



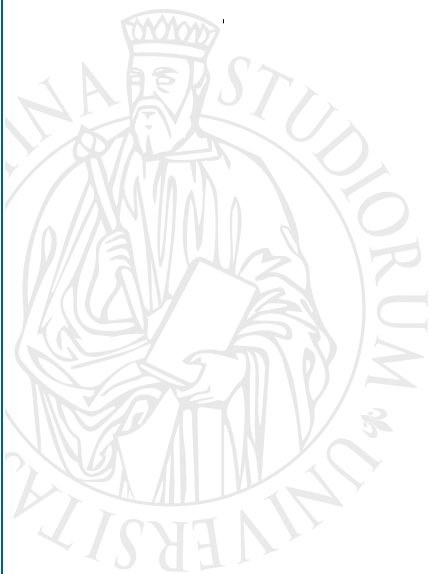
UNIVERSITÀ
DEGLI STUDI
FIRENZE

DISIA

DIPARTIMENTO DI STATISTICA,
INFORMATICA, APPLICAZIONI
"GIUSEPPE PARENTI"

**Algorithms for exact and approximate
linear abstractions
of polynomial continuous systems**

Michele Boreale



**DISIA WORKING PAPER
2018/03**

© Copyright is held by the author(s).

Algorithms for exact and approximate linear abstractions of polynomial continuous systems*

Michele Boreale
Università di Firenze

Dipartimento di Statistica, Informatica, Applicazioni (DiSIA) “G. Parenti”

Abstract

A polynomial continuous system $\mathcal{S} = (F, X_0)$ is specified by a polynomial vector field F and a set of initial conditions X_0 . We study polynomial changes of bases that transform \mathcal{S} into a linear system, called *linear abstractions*. We first give a complete algorithm to find all such abstractions that fit a user-specified template. This requires taking into account the algebraic structure of the set X_0 , which we do by working modulo an appropriate invariant ideal. Next, we give necessary and sufficient syntactic conditions under which a *full* linear abstraction exists, that is one capable of representing the behaviour of the individual variables in the original system. We then propose an approximate linearization and dimension-reduction technique, that is amenable to be implemented “on the fly”. We finally illustrate the encouraging results of a preliminary experimentation with the linear abstraction algorithm, conducted on challenging systems drawn from the literature.

Keywords: ordinary differential equations, abstraction, invariants, linearization

1 Introduction

Recent years have seen a surge of interest in computational models based on ordinary differential equations (ODE’s), due to the prominence of such fields as system biology [5] and hybrid systems [20, 27, 16, 22].

In general terms, the dynamics of a continuous system is specified by a system of generally nonlinear, ordinary differential equations (ODE’s) $\dot{\mathbf{x}} = F(\mathbf{x})$ and a set of initial states X_0 . In the present paper we focus on the case where F is a polynomial vector field. A vast range of functions can be expressed either directly or coded up in this formalism. Unfortunately, polynomial systems very seldom admit an explicit solution. Therefore, one seeks for indirect ways to understand the system’s behaviour. In the present paper, we study algorithms to build tractable, linear abstractions of polynomial systems. These abstractions may be useful for the purpose of both numerical simulations and formal verification of safety properties. A special case of linear abstractions are the so called *scale consecutions* [21, 17], a type of invariant that can be used to establish safety properties of a system. We give a more detailed account of our work below.

Linear abstractions have been long studied in the field of dynamical systems. The idea is to devise a simpler linear system of ODE’s, easier to deal with, but equivalent to, or at least preserving some interesting properties of, the original system. In particular, Sankaranarayanan [23, 24]

*A slightly abridged version of this work appears in the *Proc. of HSCC’18: 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, April 11–13, 2018, Porto, Portugal. ISBN978-1-4503-5642-8/18/04, ACM, 2018. Author’s address: Michele Boreale, Dipartimento di Statistica, Informatica, Applicazioni (DiSIA) “G. Parenti”, Viale Morgagni 65, I-50134 Firenze, Italy. E-mail: michele.boreale@unifi.it.

offers algorithms to build linear abstractions through polynomial *change of basis* transformations of the original system. We extend this work in two directions.

As a first contribution, we provide an algorithm that, under reasonable assumptions, is *complete*: it finds all and only the linear abstractions given by polynomial changes of basis that fit a given template. Doing so requires taking appropriately into account the structure of the set of initial states, X_0 . To appreciate what is at stake here, let us look at a simple example. Consider a system \mathcal{S} where $\dot{x} = y^2$, $\dot{y} = xy$ and X_0 is the set of pairs in the line $x = y$. In every trajectory starting from X_0 , one has $x(t) = y(t)$. Thus one has, for instance, $\frac{d}{dt}(y^2 - xy) = 0$, which makes $y^2 - xy$ a linear abstraction of \mathcal{S} , in fact a strong scale consecution. These abstractions escape existing algorithms based on polynomial changes of basis [23, 24], essentially because those algorithms have no way of incorporating the information that $x(t) = y(t)$ starting from X_0 . We recover completeness working modulo an appropriate polynomial ideal J representing X_0 , rather than with pure syntax. Continuing with \mathcal{S} , one notes that the Lie derivative of $y^2 - xy$ is 0 modulo $J = \langle x - y \rangle$. Once a basis of J is given, the method is entirely based on simple linear algebra and constraints propagation. Preliminary experiments indicate that the resulting algorithm works reasonably well on nontrivial systems drawn from the literature (see Section 5). Going further, one can ask under which conditions a *full* linear abstraction exists, that is one capable of representing the behaviour of the individual variables in the original system – hence of any combination of them. We provide a syntactic necessary and sufficient condition for this to be the case. This condition can be checked against any Gröbner basis of J .

As a second contribution, we provide a method for linearization and dimension reduction, which permits trading off approximation with computational cost. One disadvantage of exact linearization methods is that the dimension of the linear abstraction, even when it exists finite, can be much higher than the original nonlinear system’s. Moreover, often no polynomial change of basis exists yielding a linear abstraction. We propose a linearization method that, for a prescribed $m \geq 1$, returns an abstraction of dimension m that approximates a given function of the trajectory. At the very least, the approximation error is guaranteed to decrease as $O(t^m)$ around the origin (under mild conditions). Results to bound the absolute error in a given closed interval are also provided. We argue that this technique might give good results when used with an ansatz of Chebyshev polynomials. Again, the method is mostly based on simple linear algebra. Moreover, it is amenable to an “on the fly” implementation that avoids the explicit construction of large matrices.

Related work The concept of scale consecution has been proposed by Sankaranarayanan et al. [21] and subsequently generalized by Matringe et al. [17]. The mentioned papers by Sankaranarayanan [23, 24] build on these works and offer fixed-point algorithms to compute changes of basis yielding linear abstractions. These algorithms are not complete in the sense made precise above. Scale consecutions are generalized by *Darboux polynomials*, aka polynomial consecutions, which cannot be captured by linear abstractions. See e.g. [21, 12] and more recently [13] for examples of use of such polynomials in the context of verification.

Our work is also strongly related to methods for finding ideal/algebraic invariants, which are more general than consecutions. In fact, our method requires that the ideal J we work up to, be an invariant whose variety includes X_0 . Such invariants can in principle be obtained by any method proposed in the literature. See for instance Ghorbal and Platzer’ technique based on differential radical invariants [10], and references therein. More recently, Boreale [7] has introduced a (in a precise sense, complete) method to obtain a small algebraic invariant that includes a given algebraic X_0 . The abstraction method in the present paper, therefore, can be seen as building on all such approaches.

Our linearization techniques are also related to *Carleman embedding* [14], which can be used to transform a given nonlinear system into an *infinite* linear system. Bellman and Richardson [4] have studied in some special cases the effect of truncating the obtained infinite linear system at a prescribed cutoff. Our approximate linearization technique draws from and generalizes these ideas. But in order to achieve dimension reduction, we blend Carleman embedding with orthogonal projection techniques. Basically, we pursue dimension reduction via projection, by working in the embedding coordinate space. In nonlinear Control Theory as well there is a fair amount of work based on Carleman (bi)linearization followed by projection. These approximations focus on the frequency domain; see e.g. [15, 2]. Another approach relies on approximate linearization via truncated Taylor expansion, repeated at various points along a trajectory, to keep the approximation error globally small: a technique known as *trajectory piece-wise linear* (TPWL) model order reduction [18]. See [3] for a recent survey. The relation of our work with these techniques deserves further investigation. In particular, we wonder whether our approximate linearization technique might conveniently serve as a building block of TPWL and similar strategies.

Our approach is very different from *discrete* abstractions, where the state space is subdivided into small regions, each of which can be approximated by a linear system. See the discussions in [23, 24, 25] and the references therein. The number of regions can grow exponentially as the number of dimensions grows, making the subdivision very expensive. Techniques based on changes of basis might offer an alternative in some cases. Here we offer some preliminary experimental evidence that our techniques might be useful, but leave a systematic comparison with discretization techniques for future work.

Structure of the paper The rest of the paper is organized as follows. In Section 2 we introduce the necessary preliminary notions and terminology. Section 3 presents results on exact linear abstractions. Section 4 introduces our approximate linearization and dimension reduction technique. Section 5 presents the results of the experiments we have conducted on some systems drawn from the literature, using the abstraction technique of Section 3. We draw some concluding remarks and discuss further work in Section 6. For ease of reading, a few technical proofs and an additional example have been confined to separate appendices.

2 Preliminaries

Throughout the paper ‘vectors’ are considered as *column* vectors unless otherwise specified. For L a vector or matrix, we denote by L^T its transpose.

Continuous systems and linear abstractions Fix $N \geq 1$ and a tuple of distinct variables $\mathbf{x} = (x_1, \dots, x_N)^T$. A system of polynomial ODE’s has the form

$$\dot{\mathbf{x}} = F(\mathbf{x}) \tag{1}$$

where $F = (f_1, \dots, f_N)^T$ is a *vector field* of multivariate polynomials in \mathbf{x} . The functions f_i are called *drifts* in this context. Once N and \mathbf{x} are fixed, we can identify a system of ODE’s with its vector field F . Given an initial condition $x_0 \in \mathbb{R}^N$, we have an *initial value problem*. A solution of the problem is a differentiable function $\mathbf{x}(t) : U \rightarrow \mathbb{R}^N$, defined on some nonempty open interval $U \subseteq \mathbb{R}$ containing 0, such that $\mathbf{x}(0) = x_0$ and equation (1) is fulfilled: $\frac{d}{dt}\mathbf{x}(t) = F(\mathbf{x}(t))$ for each $t \in U$. By the Picard-Lindelöf theorem [1], there exists a nonempty open interval U containing 0, over which there is a *unique* solution, say $\mathbf{x}(t) = (x_1(t), \dots, x_N(t))^T$, of the problem. In our case F is infinitely often differentiable, so the solution is seen to be *analytic* in U : each

$x_i(t)$ admits a Taylor series expansion in a neighborhood of 0. For definiteness, we will take the domain of definition U of $\mathbf{x}(t)$ to be the largest symmetric open interval where the Taylor expansion from 0 of each of the $x_i(t)$ converges (possibly $U = \mathbb{R}$). 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 $x_0 \in \mathbb{R}^N$. We shall write them as $\mathbf{x}(t; x_0)$ and U_{x_0} , respectively, whenever we want to make this dependence explicit in the notation.

For any analytic function $g : \mathbb{R}^N \rightarrow \mathbb{R}$, the function $g(\mathbf{x}(t; x_0)) : U \rightarrow \mathbb{R}$, obtained by composing g as a function with the time trajectory $\mathbf{x}(t; x_0)$, is analytic. We let $g(t; x_0)$ **denote the extension of $g(\mathbf{x}(t; x_0))$** over the largest symmetric open time interval of convergence (possibly coinciding with \mathbb{R}) of its Taylor expansion from 0. A *continuous system* is a pair of a vector field F and a subset of *initial conditions* $X_0 \subseteq \mathbb{R}^N$:

$$\mathcal{S} = (F, X_0). \quad (2)$$

Definition 1 (linear abstraction) Consider the continuous system \mathcal{S} as defined in (2). A vector of analytic functions $\phi = (\phi_1, \dots, \phi_k)^T$, with $\phi_i : \mathbb{R}^N \rightarrow \mathbb{R}$, is a linear abstraction for \mathcal{S} if there is a $k \times k$ real matrix A such that, for each $x_0 \in X_0$ the function $\phi(t; x_0)$ satisfies the equation (with $\mathbf{z} = (z_1, \dots, z_k)^T$ a vector of distinct variables):

$$\dot{\mathbf{z}} = A\mathbf{z}. \quad (3)$$

A linear abstraction of dimension $k = 1$ is called constant scale consecution of \mathcal{S} .

The function vector ϕ will be referred to as the (*change of*) *basis* of the abstraction. The matrix A in (3) will be referred to as the abstraction matrix. Seen as a function $\mathbb{R}^N \rightarrow \mathbb{R}^k$, the basis ϕ transforms the original system $\mathcal{S} = (F, X_0)$ into a linear system $\mathcal{T} = (H, Y_0)$, where the vector field H is the linear function described by A and $Y_0 = \phi(X_0)$. The elements of ϕ are drawn from any chosen *ansatz* (class) $\mathcal{A} = \{\phi_1, \phi_2, \dots\}$ of analytic functions.

A first, obvious use of linear abstractions is the effective, analytic or numerical, computation of functions g of the trajectory that arise as linear combinations of the functions in ϕ , without having to explicitly solve the original system \mathcal{S} , which can be impossible or costly. Note, however, that the dimension k of the linear system (3) can be (much) larger than the dimension N of the original system. Dimension reduction, possibly at the cost of approximation, will be discussed in Section 4. Under certain conditions, linear abstractions are also useful to reason about properties of the original system such as stability, a theme that will not be covered here; see e.g. [23] and references to the specialized literature therein.

Linear abstractions are also useful to prove safety properties of \mathcal{S} . Indeed, it is easy to see that for a (polynomial) abstraction ϕ , if $X_0 \subseteq \mathbf{V}(\phi) \triangleq \mathbf{V}(\{\phi_1, \dots, \phi_k\})$ then no trajectory starting from X_0 can ever leave $\mathbf{V}(\phi)$. In order to prove unreachability of a given unsafe set X_u , it is therefore sufficient to find an abstraction ϕ such that $X_0 \subseteq \mathbf{V}(\phi)$ and $\mathbf{V}(\phi) \cap X_u = \emptyset$. We refer the reader to [23, 7] for discussions concerning this aspect. We point out again that our definition of linear abstraction is more flexible than the one considered in [23, 24]. First, we explicitly take into account the set of initial conditions X_0 . Second, the definition is semantic - given in terms of time derivative -, as opposed to syntactic - in terms of Lie derivatives (see below), as often found in other works. Let us illustrate the extra flexibility provided by this definition with the following simple example.

Example 1 Consider the system $\mathcal{S} = (F, X_0)$ in the variables $\mathbf{x} = (x, y)^T$, where $F = (y^2, xy)^T$ and $X_0 = \{(\lambda, \lambda)^T \in \mathbb{R}^2\}$. It is not difficult to see that, for $x_0 \in X_0$, $x(t; x_0) = y(t; x_0)$,

hence $x - y$ is a linear abstraction, and in fact a constant scale consecution for \mathcal{S} . Similarly, e.g. $y^2 - xy$ and similar combinations all are constant scale consecutions for \mathcal{S} . Note that by taking the Lie derivative (see below) of each of these polynomials the degree gets increased. This implies that if we consider the system \mathcal{S}_0 with $X_0 = \mathbb{R}^2$, these polynomials fail to form linear abstractions.

Lie derivatives Given a differentiable function $g : E \rightarrow \mathbb{R}$, for some open set $E \subseteq \mathbb{R}^N$, the Lie derivative of g along F is the function $E \rightarrow \mathbb{R}$ defined as: $\mathcal{L}_F(g) \triangleq \langle \nabla g, f \rangle = \sum_{i=1}^N (\frac{\partial g}{\partial x_i} \cdot f_i)$. More generally, we can define inductively $\mathcal{L}_F^{(0)}(g) \triangleq g$ and $\mathcal{L}_F^{(j+1)}(g) \triangleq \mathcal{L}_F(\mathcal{L}_F^j(g))$. The Lie derivative of the sum $h + g$ and product $h \cdot g$ functions obey the familiar rules: $\mathcal{L}_F(h + g) = \mathcal{L}_F(h) + \mathcal{L}_F(g)$ and $\mathcal{L}_F(h \cdot g) = h \cdot \mathcal{L}_F(g) + \mathcal{L}_F(h) \cdot g$. When $g = p \in \mathbb{R}_d[\mathbf{x}]$ 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 vector field F as a purely syntactic mechanism, that is as a function $\mathcal{L}_F : \mathbb{R}[\mathbf{x}] \rightarrow \mathbb{R}[\mathbf{x}]$. In particular, if $p = x_i$ (projection on the i -th component), $\mathcal{L}_F(p) = f_i$.

The connection between the Lie derivatives of g along F and the system (1) is given by the following equalities that can be readily checked. For any $x_0 \in \mathbb{R}^N$, with the notation introduced before Definition 1, we have: $g(t; x_0)|_{t=0} = g(x_0)$ and $\frac{d}{dt}g(t; x_0) = (\mathcal{L}_F(g))(t; x_0)$. More generally, we have the following equation for the j -th derivative of $g(t; x_0)$ ($j = 0, 1, \dots$): $\frac{d^{(j)}}{dt^j}g(t; x_0) = (\mathcal{L}_F^{(j)}(g))(t; x_0)$. In the sequel, we shall omit the subscript F from \mathcal{L}_F when clear from the context.

Algebraic geometry concepts 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) $p_1, \dots, p_m \in I$ and $h_1, \dots, h_m \in \mathbb{R}[\mathbf{x}]$ implies $\sum_{i=1}^m h_i p_i \in I$. The ideal generated by a set $P \subseteq \mathbb{R}[\mathbf{x}]$ is $\langle P \rangle \triangleq \{ \sum_{i=1}^m h_i p_i : m \geq 0 \text{ and } h_i \in \mathbb{R}[\mathbf{x}], p_i \in P \text{ for } i = 1, \dots, m \}$. 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 *basis* for I . Hilbert's basis theorem implies that: (a) any ideal $I \subseteq \mathbb{R}[\mathbf{x}]$ has a finite basis; (b) any infinite ascending chain of ideals $I_0 \subseteq I_1 \subseteq \dots$ stabilizes in a finite number of steps (*ascending chain condition*). Once a *monomial order* (e.g. lexicographic) is fixed, a multivariate version of polynomial division naturally arises – see [9, Ch.2, Sect.3] for the precise definition. A *Gröbner basis* of I (w.r.t. a fixed monomial order) is a finite basis 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 the following property: $p \in I$ iff $r = 0$. As a consequence, given a Gröbner basis G of I , the *ideal membership* problem $p \in I$ can be decided¹. 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: $\mathbf{V}(P) \triangleq \{x_0 \in \mathbb{R}^N : p(x_0) = 0 \text{ for each } p \in P\}$ is the *algebraic variety* represented by P . We will have in general more than one set of polynomials P representing the same algebraic variety A . Ideals and algebraic varieties are connected as follows. For any set $A \subseteq \mathbb{R}^N$, the set of polynomials that vanish on A , $\mathbf{I}(A) \triangleq \{p : p(x_0) = 0 \text{ for each } x_0 \in A\}$, is the ideal induced by A . Note that both \mathbf{V} and \mathbf{I} are

¹Provided the involved coefficients can be finitely represented, for instance are rational.

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 $\mathbf{I}(\mathbf{V}(J)) \supseteq J$. If the last inclusion is actually an equality, we will say that J is a *real radical*. Given a finite set of polynomials P or $\psi = \mathbf{V}(P)$, there exists algorithms, albeit very costly, to build the real radical $\mathbf{I}(\psi)$. In some cases of interest, such as ψ or P a singleton, it is fortunately easy to determine the real radical.

Invariants Let F be a polynomial vector field in the variables \mathbf{x} . We say an ideal J is *invariant* under F if whenever $p \in J$ then $\mathcal{L}_F(p) \in J$. Correspondingly, we say an algebraic variety A is *invariant* under F if whenever $x_0 \in A$ then $\mathbf{x}(t; x_0) \in A$ for each t in the domain of definition U_{x_0} . It is easy to prove that A is an invariant variety if and only if there exists an invariant ideal J such that $A = \mathbf{V}(J)$; see e.g. [7]. Finally, we say a polynomial p is invariant w.r.t. F and x_0 if $p(t; x_0)$ is identically 0; such a p represents a system's conservation law that is valid from the initial state x_0 . We record the following easy fact for future reference (see [6]).

Lemma 1 *Let J be an invariant ideal under F . For each $p \in J$ and $x_0 \in A = \mathbf{V}(J)$, p is an invariant polynomial w.r.t. F and x_0 .*

3 Linear abstractions modulo an invariant

In this section, we confine ourselves to polynomial changes of basis. In other words, the functions in ϕ will be drawn from an ansatz $\mathcal{A} \subseteq \mathbb{R}[\mathbf{x}]$. We can compactly represent such an \mathcal{A} as follows. Let us arbitrarily fix $M \geq 1$ distinct monomials, $\alpha_1, \alpha_2, \dots, \alpha_M$. A *polynomial template* π is a polynomial with linear expressions² ℓ of the parameters a_i in $\mathbf{a} = (a_1, \dots, a_M)$ as coefficients, say $\pi = \sum_{j=1}^M \ell_j \alpha_j$. For any template π and $v = (\lambda_1, \dots, \lambda_M)^T \in \mathbb{R}^M$, let $\pi[v] \in \mathbb{R}[\mathbf{x}]$ be the polynomial obtained by replacing each a_i with λ_i . For $S \subseteq \mathbb{R}^M$, we let $\pi[S] \subseteq \mathbb{R}[\mathbf{x}]$ be the set $\{\pi[v] : v \in S\}$. Note that if S is a vector space, $\pi[S]$ is in turn a vector space over \mathbb{R} of dimension $\leq M$.

Let $J \subseteq \mathbb{R}[\mathbf{x}]$ be an invariant ideal. This will represent X_0 or a convenient overapproximation of X_0 . Assume a monomial ordering on the polynomial ring $\mathbb{R}[\mathbf{a}, \mathbf{x}]$ such that $a_i > x_j$ for each i, j . Let $G \subseteq \mathbb{R}[\mathbf{x}]$ be a Gröbner basis for J according to the fixed monomial ordering; note that G is a Gröbner basis also in the larger ring $\mathbb{R}[\mathbf{a}, \mathbf{x}]$. Moreover, we can assume that G is *reduced*: whenever convenient, this will allow us to write $p \bmod J$, rather than $p \bmod G$ – indeed the result of division by G does not actually depend on the specific reduced G that is chosen, but only on the chosen monomial ordering. Consider the sequence of polynomials $r_j \in \mathbb{R}[\mathbf{a}, \mathbf{x}]$, $j \geq 0$, defined as follows:

$$r_0 \stackrel{\Delta}{=} \pi \bmod J \tag{4}$$

$$r_{j+1} \stackrel{\Delta}{=} \mathcal{L}(r_j) \bmod J. \tag{5}$$

The following lemma, saying that reduction modulo G and substitutions commute, is proved in [7].

Lemma 2 *Let π be a template and $G \subseteq \mathbb{R}[\mathbf{x}]$ a Gröbner basis according to a monomial ordering as prescribed above. Then $r = \pi \bmod G$ is linear in \mathbf{a} . Moreover, for any $v \in \mathbb{R}^M$, $\pi[v] \bmod G = r[v]$.*

²For instance, $\ell = 5a_1 + 42a_2 - 3a_3$ is one such expression. Linear expressions with a constant term, such as $2 + 5a_1 + 42a_2 - 3a_3$ are *not* allowed.

We now define a descending chain of vector spaces $V_0 \supseteq V_1 \supseteq \dots$ in \mathbb{R}^M , as follows. For $i \geq 0$

$$V_i \triangleq \{v \in \mathbb{R}^M : r_j[v] \in \pi[\mathbb{R}^M] \text{ for } j = 0, \dots, i\}. \quad (6)$$

That each V_i is a vector space, stems from the linearity of the r_j 's in the parameters in \mathbf{a} . The chain $V_0 \supseteq V_1 \supseteq \dots$ must stabilize at some finite m such that $V_{m+j} = V_m$ for each $j \geq 1$. The following lemma provides a criterion for detecting stabilization. In what follows, for any $p \in \mathbb{R}[\mathbf{a}, \mathbf{x}]$ we shall abbreviate $\mathcal{L}(p) \bmod J$ as $\mathcal{L}_J(p)$; for $i \geq 0$, $\mathcal{L}_J^{(i)}$ is defined inductively as expected, once $\mathcal{L}^{(0)}(p) = p \bmod J$. Let us say that a set $S \subseteq \pi[\mathbb{R}^M]$ is *closed under \mathcal{L} and J* if for each $p \in S$, $(p \bmod J) \in S$ and $\mathcal{L}_J(p) \in S$.

Lemma 3 *Suppose that $V_{m+1} = V_m$. Then $V_{m+j} = V_m$ for each $j \geq 1$. Moreover, $\pi[V_m]$ is closed under \mathcal{L} and J .*

When $V_{m+1} = V_m$, we return V_m as the result of the algorithm, written $\text{LINABS}(\pi, J) = V_m$.

Theorem 1 *Let $\text{LINABS}(\pi, J) = V_m$. Then $\pi[V_m]$ is the largest subspace of $\pi[\mathbb{R}^M]$ that is closed under \mathcal{L} and J .*

PROOF As V_m is a vector space, so is $\pi[V_m]$, as already remarked. Lemma 3 implies that $\pi[V_m]$ is closed under \mathcal{L} and J . Conversely, consider any vector space $S \subseteq \pi[\mathbb{R}^M]$, closed under \mathcal{L} and J . Consider any $p = \pi[v] \in S$, with $v \in \mathbb{R}^M$. We show that $v \in V_m$, thus proving the inclusion $S \subseteq \pi[V_m]$. As S is closed under \mathcal{L} and J , for each $i \geq 0$ we have $\mathcal{L}_J^{(i)}(\pi[v]) \in S$: this is proven by induction on i . But, by applying Lemma 2 (see (18) in the appendix), we have that for each $i \geq 0$, $\mathcal{L}_J^{(i)}(\pi[v]) = r_i[v]$; thus in particular $r_i[v] \in \pi[\mathbb{R}^M]$. This proves that $v \in V_m$. \square

Assume $\text{LINABS}(\pi, J) = V_m$ and let $\{p_1, \dots, p_k\}$ be a basis of $\pi[V_m]$ – this can be readily obtained from any basis of V_m . For each $1 \leq i \leq k$, we can express the Lie derivative of p_i modulo J uniquely as $\mathcal{L}(p_i) \bmod J = \sum_{j=1}^k a_{ij} p_j$ for suitable coefficients a_{ij} ; let A be the square matrix of such coefficients³. The next result shows that $\pi[V_m]$ includes all and only the linear abstractions whose elements are instances of π , provided J is consistent with the initial set X_0 . Note in order to apply the corollary, it is not required that $X_0 = \mathbf{V}(J)$, although this may be a typical use-case. Also, in the “all” part, J is required to be a radical; note that any algebraic variety A is represented by the real radical $\mathbf{I}(A)$, although building it may be highly nontrivial. In this part, a mild syntactic condition⁴ on π is required: we say r_0 is an instance of π if $r_0[\mathbb{R}^M] \subseteq \pi[\mathbb{R}^M]$.

Corollary 1 (completeness) *Let $\text{LINABS}(\pi, J) = V_m$. Assume $\mathbf{V}(J) \supseteq X_0$. Then any basis $\{p_1, \dots, p_k\}$ of $\pi[V_m]$, once ordered, is a linear abstraction for \mathcal{S} , with abstraction matrix given by A as defined above.*

Conversely, assume that r_0 is an instance of π . Assume that $\mathbf{V}(J) \subseteq X_0$ and that J is a real radical. Then the elements of any linear abstraction $(q_1, \dots, q_h)^T$ for \mathcal{S} such that $q_i \in \pi[\mathbb{R}^M]$ ($1 \leq i \leq h$) are in $\pi[V_m]$.

PROOF Concerning the first part, consider any p_i in the given basis and write $\mathcal{L}(p_i) = q + r$ with $q \in J$ and $\mathcal{L}(p_i) \bmod J = r \in \pi[V_m]$, say $r = \sum_{j=1}^k a_{ij} p_j$. For any $x_0 \in X_0 \subseteq \mathbf{V}(J)$, the function

³Note that such coefficients do depend on the specific basis G that is chosen.

⁴The condition is satisfied, for example, if π is formed with all the monomials up to a given degree.

$p(t; x_0)$ satisfies $\frac{d}{dt}p(t; x_0) = \mathcal{L}(p)(t; x_0) = q(t; x_0) + r(t; x_0) = \sum_{j=1}^k a_{ij}p_j(t; x_0)$, as $q(t; x_0)$ is identically 0 (Lemma 1). Hence (p_1, \dots, p_k) satisfies (3) as required.

Conversely, consider a linear abstraction for \mathcal{S} , say $(q_1, \dots, q_h)^T$. For each i , let $\bar{q}_i \triangleq q_i \bmod J$: note that as, for some v , $\bar{q}_i = \pi[v] \bmod J = r_0[v]$ (Lemma 2), and r_0 is an instance of π , we must have $\bar{q}_i \in \pi[\mathbb{R}^M]$. Consider now the subspace $S \subseteq \pi[\mathbb{R}^M]$ spanned by $\{q_1, \dots, q_h\} \cup \{\bar{q}_1, \dots, \bar{q}_h\}$. By Theorem 1, the thesis will follow if we show that S is closed under \mathcal{L} and J . S is clearly closed under J . We check now that S is closed under \mathcal{L}_J . Consider any q_i . First, observe that by hypothesis there are real coefficients b_{ij} 's such that, for each $x_0 \in \mathbf{V}(J)$, we have, in a neighborhood of $t = 0$:

$$\frac{d}{dt}q_i(t; x_0) = \mathcal{L}(q_i)(t; x_0) = \sum_{j=1}^h b_{ij}q_j(t; x_0).$$

In particular, taking $t = 0$ in the second equality above, we deduce that $\mathcal{L}(q_i)(x_0) = \sum_{j=1}^h b_{ij}q_j(x_0)$. Since this holds for each $x_0 \in \mathbf{V}(J)$, we have that the polynomial $s \triangleq \mathcal{L}(q_i) - \sum_{j=1}^h b_{ij}q_j$ vanishes on $\mathbf{V}(J)$, that is $s \in \mathbf{I}(\mathbf{V}(J)) = J$, where the last equality follows from the fact that J is a real radical. This implies that $s \bmod J = 0$, or equivalently that

$$\mathcal{L}(q_i) \bmod J = \sum_{j=1}^h b_{ij}\bar{q}_j.$$

This shows that $\mathcal{L}_J(q_i) \in S$. On the other hand, by linearity of $\mathcal{L}(\cdot)$ and invariance of J , it is immediate to check that $\mathcal{L}_J(\bar{q}_i) = \mathcal{L}_J(q_i)$, which completes the proof for this case. \square

Remark 1 (result template) *Given a template π and $v \in \mathbb{R}^M$, checking if $\pi[v] \in \pi[V_m]$ is equivalent to checking if $v \in V_m$: this can be effectively done knowing a basis of the vector space V . In practice, it is computationally more convenient to represent the whole set $\pi[V_m]$, with V_m returned by the algorithm, compactly in terms of a new m -parameters result template π' such that $\pi'[\mathbb{R}^m] = \pi[V_m]$. The result template π' can in fact be built starting from π and propagating the linear constraints on \mathbf{a} implied by (6) as the r_j 's are generated. When we take $J = \{0\}$, which corresponds to $X_0 = \mathbf{V}(J) = \mathbb{R}^N$, this algorithm degenerates into that of [23, 24].*

Example 2 *Consider again the system \mathcal{S} of Example 1 and let π be the template given by all monomials of degree ≤ 2 . Note that $X_0 = \mathbf{V}(J)$ where $J = \langle x - y \rangle$ is an invariant ideal, and is moreover a real radical. Running $\text{LINABS}(\pi, J)$, the chain (6) stabilizes at $m = 1$, that is $V_1 = V_2 = \dots$. One can check that $\pi[V_1] = \pi'[\mathbb{R}^4]$, where $\pi' = a_1 + a_2(y - x) + a_3(y^2 - x^2) + a_4(xy - x^2)$ is the result template. Applying Corollary 1, we deduce that the space spanned by the four polynomial terms of this template represents all and only the linear abstractions of degree ≤ 2 for \mathcal{S} . In particular, all of the abstractions are constant scale consecutions for \mathcal{S} .*

On the other hand, consider the system $\mathcal{S}_0 = (F, \mathbb{R}^2)$; that is, no condition is imposed on the initial states. Running $\text{LINABS}(\pi, \{0\})$, we obtain the result template $\pi'' = a_0$. As also $\{0\}$ is trivially radical, Corollary 1 implies there is no nontrivial linear abstraction for \mathcal{S}_0 .

We end this section with a result of theoretical interest about the expressive power of linear abstractions. We say a vector of polynomials $(p_1, \dots, p_k)^T$ is a *full linear abstraction* for \mathcal{S} if for every polynomial $p \in \mathbb{R}[\mathbf{x}]$ there are coefficients $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ such that, for each $x_0 \in X_0$, $p(t; x_0) = \sum_{j=1}^k \lambda_j p_j(t; x_0)$. Note that a full linear abstraction is necessarily also a linear

abstraction for \mathcal{S} in the sense of Definition 1. The following result provides us with a syntactic characterization of systems that admit full abstractions via a polynomial change of basis. For a set of polynomials S , $\text{LM}(S)$ denotes the set of *leading monomials* in S , that is, the set of maximal monomials in each polynomial in S , according to the chosen monomial ordering [9].

Theorem 2 (full abstraction) *Let G be any Gröbner basis of J and consider the set of monomials $P = \{\alpha : \alpha \notin \langle \text{LM}(G) \rangle\}$. Assume $\mathbf{V}(J) \supseteq X_0$. If P is finite then it forms a full linear abstraction for \mathcal{S} , once it is totally ordered.*

Conversely, assume that $\mathbf{V}(J) \subseteq X_0$ and that J is a real radical. If there is a full linear abstraction for \mathcal{S} then P is finite.

Given G , it is easy to syntactically check if the set P mentioned in the statement is finite: it is necessary and sufficient that for each i , $x_i^j \in \text{LM}(G)$ for some $j > 0$. For instance, it is readily checked that, as this condition is not true for $G = \{x - y\}$, the system \mathcal{S} in Example 2 does not admit a full linear abstraction.

4 Approximation and dimension reduction

In this section, we consider a method for approximate linearization of the system (1). Rather than approximating the system as a whole, we will be searching reduced linear approximations of a *target* function g of a trajectory $\mathbf{x}(t; x_0)$, that is $g(t; x_0)$. The differential equations of the obtained linear system will not depend on the initial state x_0 , although the approximation error will, in general. The method is amenable to an “on the fly” implementation, in the following sense: it only requires building the Lie derivatives of g until a prescribed order m . This order coincides with the dimension of the obtained linear system and, at the very least, guarantees that the error is $O(t^m)$ around the origin. This can be useful to study the behaviour of the system near a (stable) equilibrium point. The method can easily be extended to work mod J , for a suitable invariant J related to the initial set X_0 . However, following a separation of concerns principle, we shall confine ourselves here to ‘pure’ Lie derivatives; equivalently, we fix $X_0 = \mathbb{R}^N$.

A general scheme Formally, let us arbitrarily fix a target function g , given as a linear combination of the elements of an ansatz $\mathcal{A} = \{\phi_1, \phi_2, \dots\}$ up to a cutoff M . That is, we fix $v = (\lambda_1, \dots, \lambda_M)^T \in \mathbb{R}^M$ and $\phi \triangleq (\phi_1, \dots, \phi_M)^T$ and consider

$$g = \sum_{i=1}^M \lambda_i \phi_i = v^T \phi. \quad (7)$$

All we require from the functions in \mathcal{A} is that they are analytic⁵ and that the Lie derivative of each ϕ_i can be expressed as a linear combination of the ansatz. That is, for each $i \geq 1$

$$\mathcal{L}(\phi_i) = \sum_{j \geq 0} a_{ij} \phi_j. \quad (8)$$

For the sake of simplicity, we shall assume that, for each i , only finitely many coefficients a_{ij} here are nonzero; this assumption is true e.g. for an ansatz of polynomials (and anyway can be easily lifted). We let $\psi \triangleq (\phi_{M+1}, \dots, \phi_{M+k})^T$, where k is chosen large enough to ensure that, for $1 \leq i \leq M$, we have $a_{ij} = 0$ for each $j > M + k$. We let B be the $M \times k$ matrix of elements $b_{i,j} = a_{i,M+j}$. For any initial condition $x_0 \in \mathbb{R}^N$ of the original system (1), we can form

⁵This can be weakened to analyticity in some open set containing the trajectory $\mathbf{x}(t; x_0)$.

the following finite linear inhomogeneous system of ODE's and initial condition, in the variables $\mathbf{z} = (z_1, \dots, z_M)^T$. Recall that $\psi(t; x_0)$, the composition of ψ with the solution $\mathbf{x}(t; x_0)$ of the original system (1), is an analytic function at the origin.

$$\dot{\mathbf{z}} = A \cdot \mathbf{z} + B \cdot \psi(t; x_0) \quad (9)$$

$$\mathbf{z}(0) = \phi(x_0) \triangleq z_0. \quad (10)$$

The following is an almost immediate consequence of the existence and uniqueness of the solution of ODE's (Picard-Lindelöf).

Theorem 3 (linearization) *Let $x_0 \in \mathbb{R}^N$. Then $\phi(t; x_0)$ is the unique solution of the system (9) with $\mathbf{z}(0)$ as in (10).*

Note that we cannot *explicitly* build the system (9), as the function $\psi(t; x_0)$ is in general not available – even when ψ and B are available. Moreover, the matrix A itself can in practice be too large to be explicitly generated. Indeed (9) is the starting point to build an approximation: essentially, we neglect the “remainder” $\psi(t; x_0)$ and then reduce the resulting linear homogeneous system, by projecting A onto an appropriate subspace of \mathbb{R}^M . We will show that in this construction, none of A , B or ψ need to be explicitly built.

In what follows, we let K_m denote a m -dimensional ($1 \leq m \leq M$; typically $m \ll M$) subspace of \mathbb{R}^M . We let $V = [v_1 | \dots | v_m]$ be an orthonormal basis of K_m , represented as a $M \times m$ matrix. Note that, for $g = v^T \phi$, with $v \in \mathbb{R}^M$, $A^T v$ yields the coefficients of g 's Lie derivative limited to ϕ , as seen from (8). We consider the projection of A^T onto K_m and represent it w.r.t. the basis V , in other words we consider the $m \times m$ matrix

$$H_m \triangleq V^T A^T V. \quad (11)$$

Given a vector of distinct variables $\mathbf{y} = (y_1, \dots, y_m)^T$, $m \geq 1$, we let the *reduced* linear system derived from (9) and the corresponding initial condition, derived from (10), be defined as:

$$\dot{\mathbf{y}} = H_m^T \mathbf{y} \quad (12)$$

$$\mathbf{y}(0) = V^T z_0 \triangleq y_0. \quad (13)$$

Note that the reduced equation system (12) does not depend on x_0 . Let $v \in \mathbb{R}^M$ represent the coefficients of g with respect to ϕ , as in (7). We are interested in studying the approximation

$$g(t; x_0) \approx v^T V \mathbf{y}(t; y_0). \quad (14)$$

This approximation will be studied in a fixed nonempty closed interval $[a, b]$, containing the origin in its interior, and included in an open interval of definition of both $\phi(t; x_0)$ and $\psi(t; x_0)$. The error function, which depends on the coefficients vector v of the target function g in (7), is defined for $t \in [a, b]$ as expected:

$$\epsilon(t; x_0) \triangleq v^T (\phi(t; x_0) - V \mathbf{y}(t; y_0)). \quad (15)$$

Our goal is to devise an easy to compute upper bound on $|\epsilon(t; x_0)|$. In what follows, let us denote by $\|\cdot\|$ an arbitrary matrix norm induced by a vector norm. We let $\rho(x_0) > 0$ such that $\|\phi(t; x_0)\|, \|\psi(t; x_0)\| \leq \rho(x_0)$ for $t \in [a, b]$. Moreover, we let R be the $M \times m$ matrix whose j -th column r_j ($1 \leq j \leq m$) is the projection of $A^T v_j$ onto K_m^\perp : in other words $r_j \triangleq A^T v_j - V V^T A^T v_j$. We also make use of matrix exponential notation $e^L \triangleq \sum_{\ell \geq 0} L^\ell / \ell!$, and denote by e_1^L the first row of the matrix e^L . We first give a general error bound. We defer the discussion on the computation of the involved quantities until we introduce a specific instantiation of the present scheme.

Theorem 4 (general error bound) Assume $v \in K_m$ and that v is orthogonal to v_j for $j \neq 1$. For $t \in [a, b]$, we have

$$|\epsilon(t; x_0)| \leq \rho(x_0) \cdot |v^T v_1| \cdot \int_0^t \left\{ \|e_1^{(t-\tau)H_m^T} V^T B\| + \|e_1^{(t-\tau)H_m^T} R^T\| \right\} d\tau. \quad (16)$$

Krylov spaces and Arnoldi We now describe a convenient way of building the subspace K_m and the matrices V, B, R, H_m needed in the above theorem. For generic $M \times M$ matrix L and vector $u \in \mathbb{R}^M$, the order m Krylov space generated by L and u is $K_m \triangleq \text{span}\{u, Lu, \dots, L^{m-1}u\}$. Here we will assume that m is small enough to satisfy $\dim(K_m) = m$. Note that, as $L^{j+1}u = L(L^j u)$, all that is needed to build K_m is the vector u and access to the matrix-vector multiplication function $u \mapsto Lu$. In fact, there exist effective and numerically stable algorithms that, given a handle to such a function, will build both the orthonormal basis V of K_m and the matrix $H_m = V^T L V$. For example, one has the *Arnoldi algorithm* [19, Ch.6]. The matrix H_m returned by Arnoldi has additional nice properties, in particular it is Hessenberg. Here, we apply the Krylov space construction to $L = A^T$ and $u = v \neq 0$. This choice guarantees that $v \in K_m$ and in fact that $v_1 = v/\|v\|_2$, so that Theorem 4 can be applied.

We now discuss the computational prerequisites of the algorithm and of the error bound in (16) when K_m is to be built with Arnoldi. Concerning the computation of the function $u \mapsto A^T u$, note that, from (8) and by definition of A, B and ψ , we have

$$\mathcal{L}(u^T \phi) = \phi^T A^T u + \psi^T B^T u. \quad (17)$$

This implies that, for each $u \in \mathbb{R}^M$, the vector $A^T u$ can be obtained by taking the Lie derivative of the function $u^T \phi$ and then collecting the nonzero coefficients of the ϕ_i 's in this derivative. As a consequence, the matrix-vector multiplication function $u \mapsto A^T u$ can be computed ‘‘on the fly’’, without building the whole A explicitly. Similarly, (17) shows that one can build $V^T B$ by collecting the nonzero coefficients of the ψ_i 's in the derivatives $\mathcal{L}(v_j^T \phi)$ for $1 \leq j \leq m$. As such derivatives are already computed by the Arnoldi algorithm to build the $A^T v_j$'s, the computation of $V^T B$ comes at no additional cost. Concerning the matrix R , it is easy to see that all of its columns are zero, but possibly the last one, r_m , which can be built again as a by-product of the Arnoldi algorithm: see [19, Ch.6, Sect.6.3]. Finally note that, being m typically quite small and H_m quasi-triangular, the numerical computation of the matrix exponentials $e^{(t-\tau)H_m^T}$, hence of the integral in (16), pose no problem.

The following theorem states an additional property of the approximation scheme when K_m is a Krylov space, saying essentially that if the basis ϕ is rich enough to express the first few derivatives of g , then the approximation is very good at the origin. Moreover, it gives a sufficient condition under which the approximation becomes exact. Recall that a vector space K is *L-invariant* if for each $u \in K$, $Lu \in K$.

Theorem 5 Let K_m be the order m Krylov space generated by A^T and v . Assume the first $M - 1$ rows of B are zero. Then $\epsilon(t; x_0)$ is $O(t^m)$ around the origin. If additionally K_m is A^T -invariant, then $g(t; x_0) = V y(t; y_0)$.

Chebyshev polynomials The above results can be made sharper by instantiating them to a specific ansatz \mathcal{A} , interval $[a, b]$ and norm $\|\cdot\|$. We consider here the ansatz \mathcal{A} of multivariate Chebyshev polynomials [11, Ch.3], $[a, b] \subseteq [-1, 1]$ and the norm $\|\cdot\|_\infty$. A multivariate Chebyshev polynomial is obtained as a product of univariate Chebyshev polynomials $T_i(x_i)$ in the variables x_i , say $\phi_i(x) = T_{i_1}(x_1) \cdots T_{i_N}(x_N)$. Note that it is always possible to reparametrize the original system so as to map any finite interval of interest into $[-1, 1]$. One advantage of considering

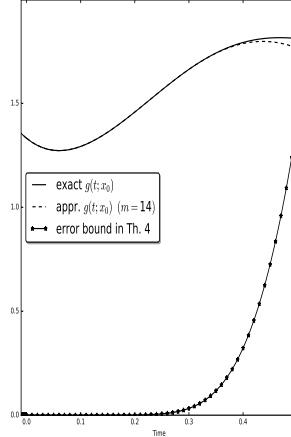


Figure 1: Plots of exact and approximate solutions and of error bound for Example 3.

Chebyshev polynomials is that the magnitude of the coefficients in B , hence $\|B\|_\infty$, decreases exponentially fast with the degree of ψ : more precisely, for any $c > 1$, such coefficients are of magnitude $O(c^{-d})$, where d is the minimal degree of the polynomials in ψ ; see e.g. [11, Th.3.13]. As $\|V^T\|_\infty \leq \sqrt{M}$ (by orthonormality and a general relation between norms ∞ and 2), we have $\|V^T B\|_\infty \leq \sqrt{M} \cdot \|B\|_\infty$: this implies, as seen from (16), that increasing d is beneficial for the global error. Estimation of $\rho(x_0)$ is less straightforward. Assuming $\|x(t; x_0)\|_\infty \leq 1$ for $t \in [a, b]$, by the properties of Chebyshev polynomials [11, Th.3.6] we can set $\rho(x_0) = 1$ in Theorem 4. Note that, in principle, one can always re-scale the original system to make this assumption true. We leave a more detailed treatment of this aspect for future work and end the section with a simple example.

Example 3 Consider the vector field $F = (4x(y - 1), -4x^2 - 4y^2 + 16)^T$. This system has a stable equilibrium (in particular, a focus) at $(\sqrt{3}, 1)$. We consider an initial condition close to this equilibrium, $x_0 = (\sqrt{3} - 2/5, 3/5)^T$, and the time interval $[a, b] = [-1/10, 1/2]$. Figure 1 shows the result of applying the method described above with $m = 14$ to $g(t; x_0) = x(t; x_0)$. The basis of Chebyshev polynomials ϕ has been chosen so as to satisfy the assumption of Theorem 5. For $m = 15$, the approximate plot in the considered interval is nearly indistinguishable from the exact one.

5 Experiments

We report below the outcomes of two experiments we have conducted, applying the algorithm LINABS to two challenging systems taken from the literature. A third example (coupled spring-mass system) is reported in Appendix C. In each case, we compare the results obtained analysing the system \mathcal{S}_0 with unconstrained initial states to those obtained analysing a system \mathcal{S} with certain constraints imposed on the initial states. The execution times reported below are for an implementation in Python 3, running under Windows 10 on a Core i5 machine⁶. A systematic experimentation of the approximate linearization technique discussed in Section 4 is left for future work.

⁶Code and examples available at <http://local.disia.unifi.it/boreale/papers/LinAbs.py>.

Collision avoidance We consider the two-aircraft dynamics used to study collision avoidance, discussed in many papers on hybrid systems [22, 10]. 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; we also introduce the auxiliary variables (parameters, hence 0 derivative) ω_1 and ω_2 , representing the angular velocities of the 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).

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

We 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 first consider $\mathcal{S}_0 = (F, \mathbb{R}^{18})$, that is, we do not impose any constraint on the initial states. We run $\text{LINABS}(\pi, J)$ with $J = \{0\}$, which terminates after $m = 3$ iterations and 15s, yielding a result template π' , formed with 72 polynomial terms. As in this case $\mathbf{V}(J) = X_0$ and J is trivially radical, Corollary 1 implies that these terms form the largest, in terms of spanned space, linear abstraction of \mathcal{S}_0 of degree ≤ 2 . Removing from π' the trivial terms formed purely by constants, leaves 6 nontrivial terms, all of which are in fact constant scale consecutions.

Consider now the system $\mathcal{S} = (F, X_0)$, where $X_0 = \mathbf{V}(\{x_1 - x_{10}, x_2 - x_{20}, \dots\})$, that is, X_0 just equals each variable to the corresponding generic initial value. Using e.g. the method in [7], we can find the invariant ideal J that contains all and only the invariant polynomials of degree ≤ 2 for all $x_0 \in X_0$. In particular, as for each $x_0 \in X_0$ and $p \in J$, $p(t; x_0)|_{t=0} = p(x_0) = 0$ (Lemma 1), we have $X_0 \subseteq \mathbf{V}(J)$. A Gröbner basis of J consists of 12 polynomials and is described in [7]. We run $\text{LINABS}(\pi, J)$, which terminates after $m = 3$ iterations and 20s, yielding a result template π' formed with 76 polynomial terms. As $\mathbf{V}(J) \supseteq X_0$, the first part of Corollary 1 implies that these terms form a linear abstraction of \mathcal{S} . Removing from π' the trivial terms consisting purely of constants, leaves a nontrivial abstraction of 10 terms, so 4 more than in the case of \mathcal{S}_0 . As an example, $p = y_{10}e_2 - y_{20}e_1 + e_{10}y_2 - e_{20}y_1 - y_1e_2 + y_2e_1$ is a constant scale consecution for \mathcal{S} , but not for \mathcal{S}_0 .

Airplanes vertical motion We consider the 6-th order longitudinal equations that capture the vertical motion (climbing, descending) of an airplane [26, 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 $\sin \theta$. 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 [26] (see also [10, 13]); and u_0, w_0, x_0, z_0, q_0 , standing for generic initial values of the corresponding variables. Overall, the system's vector field F_2 consists of 17 polynomials over as many variables.

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

We consider a complete template π of degree 2 over all the original system's variables, plus two auxiliary variables for the monomials qu and qw . Here π is a linear combination of $n = 207$ monomials that uses as many parameters.

We first consider $\mathcal{S}_0 = (F, \mathbb{R}^{17})$, that is, we do not impose any constraint on the initial states. We run $\text{LINABS}(\pi, J)$ with $J = \{0\}$, which terminates after $m = 3$ iterations and 29s, yielding

a result template π' formed with 80 polynomial terms. As in this case $\mathbf{V}(J) = X_0$ and J is trivially radical, Corollary 1 implies that these summands form the largest, in terms of spanned space, linear abstraction of \mathcal{S}_0 of degree ≤ 2 in the given variables. Removing from π' the trivial consecutions, consisting purely of constants or that are multiple of q or θ , leaves 2 nontrivial terms, which are constant scale consecutions.

Consider now the system $\mathcal{S} = (F, X_0)$, where X_0 imposes $\theta = 0$ and equals the remaining variables to the corresponding (generic) initial values. In other words, $X_0 = \mathbf{V}(\{\theta, \sin \theta, \cos \theta - 1, u - u_0, w - w_0, x - x_0, z - z_0, q - q_0\})$. Using e.g. the method in [7], we can find the invariant ideal J that contains all and only the invariant polynomials of degree ≤ 2 for all $x_0 \in X_0$. In particular, just like in the previous example, we have $X_0 \subseteq \mathbf{V}(J)$. A Gröbner basis of J consists of 4 polynomials and is described in [7]. We run `LINABS`(π, J), which terminates after $m = 3$ iterations and about 37s, yielding a result template π' formed with 81 polynomial terms. As $\mathbf{V}(J) \supseteq X_0$, the first part of Corollary 1 implies that these summands form a linear abstraction of \mathcal{S} . The extra term is here $p = z(M/I_{yy}) + \cos \theta(X/m) + \sin \theta(Z/m) + uq \sin \theta - wq \cos \theta$, which is a constant scale consecution for \mathcal{S} , but not for \mathcal{S}_0 .

6 Conclusion and further work

We have proposed a complete method to find all linear abstractions given by polynomial changes of bases that fit a user-specified template. Existence of full abstractions has been characterized. An approximate and computationally cheaper linearization method has also been proposed. A preliminary experimentation, conducted on challenging systems drawn from the literature, has given encouraging results.

As for future work, it would be interesting to extend the present approach to a method for finding (all) Darboux polynomials of a system, as they naturally generalize constant scale consecutions. We note that the invariant ideals returned by the algorithm in [7] already contain all polynomial consecutions (fitting a given template); however, it is not obvious how such consecutions can be extracted from the ideal's Gröbner basis returned by the algorithm.

Acknowledgments

The HSCC'18 reviewers have provided valuable comments.

References

- [1] V.I. Arnold. *Ordinary Differential Equations*. The MIT Press, ISBN 0-262-51018-9, 1978.
- [2] Z. Bai, and D. Skoogh. A projection method for model reduction of bilinear dynamical systems. *Linear Algebra Appl.* 415(2- 3):406-425, 2006.
- [3] U. Baur, P. Benner, and L. Feng. Model Order Reduction for Linear and Nonlinear Systems: A System-Theoretic Perspective. *Arch. Computat. Methods Eng.* 21:331-358, 2014.
- [4] R. Bellman, J.M. Richardson. On some questions arising in the approximate solution of nonlinear differential equations. *Quarterly of Applied Mathematics* 20(4):333-339, January, 1963.

- [5] 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.
- [6] M. Boreale. Algebra, coalgebra, and minimization in polynomial differential equations. In *Proc. of FoSSACS 2017*, LNCS 10203:71-87, Springer, 2017.
- [7] M. Boreale. Complete algorithms for algebraic strongest postconditions and weakest preconditions in polynomial ODE's. *SOFSEM'18*, LNCS 10706:442-455, 2018. Full version available as *CoRR*, *abs/1708.05377*.
- [8] C. Chicone. *Ordinary Differential Equations with Applications*, 2/e. Texts in Applied Mathematics, Springer-Verlag, 2006.
- [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] K. Ghorbal, A. Platzer. Characterizing Algebraic Invariants by Differential Radical Invariants. *TACAS 2014*, LNCS 8413:279-294, 2014.
- [11] A. Gil, J. Segura, and N.M. Temme. *Numerical Methods for Special Functions*. SIAM, 2007.
- [12] E. Goubault, J.-H. Jourdan, S. Putot, S. Sankaranarayanan. Finding non-polynomial positive invariants and lyapunov functions for polynomial systems through Darboux polynomials. *ACC 2014*: 3571-3578, 2014.
- [13] H. Kong, S. Bogomolov, Ch. Schilling, Yu Jiang, Th.A. Henzinger. Safety Verification of Nonlinear Hybrid Systems Based on Invariant Clusters. In *HSCC 2017*:163-172, ACM, 2017.
- [14] K. Kowalski, W.-H. Steeb. *Nonlinear Dynamical Systems and Carleman Linearization*. World Scientific, 1991.
- [15] J.R. Phillips. Projection-based approaches for model reduction of weakly nonlinear time-varying systems. *IEEE Trans. Comput.-Aided Des.*, vol. 22, no. 2, pp. 171-187, 2003.
- [16] A. Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reasoning* 41(2), 143-189, 2008.
- [17] R. Rebiha, A. V. Moura, and N. Matringe. Generating invariants for non-linear hybrid systems. *Theoretical Computer Science*, 594:180-200, 2015.
- [18] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans. Comput.-Aided Des.*, vol. 22, no. 2, pp. 155-170, 2003.
- [19] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [20] S. Sankaranarayanan, H.B. Sipma, and Z. Manna. Non-linear loop invariant generation using Gröbner bases. *POPL 2004*: 318-329, ACM, 2004.
- [21] S. Sankaranarayanan, H.B. Sipma, and Z. Manna. Constructing invariants for hybrid systems. *Formal Methods in System Design* 32(1): 25-55, 2008.

- [22] S. Sankaranarayanan. Automatic invariant generation for hybrid systems using ideal fixed points. *HSCC 2010*: 221-230, 2010.
- [23] S. Sankaranarayanan. Automatic abstraction of non-linear systems using change of bases transformations. *HSCC 2011*: 143-152.
- [24] S. Sankaranarayanan. Change-Of-Bases Abstractions for Non-Linear Systems. *CoRR abs/1204.4347*, 2012.
- [25] A. Sogokon, K. Ghorbal, P.B. Jackson and A. Platzer. A method for invariant generation for polynomial continuous systems. *VMCAI 2016*, LNCS 9583:268-288. Springer, 2016.
- [26] R.F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.
- [27] A. Tiwari, G. Khanna. Nonlinear systems: Approximating reach sets. In *HSCC 2004*:600-614, ACM, 2004.

A Proofs of Section 3

PROOF OF LEMMA 3. We first show that $\pi[V_m]$ is closed under \mathcal{L} and J . Note that $r_i = \mathcal{L}_J^{(i)}(\pi)$ for $i = 0, 1, \dots$. Using the fact that for any π and v we have $\mathcal{L}(\pi[v]) = \mathcal{L}(\pi)[v]$ and Lemma 2 above, it is easy to check that, for each v and $i \geq 0$

$$\mathcal{L}_J^{(i)}(\pi[v]) = (\mathcal{L}_J^{(i)}(\pi))[v] = r_i[v]. \quad (18)$$

Now consider $v \in V_m$ and $p = \pi[v]$. We first show that $r_0[v] = p \bmod J \in \pi[V_m]$. By assumption on v , $r_0[v] \in \pi[\mathbb{R}^M]$, hence $r_0[v] = \pi[w]$, for some $w \in \mathbb{R}^M$. To prove that $\pi[w] \in \pi[V_m]$, it is enough to prove that $w \in V_m$. This is in turn equivalent to $r_0[w], \dots, r_m[w] \in \pi[\mathbb{R}^M]$. Now, for each $i \geq 0$, using the definition of $r_i = \mathcal{L}_J^{(i)}(\pi)$ and (18), we have

$$\begin{aligned} r_i[w] &= (\mathcal{L}_J^{(i)}(\pi))[w] \\ &= \mathcal{L}_J^{(i)}(\pi[w]) \\ &= \mathcal{L}_J^{(i)}(r_0[v]) \\ &= \mathcal{L}_J^{(i)}(r_0)[v] \\ &= r_i[v]. \end{aligned} \quad (19)$$

Since $v \in V_m$, we have that $r_i[w] = r_i[v] \in \pi[\mathbb{R}^M]$ for $i = 0, \dots, m$, which is the wanted claim for this case.

Consider now the Lie derivative modulo J , $\mathcal{L}_J(\pi[v]) = r_1[v]$. We show that $r_1[v] \in \pi[V_m]$. By assumption, $r_1[v] \in \pi[\mathbb{R}^M]$, hence $r_1[v] = \pi[w]$, for some w . We have to prove that $\pi[w] \in \pi[V_m]$, for which it is enough to prove that $w \in V_m$. This is in turn equivalent to $r_0[w], \dots, r_m[w] \in \pi[\mathbb{R}^M]$. We can repeat the reasoning above replacing r_0 with r_1 and the last step (19) with the equality $\mathcal{L}_J^{(i)}(r_1)[v] = r_{i+1}[v]$. Noting that $v \in V_m = V_{m+1}$, we therefore have that $r_{i+1}[v] \in \pi[\mathbb{R}^M]$ for $i = 0, \dots, m$, which is the wanted claim for this case. This proves the second part of the statement.

The first part of the statement, that is $V_m = V_{m+1} = V_{m+2} = \dots$, is now a consequence of the first part. Indeed, consider any $v \in V_m$ and $j \geq 1$. By the closure properties proven above, $r_0[v], r_1[v], \dots, r_{m+j}[v] \in \pi[V_m] \subseteq \pi[\mathbb{R}^M]$. This implies $v \in V_{m+j}$. \square

PROOF OF THEOREM 2. Concerning the first part of the statement, we observe that $\mathbb{R}[\mathbf{x}] \bmod J$ is spanned by P as a vector space (see [9, Ch.5, Sect.3, Prop.1]). Hence, if P is

finite, for each $p \in \mathbb{R}[\mathbf{x}]$, we can write $p = q + r$, where $q \in J$ and $r \in \text{span}(P)$. Then for each $x_0 \in X_0$ we have, around $t = 0$, $p(t; x_0) = q(t; x_0) + r(t; x_0)$. But as $x_0 \in \mathbf{V}(J)$ and $q \in J$, we have that $q(t; x_0)$ is identically 0 (Lemma 1).

Conversely, assume that $\mathbf{V}(J) \subseteq X_0$, that J is a real radical and that P is infinite. Suppose by contradiction there is a full linear abstraction $(p_1, \dots, p_k)^T$; note we can assume w.l.o.g. that $p_i \bmod J = p_i$, otherwise we can consider $(p_1 \bmod J, \dots, p_k \bmod J)^T$, which also forms a full abstraction. Take a monomial $\alpha \in P$ of degree higher than the maximum degree of the p_i 's in the abstraction. Note that $\alpha \bmod J = \alpha$. By definition of full abstraction, there are real coefficients λ_j 's such that for each $x_0 \in \mathbf{V}(J)$, we have that $\alpha(t; x_0) = \sum_{j=1}^k \lambda_j p_j(t; x_0)$, in particular for $t = 0$ this equality implies $\alpha(x_0) = \sum_{j=1}^k \lambda_j p_j(x_0)$. Hence the polynomial $s \triangleq \alpha - \sum_{j=1}^k \lambda_j p_j$ vanishes on $\mathbf{V}(J)$, that is $s \in \mathbf{I}(\mathbf{V}(J)) = J$, where the last equality follows from the fact that J is a real radical. In other words, $s \bmod J = 0$, or equivalently $\alpha = \alpha \bmod J = \sum_{j=1}^k \lambda_j p_j$. But this is a contradiction, as $\deg(\alpha) > \deg(p_i)$ for each i . \square

B Proofs of Section 4

PROOF OF THEOREM 3. Consider any function ϕ_i in the basis ϕ , for $1 \leq i \leq M$. Recalling the definition of $\phi_i(t; x_0)$ and the relation between time and Lie derivatives from Section 2, we have that

$$\begin{aligned} \frac{d}{dt} \phi_i(t; x_0) &= \mathcal{L}(\phi_i)(t; x_0) \\ &= \sum_{j=1}^M a_{ij} \phi_j(t; x_0) + \sum_{j=1}^k b_{ij} \psi_j(t; x_0) \end{aligned}$$

where the last equality stems from (8) and the definition of ϕ, ψ, A and B . But, when considered for all $1 \leq i \leq M$, this is just the equality (9). Moreover, $\phi(t; x_0)|_{t=0} = \phi(x_0)$, so that also (10) is satisfied. Uniqueness in a suitable open interval follows from the Picard-Lindelöf theorem. \square

PROOF OF THEOREM 4. Consider the $m \times 1$ vector function

$$\delta(t; x_0) \triangleq V^T \phi(t; x_0) - \mathbf{y}(t; x_0)$$

which is defined and analytic for t in some open interval including $[a, b]$; moreover, $\delta(0; x_0) = 0$. We have the following equalities for the derivative of $\delta(t; x_0)$, where we abbreviate $\phi(t; x_0), \psi(t; x_0)$ and $\mathbf{y}(t; x_0)$ as ϕ, ψ and \mathbf{y} , respectively.

$$\begin{aligned} \dot{\delta}(t; x_0) &= V^T \dot{\phi} - \dot{\mathbf{y}} \\ &= V^T (A\phi + B\psi) - H_m^T \mathbf{y} \end{aligned} \tag{20}$$

$$\begin{aligned} &= (\phi^T A^T V)^T + V^T B\psi - H_m^T \mathbf{y} \\ &= (\phi^T V H_m + \phi^T R)^T + V^T B\psi - H_m^T \mathbf{y} \end{aligned} \tag{21}$$

$$\begin{aligned} &= H_m^T V^T \phi + R^T \mathbf{y} + V^T B\psi - H_m^T \mathbf{y} \\ &= H_m^T \delta(t; x_0) + V^T B\psi + R^T \phi \end{aligned} \tag{22}$$

where: (20) follows from Theorem 3 and (12); (21) follows from $A^T V = V V^T A^T V + R$ and the definition of H_m (11); (22) follows from the definition of $\delta(t; x_0)$. We have therefore obtained a linear inhomogeneous equation for $\delta(t; x_0)$, that is $\dot{\delta}(t; x_0) = H_m^T \delta(t; x_0) + V^T B\psi(t; x_0) + R^T \phi(t; x_0)$, together with the initial condition $\delta(0; x_0) = 0$. The unique solution of the resulting initial value problem satisfies [8, Prop.2.67]

$$\delta(t; x_0) = \int_0^t e^{(t-\tau)H_m^T} (V^T B\psi(\tau; x_0) + R^T \phi(\tau; x_0)) \, d\tau. \tag{23}$$

As $\delta(t; x_0)$ is analytic, equation (23) is valid in the whole open interval of definition of $\delta(t; x_0)$, which includes $[a, b]$. As by hypothesis $v \in K_m$ and v is orthogonal to v_j for $j > 1$, we can write $v^T = v^T V V^T$. Therefore, denoting by δ_1 the first component of δ , we have:

$$\begin{aligned}
\epsilon(t; x_0) &= v^T \phi(t; x_0) - v^T V \mathbf{y}(t; x_0) \\
&= v^T V V^T \phi(t; x_0) - v^T V \mathbf{y}(t; x_0) \\
&= v^T V \delta(t; x_0) \\
&= v^T v_1 \cdot \delta_1(t; x_0) \\
&= v^T v_1 \cdot \int_0^t e_1(t-\tau) H_m^T (V^T B \psi(\tau; x_0) + R^T \phi(\tau; x_0)) \, d\tau.
\end{aligned}$$

As a consequence, we have, by applying basic properties of norms and integrals, and taking into account how $\rho(x_0) > 0$ has been chosen:

$$\begin{aligned}
|\epsilon(t; x_0)| &= |v^T v_1| \cdot \left| \int_0^t e_1^{(t-\tau)H_m^T} (V^T B \psi(\tau; x_0) + R^T \phi(\tau; x_0)) \, d\tau \right| \\
&\leq |v^T v_1| \cdot \int_0^t \left| e_1^{(t-\tau)H_m^T} (V^T B \psi(\tau; x_0) + R^T \phi(\tau; x_0)) \right| \, d\tau \\
&\leq |v^T v_1| \cdot \int_0^t \left\{ |e_1^{(t-\tau)H_m^T} V^T B \psi(\tau; x_0)| + |e_1^{(t-\tau)H_m^T} R^T \phi(\tau; x_0)| \right\} \, d\tau \\
&\leq |v^T v_1| \cdot \int_0^t \left\{ \|e_1^{(t-\tau)H_m^T} V^T B\| \cdot \|\psi(\tau; x_0)\| + \|e_1^{(t-\tau)H_m^T} R^T\| \cdot \|\phi(\tau; x_0)\| \right\} \, d\tau \\
&\leq \rho(x_0) \cdot |v^T v_1| \cdot \int_0^t \left\{ \|e_1^{(t-\tau)H_m^T} V^T B\| + \|e_1^{(t-\tau)H_m^T} R^T\| \right\} \, d\tau.
\end{aligned}$$

This is the wanted inequality, which holds for t in the interval of definition of $\delta(t; x_0)$, including $[a, b]$. \square

PROOF OF THEOREM 5. Under the given hypotheses, it is easy to see, using (8), that $\frac{d^{(j)}}{dt^j} g(t; x_0)|_{t=0} = \mathcal{L}^{(j)}(g)(x_0) = v^T A^j \phi(x_0) = v^T A^j z_0 = z_0^T (A^T)^j v$, for $0 \leq j < m$. The last term, using the decomposition $z_0 = V V^T z_0 + w_0$ for some $w_0 \in K_m^\perp$, can be written as $(V V^T z_0)^T (A^T)^j v$, because $(A^T)^j v \in K_m$ hence $w_0 \perp (A^T)^j v$. Now, using the definition (11) of H_m and orthonormality of V , we can prove that $(A^T)^j v = V H_m^j V^T v$ for $0 \leq j < m$. Using this equation, we get $(V V^T z_0)^T (A^T)^j v = (V V^T z_0)^T V H_m^j V^T v$. Taking the transpose, the last term equals $v^T V (H_m^T)^j V^T V V^T z_0 = v^T V (H_m^T)^j V^T z_0 = v^T V (H_m^T)^j y_0$. But the last term is seen to be, from (12), the j -th derivative at $t = 0$ of $v^T V \mathbf{y}(t; x_0)$. Therefore the first m terms in the Taylor expansion from $t = 0$ of $g(t; x_0)$ and of $v^T V \mathbf{y}(t; x_0)$ do coincide, which means that the difference between the two is $O(t^m)$.

If additionally K_m is A^T -invariant, then we can extend the equality $(A^T)^j v = V H_m^j V^T v$, hence the subsequent reasoning, to all $j \geq 0$, thus proving that $g(t; x_0) = v^T V \mathbf{y}(t; x_0)$. \square

C Experiments with coupled spring-mass system

We consider a coupled spring-mass system, consisting of two bodies of masses m_1 and m_2 that are connected to each other and to a fixed end using springs of constants k_1 and k_2 , respectively. See e.g. [22]. The system described by the differential equations below, where the variables x_1, x_2, v_1, v_2 represent the two bodies' positions and velocities; moreover, we represent the constants $k_1, k_2, m_1, m_2, 1/m_1, 1/m_2$ and the generic initial values of the positions and velocities, $x_{10}, x_{20}, v_{10}, v_{20}$, as constant (0 derivative) functions. Overall, the system's vector field

F_3 consists of 14 polynomials over as many variables.

$$\begin{aligned}\dot{x}_1 &= v_1 & \dot{v}_1 &= -(1/m_1)(k_1x_1 - k_2(x_1 - x_2)) \\ \dot{x}_2 &= v_2 & \dot{v}_2 &= (k_2/m_2)(x_1 - x_2)\end{aligned}$$

We consider a complete template π of degree 3 over all the variables: π is a linear combination of $n = 680$ monomials that uses as many parameters.

We first consider $\mathcal{S}_0 = (F, \mathbb{R}^{14})$, that is, we do not impose any constraint on the initial states. We run $\text{LINABS}(\pi, J)$ with $J = \{0\}$, which terminates after $m = 4$ iterations and 182.6s, yielding a result template π' formed with 286 polynomial terms, all of which are trivial, i.e. formed purely by constants. As in this case $\mathbf{V}(J) = X_0$ and J is trivially radical, Corollary 1 implies that these are all the linear abstractions for \mathcal{S}_0 of degree ≤ 3 .

Consider now the system $\mathcal{S} = (F, X_0)$, where X_0 imposes $m_i \cdot (1/m_i) = 1$ for $i = 1, 2$ and assigns (generic) initial values to the remaining variables. In other words, $X_0 = \mathbf{V}(\{x_1 - x_{10}, x_2 - x_{20}, v_1 - v_{10}, v_2 - v_{20}, m_1 \cdot (1/m_1) - 1, m_2 \cdot (1/m_2) - 1\})$. Using e.g. the method in [7], we can find the invariant ideal J that contains all and only the invariant polynomials of degree ≤ 3 for all $x_0 \in X_0$; a Gröbner basis G of J consists of 3 polynomials, and is reported below. We then run $\text{LINABS}(\pi, J)$, which terminates after $m = 6$ iterations and 264s, yielding a result template π' formed with 295 polynomial terms, of which the only nontrivial one is $p = k_1x_1^2 + k_2x_1^2 - 2k_2x_1x_2 + k_2x_2^2 + m_1v_1^2 + m_2v_2^2$. As in this case $\mathbf{V}(J) \supseteq X_0$, the first part of Corollary 1 implies that p is a linear abstraction of \mathcal{S} , indeed a constant scale consecution.

For the sake of completeness, a Gröbner basis G of J is reported below. Lexicographic ordering is assumed.

$$\begin{aligned}G &= \{x_{10}^2k_1 + x_{10}^2k_2 - 2x_{10}x_{20}k_2 + x_{20}^2k_2 + m_1v_{10}^2 \\ &\quad + m_2v_{20}^2 - k_1x_1^2 - k_2x_1^2 + 2k_2x_1x_2 - k_2x_2^2 \\ &\quad - m_1v_1^2 - m_2v_2^2, m_1(1/m_1) - 1, m_2(1/m_2) - 1\}.\end{aligned}$$

