Nonlinear Systems: Approximating Reach Sets*

Ashish Tiwari[1] and Gaurav Khanna[2,3]

[1] SRI International, 333 Ravenswood Ave, Menlo Park, CA, U.S.A
Tel:+1.650.859.4774, Fax:+1.650.859.2844, tiwari@csl.sri.com
[2] Theoretical and Computational Studies Group
Long Island University, Southampton NY 11968, gkhanna@liu.edu
[3] Department of Physics, U. of Massachusetts, Dartmouth, MA 02747

Abstract. We describe techniques to generate useful reachability information for nonlinear dynamical systems. These techniques can be automated for polynomial systems using algorithms from computational algebraic geometry. The generated information can be incorporated into other approaches for doing reachability computation. It can also be used when abstracting hybrid systems that contain modes with nonlinear dynamics. These techniques are most naturally embedded in the hybrid qualitative abstraction approach proposed by the authors previously. They also show that the formal qualitative abstraction approach is well suited for dealing with nonlinear systems.

1 Introduction

Computing the set of states reachable from the initial states is central to the problem of proving that a system is safe, that is, it does not enter a "bad" region. Exact reachability set computation is, however, elusive for both discrete transition systems and continuous dynamical systems. Hybrid systems combine these two formalisms and inherit their complexities. For purposes of proving safety, though, the exact reachability set is not required and suitable over approximations of this set suffice. The approaches for computation of (over approximations of) the reachability set can be broadly be classified into two categories: (i) methods based on explicit computation of the reachability set by forward simulating the system and widening [7, 13, 6], and (ii) methods based on abstraction [1, 23]. In this paper, we present analytical results on computing over approximations of the set of reachable states, which can be integrated with both these methods. Despite their generality, these results are best motivated in the context of our hybrid qualitative abstraction approach.

Our approach to analyzing hybrid system is based on constructing sound discrete abstractions of the hybrid system [23]. The abstraction methodology combines predicate abstraction technique [10], for abstracting the discrete component of the hybrid system, and qualitative abstraction [23, 12], for handling the continuous dynamics.

* Research of the first author was supported in part by the National Science Foundation under grant CCR-0311348, and under NASA Langley Research Center contract NAS1-00108 to Rannoch Corporation.

The effectiveness of our abstraction method, as with all approaches based on abstraction, depends on the choice of abstraction predicates. The discrete component of a hybrid system is "simpler", in this respect, than the continuous component. This is because the guard conditions, state invariants, and reset assignments immediately suggest what predicates are important for the discrete logic. And in most real world problems, the discrete logic is simple enough that this choice is adequate. The choice of predicates for constructing good (qualitative) abstractions of the continuous components is not always so obvious. When we first described the hybrid abstraction algorithm [23], we proposed the use of first, second, and higher-order derivatives of the expressions that occur in the guards, property, and initialization. For example, if the dynamics is given by $\dot{x} = 100 - x$, and interest is in the value of x, then clearly $100 - x$ is a good candidate expression to monitor. But this does not work always. For example, in the two-dimensional linear system $\dot{x} = -2x + y$, $\dot{y} = x - 2y$, say initially $x = 5$ and $y = 0$, and we are interested in proving that $y \leq 5$ always, given the state invariant $x \geq 0$. The (higher-order) derivatives of $y - 5$ or x do not help much. However, if we look at the function $x + y$, then we immediately notice that $\dot{(x + y)} = -(x + y)$ and we can (just using qualitative reasoning) get to the desired conclusion.

The above observation suggests that concepts, such as Lyapunov functions, from linear and nonlinear systems theory can yield useful reachability information too. In a previous article [22], we considered linear systems and suggested the use of left eigenvectors for generating useful functions for the process of qualitative abstraction. In this paper, we present some initial results for nonlinear systems.

In the case of linear systems, using simple linear algebra, we can always get *linear* functions $V : \Re^n \mapsto \Re$ that are either (i) exponentially changing ($\dot{V} = \lambda V$), (ii) oscillating ($\ddot{V} = \lambda V$), or (iii) oscillating with exponentially decaying amplitude ($\ddot{V} = a\dot{V} + bV$.) The *exact* reach set is computable (in some cases) when there are enough (different) functions V of the first kind [16, 15]. In [22], we showed that even when there do not exist sufficiently many such V's, useful *partial* reachability information can be obtained.

The functions V can be interpreted as representing the total "energy" of the system (though V can be increasing in our case). In this paper, we are interested in computing linear and nonlinear functions V for nonlinear systems. Such functions are used in the process of abstracting the continuous dynamics inside a mode of a hybrid system. Whenever possible, we shall explicitly show the resulting over approximation of the reach set. We also present methods for computing these functions whenever the function itself is a polynomial and the nonlinear system too is described using polynomials.

Unlike linear systems, there are no standard concepts, like eigenvectors, that can be used for nonlinear systems. The functions V satisfying (i), (ii), or (iii) need not be linear, in fact they need not even be polynomials. But we introduce some new concepts such as *exact-ideal*, a subset of a polynomial ideal, to compute polynomial V's whenever they exist.

1.1 Preliminaries

A (nonlinear) dynamical system S consists of a finite set $x_1, x_2, \ldots, x_n$ of real valued variables, a set of differential equations $dx_1/dt = p_1(x_1, x_2, \ldots, x_n)$, $dx_2/dt = p_2(x_1, x_2, \ldots, x_n), \ldots, dx_n/dt = p_n(x_1, x_2, \ldots, x_n)$, a set *Init* $\subseteq \Re^n$ of initial states, and a set *Inv* $\subseteq \Re^n$ of the invariant region. We use matrix notation to represent the dynamics. The n variables are represented as a $n \times 1$ column vector $\boldsymbol{x}$ and the $n \times 1$ column vector consisting of the functions $p_1, p_2, \ldots, p_n$ is called a *vector field* and is denoted by $\boldsymbol{p}$. In short, the dynamics are written as $\dot{\boldsymbol{x}} = \boldsymbol{p}$. We often assume that the p_i's are polynomials over the n variables $x_1, x_2, \ldots, x_n$.

The invariant region *Inv* is specified as a formula ϕ with free variables $x_1, \ldots, x_n$ in the theory of reals. We assume no particular representation for the set of initial states *Init*. The theory of reals, and the set of all reals, are both denoted by $\Re$ – the intention is disambiguated by the context. The notation $\Re \vdash \psi$ means that the formula ψ (universally quantified) is valid in the theory $\Re$, that is, it is true for all real valuations of the variables. For example, $\Re \vdash (x_1^2 + x_2^2 \geq 0)$.

The semantics $[[S]]$ of a dynamical system S, with dynamics $\dot{\boldsymbol{x}} = \boldsymbol{p}$, initial states *Init* and invariant *Inv*, over an interval $I = [t_0, t_1] \subseteq \mathbb{R}$ is a collection of mappings $\boldsymbol{x} : I \mapsto \mathbf{X}$ satisfying (i) the initial condition: $\boldsymbol{x}(t_0) \in$ *Init*, (ii) the continuous dynamics: for all $t \in [t_0, t_1]$, $\dot{\boldsymbol{x}}(t) = \boldsymbol{p}(t)$, and (iii) the invariant: for all $t \in (t_0, t_1)$, $\boldsymbol{x}(t) \in$ *Inv*. In case the interval I is left unspecified, it is assumed to be the interval $[0, \infty)$. The motivation for the invariant region *Inv* comes from hybrid systems and informally, the semantics is given such that only those trajectories of the dynamical system are valid which do not take the system out of the set *Inv*. We say that a state $\boldsymbol{s} \in \Re^n$ is *reachable* in the system S if there exists a function $\boldsymbol{x} \in [[S]]$ such that $\boldsymbol{s} = \boldsymbol{x}(t)$ for some $t \in I$. The reach set, *Reach*(S), is defined as the set of all reachable states of the system S.

We are interested in smooth functions $V : \Re^n \mapsto \Re$ satisfying certain nice properties. A $1 \times n$ row vector of such functions will be called a **1-form**. If V is a function, then the notation $\boldsymbol{d}V$ denotes the 1-form consisting of partial derivatives of V with respect to the n variables, that is, $\boldsymbol{d}V = [\partial V/\partial x_1, \partial V/\partial x_2, \ldots, \partial V/\partial x_n]$. In matrix notation, a 1-form is denoted by $\boldsymbol{q}^T$. We use some differential geometry terminology in this paper, but it is always backed up with detailed expansions, and hence the presentation may appear verbose to an expert reader.

2 Linear Invariants

We consider (time invariant) polynomial nonlinear dynamical systems, that is, in the dynamics $\dot{\boldsymbol{x}} = \boldsymbol{p}$, each component of the vector field $\boldsymbol{p}$ is specified by a (possibly nonlinear) polynomial $p_i(x_1, \ldots, x_n)$ over the variables $x_1, \ldots, x_n$ in $\boldsymbol{x}$. We will separate out the nonlinear component from the linear component and represent such a dynamical system as

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{y}$$

where $\boldsymbol{y}$ is the vector of non-linear power-products of state variables in $\boldsymbol{x}$. Here A is an $n \times n$ matrix, B is an $n \times m$ matrix, $\boldsymbol{x}$ is a $n \times 1$ vector, and $\boldsymbol{y}$ is a $m \times 1$ vector. In the case of linear systems, $m = 0$. Example 1 gives a simple illustration of this notation.

Let $\boldsymbol{c}$ be a real eigenvector of A^T which is also in the *kernel* of B^T (that is, the linear subspace of zeros of B^T), that is,

$$A^T \boldsymbol{c} = \lambda \boldsymbol{c} \qquad B^T \boldsymbol{c} = \boldsymbol{0},$$

where the components of $\boldsymbol{c}$ are reals.

The transpose $\boldsymbol{c}^T$ of the vector $\boldsymbol{c}$ is a 1-form. Consider the linear function $V = \boldsymbol{c}^T \boldsymbol{x}$.

$$\dot{V} = \boldsymbol{c}^T \dot{\boldsymbol{x}} = \boldsymbol{c}^T (A\boldsymbol{x} + B\boldsymbol{y}) = (A^T \boldsymbol{c})^T \boldsymbol{x} + (B^T \boldsymbol{c})^T \boldsymbol{y} = (\lambda \boldsymbol{c})^T \boldsymbol{x} + \boldsymbol{0} = \lambda V.$$

The value of the function V will either (i) monotonically increase or decrease while remaining sign-invariant (if $\lambda > 0$), or (ii) asymptotically converge to 0 (if $\lambda < 0$), or (iii) remain constant (if $\lambda = 0$). Thus, $\boldsymbol{c}^T \boldsymbol{x}$ can be used to generate useful invariants of the dynamical system and give bounds on the reach sets of such systems as summarized in the following theorem.

Theorem 1. *Let $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{y}$ be a nonlinear dynamical system with initial states Init. Let λ be a real eigenvalue of the matrix A^T, $\boldsymbol{c} = [c_1, c_2, \ldots, c_n]^T$ be a corresponding eigenvector (of A^T), and $V = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = \boldsymbol{c}^T \boldsymbol{x}$ be the corresponding linear function. Suppose that $\boldsymbol{c}$ is also in the kernel of B^T.*

If d_{min} and d_{max} denote, respectively, the minimum and maximum values in the set $\{\boldsymbol{c}^T \boldsymbol{x}(0) : \boldsymbol{x}(0) \in Init\}$, then the formula ϕ, as defined below, is an over approximation of the reach set of the nonlinear system:
(i) Case $\lambda > 0$: if $d_{min} > 0$, then ϕ is $V \geq d_{min}$, if $d_{max} < 0$, then ϕ is $V \leq d_{max}$, and if $d_{min} = d_{max} = 0$, then ϕ is $V = 0$;
(ii) Case $\lambda < 0$: ϕ is defined as $min\{0, d_{min}\} \leq V \wedge V \leq max\{0, d_{max}\}$; and
(iii) Case $\lambda = 0$: ϕ is defined as $V = V(0)$.

We illustrate the above technique from an example from the sporulation initiation network of the bacteria *B.Subtilis* [24].

Example 1. The proteins (*SinI*) and (*SinR*) are known to be crucial in the decision for committing to sporulation. High concentration of (*SinR*) inhibits sporulation. Under conditions of stress, the production of (*SinI*) increases, (*SinI*) binds with (*SinR*), thus reducing the concentration of free (*SinR*), and thus promoting sporulation. The dynamics of these two protein concentrations is given by:

$$\begin{aligned} (\dot{SinI}) &= \mathit{\Delta}_{(SinI)} - \lambda(SinI) - k(SinI)(SinR) \\ (\dot{SinR}) &= \mathit{\Delta}_{(SinR)} - \lambda(SinR) - k(SinI)(SinR) \end{aligned}$$

where $\mathit{\Delta}_{(SinI)}$ and $\mathit{\Delta}_{(SinR)}$ are determined by the corresponding transcription and translation rates and k is the rate of reaction that binds (*SinI*) to (*SinR*).

In a *hybrid* model of the network, the dynamics of the values of $\Delta_{(SinI)}$ and $\Delta_{(SinR)}$ are captured through *discrete* mode transitions. In any given mode, the values of $\Delta_{(SinI)}$ and $\Delta_{(SinR)}$ can be assumed constants.

In matrix notation, let $\boldsymbol{x} = [(SinI), (SinR), z]^T$, and let $\boldsymbol{y} = [(SinI)(SinR)]$, we have $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{y}$, where

$$A = \begin{bmatrix} -\lambda & 0 & \Delta_{(SinI)} \\ 0 & -\lambda & \Delta_{(SinR)} \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} -k \\ -k \\ 0 \end{bmatrix}$$

Consider $\boldsymbol{c} = [-\lambda, \lambda, \Delta_{(SinI)} - \Delta_{(SinR)}]^T$. This vector is an eigenvector of A^T with corresponding eigenvalue $-\lambda$ and it is also in the kernel of B^T.

Define $V = \boldsymbol{c}^T\boldsymbol{x}$. If the cell is stressed and $\Delta_{(SinI)} = \Delta_{(SinR)}$, and the cell goes into the state where $(SinI) \geq (SinR)$, then (in all subsequently reachable states) it will always be the case that $(SinI) \geq (SinR)$. Thus $(SinI) \geq (SinR)$ would be a stable region.

The above method can be implemented since it only involves computing eigenvectors and testing if an eigenvector is in the kernel of another matrix. This can be efficiently done when the eigenvalue λ is a rational (as in the above example). If not, then we will need to represent and compute with algebraic numbers.

This method can be effectively used on most of the hybrid models that result from modeling of genetic regulatory networks. However, for other more general classes of systems, it is not quite as effective, since it can only generate *linear* functions V. We next discuss approaches to discover nonlinear functions V.

3 Polynomial Invariants

We are interested in polynomial invariants of nonlinear systems. We generate such invariants using various computational techniques from the field of algebraic geometry, most notably Gröbner bases and Syzygy bases computations [5], and the Frobenius theorem [27].

Consider the dynamics given by $\dot{\boldsymbol{x}} = \boldsymbol{p}$, where $\boldsymbol{p}$ is a vector field. The syzygies, *Syz*, of the vector field $\boldsymbol{p}$ is defined as the set of all 1-forms $\boldsymbol{h}^T$ s.t. $\boldsymbol{h}^T\boldsymbol{p} = 0$:

$$Syz(\boldsymbol{p}) := \{\boldsymbol{h}^T : \boldsymbol{h}^T\boldsymbol{p} = 0\}$$

A 1-form $\boldsymbol{h}^T$ is *exact* if there exists a smooth function (polynomial, in our case) V such that $\boldsymbol{h}^T = \boldsymbol{d}V$.

Suppose there is a syzygy $\boldsymbol{q}^T$ of the vector field $\boldsymbol{p}$ which is also exact. Consequently, there is a polynomial V such that

$$\partial V/\partial x_1 = q_1,\ \partial V/\partial x_2 = q_2,\ \ldots,\ \partial V/\partial x_n = q_n \text{ and}$$
$$q_1p_1 + q_2p_2 + \cdots + q_np_n = 0$$

Under these assumptions, it is easy to note that the Lie derivative of V with respect to the vector field $\boldsymbol{p}$ vanishes, that is,

$$\begin{aligned}\frac{dV}{dt} = \boldsymbol{dV}\boldsymbol{p} &= \frac{\partial V}{\partial x_1}\frac{dx_1}{dt} + \frac{\partial V}{\partial x_2}\frac{dx_2}{dt} + \cdots + \frac{\partial V}{\partial x_n}\frac{dx_n}{dt} \\ &= q_1 p_1 + q_2 p_2 + \cdots + q_n p_n = 0\end{aligned}$$

Hence, the value of the expression $V(x_1, x_2, \ldots, x_n)$ remains invariant through the time evolution of the nonlinear system.

Theorem 2. *Let $\dot{\boldsymbol{x}} = \boldsymbol{p}$ be a nonlinear dynamical system. Suppose the 1-form $\boldsymbol{h}^T$ is a syzygy of $\boldsymbol{p}$ and is exact such that $\boldsymbol{h}^T = \boldsymbol{dV}$.*

If $\boldsymbol{x}(0)$ is some initial state, then the formula $V = V(\boldsymbol{x}(0))$ denotes an over approximation of the set of states reachable from $\boldsymbol{x}(0)$.

Given a polynomial vector field $\boldsymbol{p}$, the set of generators for the set $Syz(\boldsymbol{p})$ is computable using well-known techniques from computational algebraic geometry. Frobenius theorem can be used to check if a given syzygy is exact.

Example 2. Consider the nonlinear dynamical system:

$$\dot{x_1} = x_1 x_2 \qquad \dot{x_2} = -x_1$$

It is the case that $1x_1x_2 + x_2(-x_1) = 0$ and hence $(1, x_2)$ is a syzygy of the polynomials $x_1x_2, -x_1$. A solution for V that satisfies both $\partial V/\partial x_1 = 1$ and $\partial V/\partial x_2 = x_2$, is $V = x_1 + x_2^2/2$. It is easily observed that $\dot{V} = 0$ and hence V is an invariant of the above dynamical system.

Example 3. Consider the rotational motion of a rigid body in three-dimensional space. In the absence of external torques, the motion can be described by

$$\dot{x_1} = ax_2x_3 \qquad \dot{x_2} = -bx_1x_3 \qquad \dot{x_3} = cx_1x_2$$

in a suitably chosen coordinate axes. A syzygy for the vector field $\boldsymbol{p} = [ax_2x_2, -bx_1x_3, cx_1x_2]^T$ is $[a'x_1, b'x_2, c'x_3]$ whenever $a'a - b'b + c'c = 0$. Thus, we are constrained to find an V such that $\partial V/\partial x_1 = a'x_1$, $\partial V/\partial x_2 = b'x_2$, and $\partial V/\partial x_3 = c'x_3$. By Frobenius theorem, we know these constraints are satisfiable and indeed $V = a'x_1^2/2 + b'x_2^2/2 + c'x_3^2/2$ is the desired invariant function.

Details of the computability issues are postponed to a later section. We move on to more general forms of polynomial invariants.

3.1 Exponentially Changing Functions

A set of functions that we have successfully used for computing approximate reachability for linear systems are polynomials V such that $dV/dt = \lambda V$, for some real constant λ. In the case of linear systems, it is known that the exact reachability set is *computable* whenever n such linear functions exist [15]. In [22], we showed that approximate reachability sets can be computed in the case when fewer than n such functions exist. Computation of such (linear) functions for linear systems reduces to eigenvector computation.

For a system with dynamics $\dot{\boldsymbol{x}} = \boldsymbol{p}$, it is the case that

$$\frac{dV}{dt} = \lambda V \qquad \text{iff} \qquad \boldsymbol{d}V\boldsymbol{p} = \lambda V$$

Define the ideal, *Ideal*($\boldsymbol{p}$), generated by a polynomial vector field $\boldsymbol{p}$ as

$$\mathit{Ideal}(\boldsymbol{p}) := \{r : r = \boldsymbol{q}^T\boldsymbol{p}, \boldsymbol{q}^T \text{ is any polynomial 1-form}\}$$

The statement $\boldsymbol{d}V\boldsymbol{p} = \lambda V$ is equivalent to saying that (i) $V \in \mathit{Ideal}(\boldsymbol{p})$, that is, there exists a polynomial 1-form $\boldsymbol{q}^T$ such that $V = \boldsymbol{q}^T\boldsymbol{p}$ and (ii) $\boldsymbol{d}V = \lambda^{-1}\boldsymbol{q}^T$.

Ideal membership is efficiently decided by Gröbner basis computation [5]. The details of computability are relegated to a later section. If we have computed a polynomial V and constant λ satisfying the above two conditions, then we immediately get the following upper-bound on the set of reachable states of the nonlinear system.

Theorem 3. *Let $\dot{\boldsymbol{x}} = \boldsymbol{p}$ be a nonlinear dynamical system and Init denote the set of initial states of the system. Suppose V is a (nonlinear) polynomial such that*

- *$V \in \mathit{Ideal}(\boldsymbol{p})$, that is, $V = \boldsymbol{q}^T\boldsymbol{p}$, and*
- *$\boldsymbol{d}V = \lambda^{-1}\boldsymbol{q}^T$.*

If d_{min} and d_{max} denote, respectively, the minimum and maximum values in the set $\{V(\boldsymbol{x}(0)) : \boldsymbol{x}(0) \in \mathit{Init}\}$, then the formula ϕ, as defined below, is always an over approximation of the set of reachable states of the system:
(i) Case $\lambda > 0$: if $d_{min} > 0$, then ϕ is $V \geq d_{min}$, if $d_{max} < 0$, then ϕ is $V \leq d_{max}$, and if $d_{min} = d_{max} = 0$, then ϕ is $V = 0$;
(ii) Case $\lambda < 0$: ϕ is $min\{0, d_{min}\} \leq V \ \wedge \ V \leq max\{0, d_{max}\}$; and
(iii) Case $\lambda = 0$: ϕ is $V = V(\boldsymbol{x}(0))$.

Proof. The Lie derivative of the function V with respect to the vector field $\boldsymbol{p}$ is λV, and hence $V(\boldsymbol{x}(t)) = V(\boldsymbol{x}(0))e^{\lambda t}$. The conclusions follow.

We note here that for linear systems, the 1-form $\boldsymbol{q}^T$ can be chosen to be the left eigenvector of the A matrix [22] and hence such $\boldsymbol{q}^T$'s can be easily computed. For nonlinear systems, $\boldsymbol{q}^T$ can be seen as a suitable generalization of the concept of an (left) eigenvector, but computing such an $\boldsymbol{q}^T$ is not so simple, see Section 4.

3.2 Nondecreasing Functions

Theorem 3 makes strong assumptions which restrict its applicability. We weaken the conditions by noticing that λ need not be a constant, it can be any nonpositive or nonnegative definite function. We say that a function r (from $\Re^n$ to $\Re$)

is *nonnegative definite relative to* ψ if it is the case that the formula ψ implies $r \geq 0$ in the theory of reals, that is, $\Re \vdash \psi \Rightarrow r \geq 0$. The formula ψ would represent the state invariant of the mode of the hybrid system whose dynamics are being studied.

A minor variant of the method of Section 3.1 considers functions V such that the Lie derivative of V with respect to the vector field $\boldsymbol{p}$ of the nonlinear system is (relatively) nonnegative definite. In other words, the function V satisfies the equation

$$\frac{dV}{dt} = r$$

where the polynomial r is nonnegative definite (relative to the state invariant). Note that even when no state invariant is given (that is, ψ is just *True*), we can still have nontrivial p's which are nonnegative definite. In particular, these will be sums of squares of polynomials.

Theorem 4. *Let $\dot{\boldsymbol{x}} = \boldsymbol{p}$ be a nonlinear dynamical system with initial states Init and state invariant ψ. If r is nonnegative definite relative to ψ and such that*

- *$r \in Ideal(\boldsymbol{p})$, that is, $r = \boldsymbol{q}^T\boldsymbol{p}$, and*
- *the 1-form $\boldsymbol{q}^T$ is exact, that is, $\boldsymbol{d}V = \boldsymbol{q}^T$,*

then the formula $V \geq d_{min}$ is an over approximation of the set of reachable states of the system, where $d_{min} = min\{V(\boldsymbol{x}(0)) : \boldsymbol{x}(0) \in Init\}$.

We can weaken the conditions of Theorem 4 and require the polynomial function r to be nonnegative definite relative to either $\psi \wedge V > 0$ or $\psi \wedge V < 0$. The conclusion is correspondingly weakened.

Corollary 1. *Let $\dot{\boldsymbol{x}} = \boldsymbol{p}$ be a nonlinear dynamical system with initial states Init and state invariant ψ. Let r and V be such that*

- *$r = \boldsymbol{d}V\boldsymbol{p}$ and*
- *r is nonnegative definite relative to $\psi \wedge V > 0$ (alternatively, relative to $\psi \wedge V < 0$).*

Define $d_{min} = min\{V(\boldsymbol{x}(0)) : \boldsymbol{x}(0) \in Init\}$. If $d_{min} > 0$ (alternatively, $d_{min} < 0$), then the formula $V \geq d_{min}$ is an over approximation of the set of reachable states of the system.

Proof. If V and r are defined as above, then $\dot{V} = \boldsymbol{d}V\boldsymbol{p} = r$ and $\Re \vdash \psi \wedge V > 0 \Rightarrow r > 0$. If $d_{min} > 0$, then in all initial states, $r \geq 0$, and hence the value of V is always nondecreasing, and hence $V \geq d_{min}$.

Alternatively, consider the case when $\Re \vdash \psi \wedge V < 0 \Rightarrow r \geq 0$. In this case, whenever the value of the function V drops below zero, its derivative r becomes nonnegative. In particular, when $V = d_{min} < 0$, then $\dot{V} \geq 0$, and hence $V \geq d_{min}$ always.

We illustrate the two results by some examples.

Example 4. Consider the following nonlinear system

$$\dot{x_1} = x_2 + x_1 x_2^2 \qquad \dot{x_2} = -x_1 + x_1^2 x_2$$

Let us assume the state invariant $x_1 \geq 0 \ \wedge \ x_2 \geq 0$. Gröbner basis computation reveals that $x_1 x_2 \in \mathit{Ideal}(p_1, p_2)$ and we get $x_1 \dot{x_1} - x_2 \dot{x_2} = 2x_1 x_2 = r$. A suitable function V such that $r = \boldsymbol{dVp}$ is $x_1^2/2 - x_2^2/2$. Clearly, by construction, $\dot{V} = 2x_1 x_2$ and $r \geq 0$ whenever $x_1 \geq 0$ and $x_2 \geq 0$. Hence, we conclude that V is always nondecreasing. In particular, this means that if $x_1 > x_2$ initially, then $x_1 > x_2$ always.

Example 5. Consider the nonlinear system

$$\dot{x_1} = x_1 - x_2 + x_1 x_2 \qquad \dot{x_2} = -x_2 - x_2^2.$$

The nonnegative definite polynomial x_2^2 is in the ideal generated by the $x_1 - x_2 + x_1 x_2$ and $-x_2 - x_2^2$ and correspondingly, we have $x_2^2 = -x_2(x_1 - x_2 + x_1 x_2) - x_1(-x_2 - x_2^2)$. Now, we notice that $\partial(-x_2)/\partial x_2 = \partial(-x_1)/\partial x_1 = -1$ and we get the corresponding V as $-x_1 x_2$. In all, we conclude that $-\dot{x_1 x_2} = x_2^2$. Since $x_2^2 \geq 0$ is always true, we can infer that the value of $-x_1 x_2$ is nondecreasing (Theorem 4).

Example 6. Consider another nonlinear system

$$\dot{x_1} = x_1 + 2x_2 + x_1 x_2^2 \qquad \dot{x_2} = 2x_1 + x_2 - x_1^2 x_2$$

Assume that we are given the state invariant $x_1 \geq 0$ and $x_2 \geq 0$.

We note that the polynomial $r = (x_1 + x_2)^2 + 2x_1 x_2$ is in the ideal of the two polynomials above and that $\boldsymbol{dVp} = r$ for $V = x_1^2/2 + x_2^2/2$. Now, r is nonnegative definite relative to the state invariant $x_1 \geq 0 \ \wedge \ x_2 \geq 0$. Hence, we conclude that V is nondecreasing.

A different choice of r in the ideal is $r = x_1^2 - x_2^2 + 2x_1^2 x_2^2$, where $r = \boldsymbol{dVp}$ for $V = x_1^2/2 - x_2^2/2$. The polynomial r can be expressed as $2V + 2x_1^2 x_2^2$ and hence $\Re \vdash V > 0 \Rightarrow r \geq 0$. Thus, if $V > 0$ in the initial states of the system, then $V > 0$ always subsequently. Note that we do not need the state invariant in this case.

New tools for effective sum of squares decomposition of polynomials are now available, which can be used to determine if a particular polynomial is positive or negative definite [20].

3.3 Oscillating Functions

Another useful class of functions that yield interesting reachability information are functions V whose value oscillates as the dynamical system evolves. If the frequency of oscillation was a constant, then such a V would satisfy the equation

$$\ddot{V} = -kV$$

But the frequency of oscillation is often not a constant and hence, finding V that satisfies the above property usually ends in failure. However, oscillating functions satisfy another very interesting property, which can be used to detect such behavior.

Let $\boldsymbol{p}$ be a vector field. Consider a set $\mathcal{V} = \{V_1, V_2, \ldots, V_k\}$ of k (polynomial) functions. Define the *extended monoid* of a $\mathcal{V}$ as the minimal set $Mon(\mathcal{V})$ such that (i) $\mathcal{V} \subset Mon(\mathcal{V})$, (ii) $V_1 V_2 \in Mon(\mathcal{V})$ whenever $V_1, V_2 \in Mon(\mathcal{V})$, and (iii) $rV_1 \in Mon(\mathcal{V})$ whenever $V_1 \in Mon(\mathcal{V})$ and r is nonpositive or nonnegative definite. In other words, the set $Mon(\mathcal{V})$ is the monoid over $\mathcal{V}$ *and* all nonpositive and nonnegative definite functions.

We say that a set of (polynomial) functions $V_1, V_2, \ldots, V_k$ is *closed under Lie derivative computation with respect to* $\boldsymbol{p}$ *upto multiplication* if for every function V_i, the Lie derivative of V_i w.r.t $\boldsymbol{p}$ is in $Mon(\mathcal{V})$.

For a given dynamical system $\dot{\boldsymbol{x}} = \boldsymbol{p}$, any set of functions that is closed under Lie derivative computation w.r.t $\boldsymbol{p}$ can provide useful information about oscillation or divergence of the system. We cannot state a formal theorem since the exact reachable sets depend on how the Lie derivatives are related, but we illustrate the method with a few examples.

Example 7. The Volterra predator-prey model [27] is given by

$$\dot{x_1} = -x_1 + x_1 x_2 \qquad \dot{x_2} = x_2 - x_1 x_2$$

where x_1 indicates the number of predators and x_2 indicates the number of prey. It is an easy exercise to note that the set of four polynomials $\mathcal{V} = \{x_1, x_2, (x_2 - 1), (x_1 - 1)\}$ is closed under Lie derivative computation. To see this, just factor the polynomials in the vector field $\boldsymbol{p}$ as follows:

$$\dot{x_1} = x_1(x_2 - 1) \qquad \dot{x_2} = -x_2(x_1 - 1)$$

The qualitative abstraction of this model [23] over the four polynomials in $\mathcal{V}$ shows the possible oscillatory behavior of the systems. Another choice of a closed set is $\{x_1 + x_2, x_2 - x_1, x_1 + x_2 - 2x_1x_2, 1 - 2x_1x_2\}$ and this can be used to refine the above abstraction [23].

Example 8. Consider the pendulum equations

$$\dot{x_1} = x_2 \qquad \dot{x_2} = -x_1 + x_1^3/6$$

Factoring the polynomial $-x_1 + x_1^3/6$ as $x_1(-1 + x_1^2/6)$, we note that the derivative of the factor $-1 + x_1^2/6$ is $2x_1x_2$, which is a product of x_1 and x_2. Hence, the set $\mathcal{V} = \{x_1, x_2, -1 + x_1^2/6\}$ is closed under Lie derivative computation. A qualitative abstraction over these three polynomials exhibits oscillatory behavior [23].

Example 7 also shows a weakness of the polynomial-based qualitative abstraction approach. *If* we only use polynomials, then *any* qualitative abstraction of the system in Example 7 would have trajectories that allow the dynamics to collapse onto one of the axes ($x_1 = 0$ or $x_2 = 0$) even if the initial state has $x_1 \neq 0$ and $x_2 \neq 0$. In this example, we really need nonpolynomial functions (in particular, $\ln(x_1) - x_1 + \ln(x_2) - x_2$) to show that the system oscillates. Note that Theorem 2 can be used to determine such nonpolynomial invariants, but the computability issues are a challenge.

4 Computability Issues

The real value of the results presented in the preceding sections arises from the fact that the interesting functions V can be computed in the domain of polynomials. We describe some techniques that can be used for this purpose. Gröbner bases is a canonical representation for the ideal generated by a set of polynomials. They are also used for computing the bases for the set of syzygies of a set of polynomials.

4.1 Gröbner bases computation

Given a set $p_1, p_2, \ldots, p_n$ of polynomials, Gröbner basis computation algorithms [5] work by generating new polynomials p in the ideal of these n polynomials by repeatedly eliminating highest degree power-product terms from the polynomials $p_1, p_2, \ldots, p_n$. For example, if p_1 is $x_2 + x_1 x_2^2$ and p_2 is $-x_1 + x_1^2 x_2$ (see Example 4), then Gröbner basis computation generates the polynomial $p_3 = x_1 p_1 - x_2 p_2$ because this way the highest power-products in p_1 and p_2 are canceled. The new polynomial $p_3 = 2x_1 x_2$ is added to the original set of polynomials $\{p_1, p_2\}$. The new polynomial can be used to delete p_1 from this set and replace it by $p_1 - x_2 p_3/2 = x_2$. Similarly, p_2 can be replaced by x_1. The polynomial x_2 can be used to delete p_3. Thus, the set $\{x_1, x_2\}$ is a Gröbner basis of $\{p_1, p_2\}$. It is routine to generate the 1-form $\boldsymbol{q}^T$ s.t. $\boldsymbol{q}^T \boldsymbol{p}$ is equal to the polynomials generated in this procedure. For example, $x_2 = (-x_1 x_2/2 + 1)p_1 - x_2 p_2$.

We are interested in $r \in \mathit{Ideal}(\boldsymbol{p})$ s.t. if $r = \boldsymbol{q}^T \boldsymbol{p}$, then the 1-form $\boldsymbol{q}^T$ is exact. The polynomials r in the final Gröbner basis need not satisfy this condition. But each of the intermediate polynomials generated during the computation of a Gröbner basis can be tested. In the above example, the 1-form corresponding to the intermediate polynomial p_3 is indeed exact. This is the form used in Example 4. It is also observed that all the polynomials used in other examples in this paper can be similarly generated.

It should be emphasized that the method outlined above is not *complete*, that is, there could be elements in $\mathit{Ideal}(\boldsymbol{p})$ which correspond to exact 1-forms which are not tested by the procedure. It is a very interesting problem for future work to determine if the set of ideal members generated by exact 1-forms is computable. Define

$$\mathit{ExactIdeal}(\boldsymbol{p}) = \{\boldsymbol{q}^T \boldsymbol{p} : \boldsymbol{q}^T \text{ is exact}\}$$

As far as the authors know, computability of this set is open. But as in several other practically useful real algebraic geometry computational procedures, polynomials of bounded degree in $\mathit{ExactIdeal}(\boldsymbol{p})$ can be generated and used.

4.2 Syzygy computation

In our methods, interest was in syzygies which were also exact. The Gröbner basis method can be used to construct a basis for the set of syzygies for polynomials $p_1, p_2, \ldots, p_n$ as follows: the polynomials $p_1, \ldots, p_n$ are partitioned into two non disjoint sets, say $\{p_1\}$ and $\{p_2, \ldots, p_n\}$. Next Gröbner basis is computed for the set $\{p_1(1-Y), p_2Y, p_3Y, \ldots, p_nY\}$ where Y is a new variable. Any polynomial p in the Gröbner basis *that does not contain* Y gives a syzygy. By repeating this process with $\{p_2, \ldots, p_n\}$, we can get the generators for the set of syzygies for $p_1, \ldots, p_n$ [5].

Each of the syzygies generated needs to be tested for being exact. This test can be done using the Frobenius theorem. The corresponding polynomial can then be easily generated by simple symbolic integration routines. Again note that the method described here is not complete. Just as in the case of Gröbner basis, we can define the set *ExactSyzygy*$(\boldsymbol{p})$ and a challenge for future work is to determine if this set is computable.

4.3 Linear constraint solver

A second technique for generating the functions V with the required properties is to assume that V is of a bounded degree, say it is quadratic with unknown coefficients. The properties required to be satisfied by V impose *linear constraints* on the unknown coefficient variables. Using a linear arithmetic solver, these constraints can be tested for satisfiability. Note that this method, though attractive, cannot be used for the techniques in Section 3.2 because they additionally involve unknown positive or negative definite functions.

Example 9. Consider the four-dimensional nonlinear system

$$\begin{array}{ll} \dot{x_1} = x_2 & \dot{x_2} = -x_1/2 - x_2 \\ \dot{x_3} = x_4 & \dot{x_4} = -x_3 + 2x_1x_2 - 2x_2^2 \end{array}$$

If we guess that a function V of the form $x_1^2 + bx_1x_3 + cx_1 + dx_3$ satisfies the equation $\ddot{V} = \lambda V$, then we get the following constraint:

$$\begin{aligned} \ddot{F} = {} & 2x_1(-x_1/2 - x_2) + 2x_2^2 + 2bx_2x_4 + bx_1(-x_3 + 2x_1x_2 - 2x_2^2) + \\ & bx_3(-x_1/2 - x_2) + c(-x_1/2 - x_2) + d(-x_3 + 2x_1x_2 - 2x_2^2) \\ = {} & \lambda F \end{aligned}$$

This gives rise to the following linear constraints over the variables a, b, c, d and λ.

$$\begin{array}{rrrr} -1 = \lambda & -2 + 2d = 0 & 2 - 2d = 0 & 2b = 0 \\ -b - b/2 = 0 & 2b = 0 & -2b = 0 & -b = 0 \\ -c/2 = \lambda c & c = 0 & -d = \lambda d & \end{array}$$

A satisfying assignment is $\lambda = -1$, $d = 1$, and $b = c = 0$. Thus, $x_1^2 + x_3$ is the required function.

5 Related Work and Conclusion

There is a lot of work in the theory of nonlinear systems [27, 21]. Energy functions are used to get analytical descriptions of trajectories and provide arguments for stability or periodicity. However, the problem of generating these functions and issues about computability have not been addressed. Moreover, such functions have not been used to get over approximations of the reach sets. These features distinguish this work from the well established theory of nonlinear systems.

Techniques from algebraic geometry, mainly Gröbner basis methods, have been used for generating switching surfaces [28], and generating constraints on parameters and bounds for determining Lyapunov functions for *local* stability regions [9, 8]. In most of these applications, Gröbner basis is used as a quantifier elimination procedure – a simpler alternative to quantifier elimination in the theory of reals [4].

This paper presents some first results on computing interesting polynomial functions for nonlinear systems. These functions can be used in one of two ways: they can generate over approximations of the reach set and this information can be used inside any tool for computing reachability such as [11], or they can be used as predicates in an abstraction framework to generate good abstractions, such as [2]. In particular, they can be used in the hybrid qualitative abstraction approach [23].

This work opens several interesting directions for future work. On the theoretical side, one can ask the question if we can get decidability of reachability for certain classes of nonlinear systems, whenever sufficiently many such energy functions V's exist. In the linear case, we know the answer is positive [16]. Furthermore, as in the linear case, can we extend the computational methods to richer decidable theories than the theory of reals? These decidability results can then be used to get newer classes of hybrid systems with decidable reachability problem [15, 14, 3]. In the field of computational algebraic geometry, the challenge is to find if the sets exact-ideal and exact-syzygy are computable.

The theory outlined in this paper is useful even when the answers to the above questions are unknown. Construction of useful polynomial functions can be automated using bounded degree approximations as described in this paper. And it can be made more powerful using other existing tools, such as the sum of squares tool [20], which also shows how incomplete techniques can still be very effective in solving real and challenging problems [19, 18].

References

[1] R. Alur, T. Dang, and F. Ivancic. Reachability analysis of hybrid systems via predicate abstraction. In Tomlin and Greenstreet [25].

[2] R. Alur, T. Dang, and F. Ivancic. Progress on reachability analysis of hybrid systems using predicate abstraction. In Maler and Pnueli [17].

[3] R. Alur, T. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(2):971–984, July 2000.

[4] H. Anai and V. Weispfenning. Reach set computations using real quantifier elimination. In M. D. Di Benedetto and A. L. Sangiovanni-Vincentelli, editors, *HSCC*, volume 2034 of *Lecture Notes in Computer Science*, pages 63–76. Springer, 2001.

[5] T. Becker and V. Weispfenning. *Gröbner bases: a computational approach to commutative algebra*. Springer-Verlag, Berlin, 1993.

[6] A. Chutinan and B. H. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In Vaandrager and van Schuppen [26], pages 76–90.

[7] T. Dang and O. Maler. Reachability analysis via face lifting. In T. A. Henzinger and S. Sastry, editors, *HSCC*, volume 1386 of *LNCS*, pages 96–109. Springer, 1998.

[8] K. Forsman. Construction of Lyapunov functions using Gröbner bases. In *30th CDC*. IEEE, 1991.

[9] K. Forsman and T. Glad. Constructive algebraic geometry in nonlinear control. In *29th CDC*, volume 5, pages 2825–2827. IEEE, 1990.

[10] S. Graf and H. Saïdi. Construction of abstract state graphs with PVS. In O. Grumberg, editor, *Proc. of 9th Conference on Computer-Aided Verification (CAV'97)*, volume 1254, pages 72–83. Springer Verlag, 1997.

[11] B. H. Krogh and O. Stursberg. On efficient representation and computation of reachable sets for hybrid systems. In Maler and Pnueli [17].

[12] B. J. Kuipers. *Qualitative reasoning: Modeling and simulation with incomplete knowledge*. MIT Press, Cambridge, MA, 1994.

[13] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In N. A. Lynch and B. H. Krogh, editors, *HSCC*, volume 1790 of *LNCS*, pages 202–214. Springer, 2000.

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

[15] G. Lafferriere, G. J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In Vaandrager and van Schuppen [26], pages 137–151.

[16] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computations for families of linear vector fields. *J. Symbolic Computation*, 32(3):231–253, 2001.

[17] O. Maler and A. Pnueli, editors. *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003*, volume 2623 of *LNCS*. Springer, April 2003.

[18] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometric methods in robustness and optimization*. PhD thesis, California Institute of Technology, Pasadena, 2000.

[19] P. A. Parrilo and B. Sturmfels. Minimizing polynomial functions. In *Algorithmic and quantitative real algebraic geometry*, volume 60 of *DIMACS series in discrete mathematics and theoretical computer science*, pages 83–99, 2003. `http://www/arxiv.org/abs/math.OC/0103170`.

[20] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. *SOSTOOLS: Sum of square optimization toolbox for MATLAB*, 2002.

[21] S. Sastry. *Nonlinear systems:Analysis, stability, and control*. Springer, 1999.

[22] A. Tiwari. Approximate reachability for linear systems. In Maler and Pnueli [17], pages 514–525.

[23] A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. In Tomlin and Greenstreet [25], pages 465–478.

[24] A. Tiwari, D. Wolf, A. Arkin, and P. Lincoln. Hybrid modeling and analysis of genetic regulatory networks: Sporulation initiation in bacillus subtilis, 2003. In preparation.

[25] C. Tomlin and M. R. Greenstreet, editors. *Hybrid Systems: Computation and Control, 5th International Workshop, HSCC 2002, Proceedings*, volume 2289 of *Lecture Notes in Computer Science*. Springer, 2002.

[26] F. W. Vaandrager and J. H. van Schuppen, editors. *Hybrid Systems: Computation and Control, Second International Workshop, HSCC'99, Proceedings*, volume 1569 of *Lecture Notes in Computer Science*. Springer, 1999.

[27] M. Vidyasagar. *Nonlinear systems analysis.* Prentice Hall, 1993.

[28] U. Walther, T. Georgiou, and A. Tannenbaum. On the computation of switching surfaces in optimal control:A Gröbner basis approach. *IEEE Transactions on Automatic Control*, 46(4), 2001.