma \Pi_{j \in \sigma} f_j \ : \ s_\sigma \text{ are SOS } \}$.

Theorem D.1 (Positivstellensatz) *Let $A = \{f_1, ..., f_n\}$ and $B = \{g_1, ..., g_m\}$ be two sets of polynomials. The system of (in)equations $\{f_1 \geq 0, ..., f_n \geq 0, \ g_1 = 0, ..., g_m = 0\}$ has no solution in $\mathbb{R}^N$ if and only there are $f \in C(A)$ and $g \in \langle \ B \ \rangle$ such that $f + g + 1 = 0$.*

If one writes $f \in C(A)$ as $f = s_\emptyset + \sum_{\emptyset \neq \sigma \subseteq \{1,...,n\}} s_\sigma \Pi_{j \in \sigma} f_j$ and $g \in \langle \ B \ \rangle$ as $g = \sum_{i=1}^m h_i g_i$, then finding $f \in C(A)$ and $g \in \langle \ B \ \rangle$ such that $f + g + 1 = 0$ can be formulated as follows:

$$\text{Find polynomials } h_i\text{'s and SOS } s_\sigma\text{'s such that } -\left(\sum_{\emptyset \neq \sigma \subseteq \{1,...,n\}} s_\sigma \Pi_{j \in \sigma} f_j + \sum_{i=1}^m h_i g_i + 1 \right) \text{ is SOS.} \qquad (20)$$

Now a polynomial s is SOS if and only if there is a vector of monomials $Z = (\alpha_1, ..., \alpha_K)$ and a real symmetric positive semidefinite $K \times K$ matrix M such that $s = ZMZ^T$. Once bases of monomials have been fixed for each of the (unknown) polynomials s_σ and h_i, one can consider a relaxation of problem (20), whereby one searches for polynomials built from those bases satisfying (20) (empty bases are allowed). Problem (20) becomes in this way a *semidefinite programming problem* [21], with one variable for each (unknown) polynomial coefficient, and constraints given by the various positive semidefinite conditions and by the equation (20). In fact, there are tools, like SOSTOOLS [25], to efficiently convert relaxations of problem (20) into a semidefinite programming problem and then try to solve it via numerical techniques. If successful, the obtained SOS polynomial is a certificate of insolvability of the original system. Although the number of terms in (20) is potentially exponential in n, experience has shown that, in practice, this technique tends to yield short (low degree) certificates, if the original system is actually not solvable.

For the continuous semialgebraic systems in Examples 6 and 7, denoting by p_0, p_U the polynomials defining X_0 and X_U in each case, problem (20) takes the following concrete form

$$\text{Find } h_1, h_2 \text{ and SOS } s_1, s_2, s_3 \text{ such that } -(s_1 p_0 + s_2 p_U + s_3 p_0 p_U + h_1 q_1 + h_2 q_2 + 1) \text{ is SOS.} \qquad (21)$$

For Example 6, fixing a maximum degree of 1 for the h_i's and of 2 for the s_σ's and running SOSTOOLS under Matlab[16] we solve (21), finding the following polynomials, which yield a low-degree certificate:

$$\begin{aligned}
s_1 &= 2.8733x^2 - 0.15408xx_0 - 0.6222xy - 0.11367xy_0 - 0.81927xz + 4.2254x_0^2 - 1.4885x_0y - 2.4633x_0y_0 - \\
&\quad 0.61469x_0z + 1.5901y^2 - 1.3772yy_0 - 1.4399yz + 4.071y_0^2 - 0.58541y_0z + 3.1195z^2 \\
s_2 &= 1.7131x^2 + 0.063147xx_0 - 0.33468xy + 0.074991xy0 + 0.17391xz + 1.2985x_0^2 - 0.51261x_0y + 0.19351x_0y_0 + \\
&\quad 0.15971x_0z + 0.5814y^2 - 0.53211yy_0 - 0.48663yz + 1.3512y_0^2 + 0.17931y_0z + 1.9153z^2 \\
s_3 &= 0 \\
h_1 &= 0.49855x + 0.21264x_0 - 0.29621y + 0.18602y_0 + 0.35564z + 14.8214 \\
h_2 &= 13.096x + 9.6921x_0 + 0.013631y - 36.2677y_0 + 22.7601z - 16.3507\,.
\end{aligned}$$

This takes about 0.4s on a Core i5 machine under Windows 10. A certificate for Example 7 is found in a similar way; we omit here the lengthy description of the corresponding polynomials.

[16]Matlab scripts for both examples are available at `https://github.com/micheleatunifi/postconditions/blob/master/SOSSafety.m`.