Constructing Attractors of Nonlinear Dynamical Systems by State Space Decomposition

Laurent Fribourg[1], Ulrich Kühne[2], and Romain Soulat[1]

1 **LSV, ENS Cachan & CNRS**
Cachan, France
2 **University of Bremen**
Bremen, Germany

Abstract

In a previous work, we have shown how to generate attractor sets of affine hybrid systems using a method of state space decomposition. We show here how to adapt the method to polynomial dynamics systems by approximating them as switched affine systems. We show the practical interest of the method on standard examples of the literature.

1998 ACM Subject Classification I.2.8, Control theory

Keywords and phrases Control theory, Hybrid Systems, Nonlinear dynamical systems

Digital Object Identifier 10.4230/OASIcs.FSFMA.2013.53

1 Introduction

The symbolic analysis of nonlinear dynamical systems has recently attracted considerable attention: the problem of computing the set of reachable states (reachability analysis) has thus been studied in [1, 3, 4, 5, 2], and the problem of computing polytopic invariants (invariant synthesis) has been studied in [10, 11, 12]. Here, we study a problem close to the problem of invariant synthesis: we want not only to generate a polytopic invariant P included in a given rectangle R, but we also want that all the trajectories starting from R converge to P. In other words, we want to construct an *attractor set* P of R, ideally as small as possible. We show that the state decomposition method given in [7] for computing attractors of linear systems can be extended to the case of polynomial dynamics, using the idea of local linearization developed in [1].

The plan of the paper is as follows. In Section 2, we recall the principles of the state space decomposition method for linear dynamical systems. In Section 3, we explain how to extend the method to polynomial dynamical systems. In Section 4, we apply the method to two standard examples of the literature. We conclude in Section 5.

2 Attractors for Linear Dynamics

We suppose that we are given a finite set $U = \{1, \ldots, N\}$ of elements called *modes*. We are also given a family of functions $\{f_u\}_{u \in U}$ with $f_u : \mathbb{R}^n \to \mathbb{R}^n$. Given a time step τ, a *sampled switched system* Σ is a dynamical system governed by an equation of the form $x(t + \tau) = f_\sigma(x(t))$, where σ is a *control signal*, which selects a mode $u \in U$ at each time step $\tau, 2\tau, \ldots$.

A *k-pattern* is a sequence of at most k modes of U. Given a set $X \subset \mathbb{R}^n$ and a mode $u \in U$, we define the *set of successors of* X *via* u, and denote by $Post_{f_u}(X)$, the set $\{x' \in \mathbb{R}^n \mid f_u(x) = x' \text{ for some } x \in X\}$. Given a pattern π of the form $(u_1 \cdot u_2 \cdot \cdots \cdot$

1st French Singaporean Workshop on Formal Methods and Applications 2013 (FSFMA'13).
Editors: Christine Choppy and Jun Sun; pp. 53–60

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

u_m), the *set of successors of* X *via* π, denoted by $Post_{f_\pi}(X)$, is given by: $Post_{f_\pi}(X) = Post_{f_{u_m}}(\cdots(Post_{f_{u_2}}(Post_{f_{u_1}}(X)))\cdots)$.

Suppose that we are given a *box* $R \subseteq \mathbb{R}^n$ (i.e., a cartesian product of closed intervals). We have given in [7] a general method in order to show the *controlled invariance* of Σ in R. By controlled invariance in R, we mean that if the system state is in R at some time, it will stay forever in R under the control of an appropriate signal σ. The method constructs a *k-decomposition* of R, that is, a set Δ of the form $\{(V_i, \pi_i)\}_{i\in I}$, where I is a finite set of indices, the V_is are sub-boxes of R, and the π_is are k-patterns. Furthermore, this decomposition Δ is *k-invariant* in the sense:

1. $\bigcup_{i\in I} V_i = R$
2. $Post_{f_{\pi_i}}(V_i) \subset R$, for all $i \in I$.

An algorithm of decomposition is given in [7], and is recalled in Appendix A: given a dynamical system $\{f_u\}_{u\in U}$ and a box R, it returns a k-invariant decomposition Δ of R.

▶ **Lemma 1.** *If* $\Delta = \{(V_i, \pi_i)\}_{i\in I}$ *is a k-invariant decomposition of* R*, then:*

$$Post_\Delta(R) \subset R,$$

where the operator $Post_\Delta$ *is defined, for all* $X \subset \mathbb{R}^n$*, by:*

$$Post_\Delta(X) = \bigcup_{i\in I} Post_{f_{\pi_i}}(X \cap V_i)$$

▶ **Lemma 2.** *Consider a k-invariant decomposition* $\Delta = \{(V_i, \pi_i)\}_{i\in I}$ *of* R*. The sequence* $\{R^j_\Delta\}_{j\geq 0}$ *defined by:*

- $R^0_\Delta = R$,
- $R^{j+1}_\Delta = Post_\Delta(R^j_\Delta)$

is a decreasing nested sequence and the set $R^*_\Delta = \bigcap_{j\geq 0} R^j_\Delta$ *is well-defined. Furthermore,* R^*_Δ *is an* attractor set *of* R*, i.e.:*

1. $Post_\Delta(R^*_\Delta) = R^*_\Delta$ *(*invariance*)*
2. $\forall x \in R,\ d(Post^j_\Delta(x), R^*_\Delta) \to 0$ *as* j *tends to* ∞[1] *(*attractivity*).*

Attractors and limit cycles have been studied in the context of affine dynamics in [6].

3 Nonlinear Dynamics

The decomposition procedure, explained in Section 2, is quite general, and does not suppose that the functions f_u are linear or affine. However, in the case where f_u is an affine function, the computation of the successor sets (via *Post* operator) can be done in an *exact* manner.

We now explain how to apply the state space decomposition procedure in the case of non-affine dynamics. This is done at the price of an *over-approximation* of the successor sets. Following [1], we compute (an overapproximation of) the successor sets using *local linearizations* of the system, and enlargement of the linear images by addition of *error* intervals. We will consider a system governed by a *unique* equation of the form $x(t+\tau) = f(x(t))$ where f is a polynomial. The set U is thus reduced to a single element ($U = \{1\}$). A pattern π_i associated to a subregion V_i, is now just an *integer* indicating the number of times the (local linearization of) f should be applied when the state is in V_i.

[1] d is the distance between a point and a subset of $\mathbb{R}^n$

3.1 Affine systems with uncertainty

As in [1], reachable sets are represented here by zonotopes. They are chosen because linear transformations and Minkowski sums[2] can be computed efficiently, allowing to compute reachable sets for large scale linear systems in continuous space. A zonotope is defined by a center c to which linear segments $l_i = \beta^{(i)} \cdot g^{(i)}$, $-1 \leq \beta^{(i)} \leq 1$ are added via Minkowski sum.

▶ **Definition 3.** A *zonotope* is a set

$$Z = \{x \in \mathbb{R}^n : x = c + \Sigma_{i=1}^{p} \beta^{(i)} \cdot g^{(i)},\ -1 \leq \beta^{(i)} \leq 1\}$$

with $c, g^{(1)}, \ldots, g^{(p)} \in \mathbb{R}^n$. The vectors $g^{(1)}, \ldots, g^{(p)}$ are referred to as the *generators* and c as the *center* of the zonotope. It is convenient to represent the set of generator as a matrix G. The notation is $< c, G >$, where the first element refers to the center of the zonotope and the second to the generators.

Zonotopes allow to extend easily the decomposition procedure in order to take into account small perturbations of the system dynamics (see [8]). Suppose that we the system is described by an equation of the form

$$x(t + \tau) = f_{lin}(x(t)) + \varepsilon$$

where:

- f_{lin} is an affine function defined by $f_{lin}(x) = Ax + b$ with $A \in \mathbb{R}^{n \times b}$, $b \in \mathbb{R}^n$
- ε is a disturbance vector belonging to a rectangle region $\Lambda = [-\varepsilon_1, +\varepsilon_1] \times \cdots [-\varepsilon_n, +\varepsilon_n]$ of $\mathbb{R}^n$, with $\varepsilon_i \geq 0$ for all i.

Since Λ is a product of intervals centered in 0, it can be written as a zonotope

$$Z_\Lambda =< 0, G_\Lambda > \text{ with } G_\Lambda = \begin{pmatrix} \varepsilon_1 & 0 & \ldots & 0 \\ 0 & \varepsilon_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \varepsilon_n \end{pmatrix}$$

▶ **Lemma 4.** *Consider a zonotope $Z =< c, G >$ with G a square matrix, a box $\Lambda = [-\varepsilon_1, +\varepsilon_1] \times \cdots [-\varepsilon_n, +\varepsilon_n]$ of $\mathbb{R}^n$, and a function f defined by:*
$f(x) = Ax + b + \varepsilon$, *with $\varepsilon \in \Lambda$.*
We have: $Post_f(Z) \subset < Ac + b, AG + G_\Lambda >$.

3.2 Linearization of nonlinear dynamics

Consider now a system governed by equation $x(t + \tau) = f(x(t))$ where f is a polynomial. We can write:

$$f(x) = f_{lin}(x) + P(x),$$

where $f_{lin}(x)$ corresponds to the polynomial subpart of order 1, and P to the polynomial of order greater than or equal to 2. We can then apply the method explained in Section 3.1, by computing a local over-approximation Λ of $P(x)$.

[2] The Minkowski of two sets A, B is defined by $A + B = \{a + b \mid a \in A, b \in B\}$

▶ **Lemma 5.** *Consider a function f defined by: $f(x) = f_{lin}(x) + P(x)$, where $f_{lin}(x)$ is a 1st-order polynomial of the form $b + Ax$, and $P(x)$ a 2nd-order polynomial. Given a zonotope $Z :< c, G >$, we have:*

$$Post_f(Z) \subset Post_{f_{lin}}(Z) + Z_\Lambda$$

with:

- $Post_{f_{lin}}(Z) =< f(c), AG >$
- $Z_\Lambda =< 0, \begin{pmatrix} \varepsilon_1(Z) & 0 & \dots & 0 \\ 0 & \varepsilon_2(Z) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \varepsilon_n(Z) \end{pmatrix} >$

with $(1 \leq i \leq n)$: $\varepsilon_i(Z) = max_{x \in Z}(|P_i(x) - P_i(c)|)$.

Now, in order to apply the decomposition procedure (extended with error), we just have to find an upper bound for $|P(x) - P(c)|$ componentwise. In the following, we explain on two standard examples how to compute such upper bounds. Then we apply the decomposition procedure in order to find a decomposition Δ, and construct an attractor related to R^*_Δ.

4 Case studies

These examples are taken from [2]. Given a zonotope $Z =< c, G >$, we explain how to compute $Post_{f_{lin}}(Z)$ and Z_Λ appearing in Lemma 5. Experiments have been performed with the tool MINIMATOR [9] on a machine equipped with an Intel Core2 at 2.93GHz and 2 GB of RAM memory.

4.1 Van der Pol oscillator

4.1.1 Dynamics

The dynamics of the Van der Pol oscillator are the following:

$$x(\tau) = \begin{pmatrix} 1 & \tau \\ -\tau & 1+\tau \end{pmatrix} x(0) + \begin{pmatrix} 0 \\ -x_1(0)^2 x_2(0)\tau \end{pmatrix}.$$

When linearized to a point $c \in \mathbb{R}^2$, this gives:

$$x(\tau) = \begin{pmatrix} 1 & \tau \\ -\tau & 1+\tau \end{pmatrix} x(0) + \begin{pmatrix} 0 \\ -c_1^2 c_2 \tau \end{pmatrix}.$$

Thus, we have $Post_{f_{lin}}(Z) = \begin{pmatrix} 1 & \tau \\ -\tau & 1+\tau \end{pmatrix} Z + \begin{pmatrix} 0 \\ -c_1^2 c_2 \tau \end{pmatrix} = \begin{pmatrix} 1 & \tau \\ -\tau & 1+\tau(1-c_1^2) \end{pmatrix} x(0)$. It is easy to see that for a box $V \subset \mathbb{R}^2$ we are making an error of at most 0 on the x axis and $|(c_1^2 - (c_1 + G_{1,2} + G_{2,2})^2|\tau$ on the y axis, when $Z =< c, G >$ with $c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$ and $G = \begin{pmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{pmatrix}$ Thus we need to enlarge any image of a zonotope $Z =< c, G >$ by 0 on the x-axis and $\tau|(C_1^2 - (C_1 + G_{1,2} + G_{2,2})^2|$ on the y-axis (i.e., $Z_\Lambda =< 0, \begin{pmatrix} 0 & 0 \\ 0 & |(C_1^2 - (C_1 + G_{1,2} + G_{2,2})^2|\tau \end{pmatrix} >$).

4.1.2 Attractor Construction

The Decomposition procedure is applied to $R = [-3, 3] \times [-3, 3]$ and $\tau = 0.01$ (with parameters $k = 30$, $d = 7$). At boxes located around the center of R, the length of patterns is 1 while in

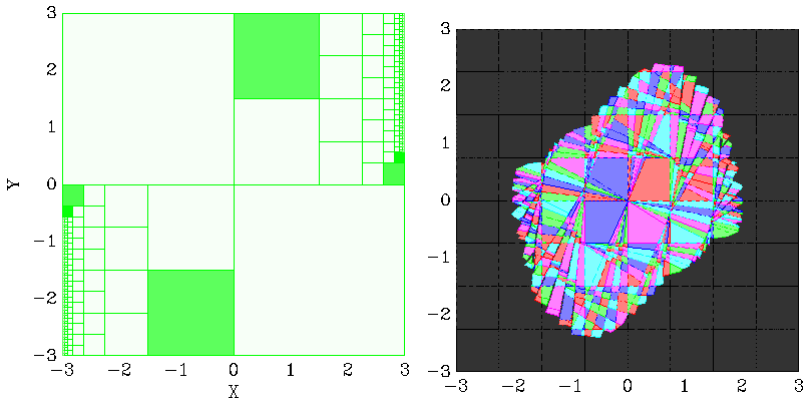

Figure 1 Decomposition for the Van der Pol oscillator (left) ; R^j_Δ for $j = 30$ (right).

the lower left and upper right edges, the length is up to 30. The result of the Decomposition is depicted in the left part of Figure 1 and the attractor set R^*_Δ in the right part. Experiments took 8 minutes to complete.

4.2 FitzHugh-Nagumo Neuron

4.2.1 Dynamics

The dynamics of the FitzHugh-Nagumo neuron are the following:

$$x(\tau) = \begin{pmatrix} 1+\tau & -\tau \\ 0.08\tau & -0.0064\tau + 1 \end{pmatrix} x(0) + \begin{pmatrix} -x_1(0)^3\tau/3 + 0.875\tau \\ 0.056\tau \end{pmatrix}$$

When linearized to a point $c \in \mathbb{R}^2$, this gives:

$$x(\tau) = \begin{pmatrix} 1+\tau & -\tau \\ 0.08\tau & -0.0064\tau + 1 \end{pmatrix} x(0) + \begin{pmatrix} -c_1^3\tau/3 + 0.875\tau \\ 0.056\tau \end{pmatrix}$$

It is easy to see that for a box $V \subset \mathbb{R}^2$ we are making an error of at most $max_{x\in V}(\frac{|x_1^3 - c_1^3|}{3})\tau$ on the x axis and 0 on the y axis. Thus we need to enlarge any image of a zonotope Z by $max_{x\in Z}(\frac{|x_1^3 - c_1^3|}{3})\tau$ on the x-axis and 0 on the y-axis ($Z_\Lambda =< 0, \begin{pmatrix} max_{x\in Z}(\frac{|x_1^3 - c_1^3|}{3})\tau & 0 \\ 0 & 0 \end{pmatrix} >$).

4.2.2 Attractor construction

The Decomposition procedure is applied to $R = [-2.5, 2.5] \times [-0.5, 2.5]$ and $\tau = 0.1$ (with parameters $k = 30$, $d = 7$). For boxes located around the center of R, the length of patterns is 1 while in the lower left and upper right corners, the length is up to 22. The result of the Decomposition is depicted in the left part of Figure 2 and the attractor set R^*_Δ in the right part. Experiments took 5 minutes to complete.

5 Future work

We have explained how to construct attractors of polynomial dynamical systems by extending a method designed for linear dynamical systems. The method consists in considering the subpolynomial subpart of order greater than 1 as a perturbation that is over-approximated.

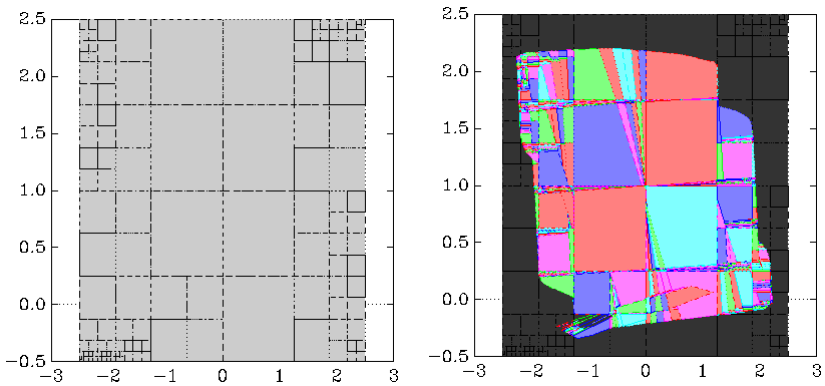

Figure 2 Decomposition for the FitzHugh-Nagumo Neuron (left) ; R^j_Δ for $j = 30$ (right).

So far, the over-approximation is done in an *ad hoc* fashion for each specific example. For future work, we plan to consolidate the method by using the formal technique of linearization of [1], based on the notion of Lagrange remainder.

A Appendix: Decomposition Algorithm

The Decomposition procedure generates a k-invariant decomposition of R, as follows:

It first calls sub-procedure Find_Pattern in order to get a k-pattern such that R is R-invariant. If it succeeds, then it is done. Otherwise, it divides R into 2^n sub-boxes $V_1, \ldots, V_{2^n}$ of equal size. If for each V_i, Find_Pattern gets a k-pattern making it R-invariant, it is done. If, for some V_j, no such pattern exists, the procedure is recursively applied to V_j. It ends with success when a k-invariant decomposition of R is found, or failure when the maximal degree d of decomposition is reached.

The algorithmic form of the procedure is given in Algorithms 1 and 2. (For the sake of simplicity, we consider the case of dimension $n = 2$, but the extension to $n > 2$ is straightforward.) The main procedure Decomposition(W,R,D,K) is called with R as input value for W, d for input value for D, and k as input value for K; it returns either $\langle \{(V_i, \pi_i)\}_i, True \rangle$ with $\bigcup_i V_i = W$ and $\bigcup_i Post_{\pi_i}(V_i) \subseteq R$, or $\langle \_, False \rangle$. Procedure Find_Pattern(W,R,K) looks for a K-pattern for which W is R-invariant: it selects all the K-patterns (which are in finite number) by non-decreasing length order until either it finds such a pattern π (output: $\langle \pi, True \rangle$), or no one exists (output: $\langle \_, False \rangle$).

The correctness of the procedure is stated as follows.

▶ **Theorem 6.** *If Decomposition(R,R,d,k) returns $\langle \Delta, True \rangle$, then Δ is a k-invariant decomposition of R.*

Algorithm 1: Decomposition(W,R,D,K)

Input: A box W, a box R, a degree D of decomposition, a length K of pattern
Output: $\langle\{(V_i, \pi_i)\}_i, True\rangle$ with $\bigcup_i V_i = W$ and $\bigcup_i Post_{\pi_i}(V_i) \subseteq R$, or $\langle \_, False\rangle$

$(\pi, b) := Find\_Pattern(W, R, K)$
if $b = True$ **then**
  return $\langle\{(W, \pi)\}, True\rangle$
else
  if $D = 0$ **then**
    return $\langle \_, False\rangle$
  else
    Divide equally W into (W_1, W_2, W_3, W_4) `/* (case` $n = 2$`)` `*/`
    (Δ_1, b_1) := Decomposition(W_1,R,$D-1$,K)
    (Δ_2, b_2) := Decomposition(W_2,R,$D-1$,K)
    (Δ_3, b_3) := Decomposition(W_3,R,$D-1$,K)
    (Δ_4, b_4) := Decomposition(W_4,R,$D-1$,K)
    return $(\Delta_1 \cap \Delta_2 \cap \Delta_3 \cap \Delta_4, b_1 \wedge b_2 \wedge b_3 \wedge b_4)$

Algorithm 2: Find_Pattern(W,R,K)

Input: A box W, a box R, a length K of pattern
Output: $\langle \pi, True\rangle$ with $Post_\pi(W) \subseteq R$, or $\langle \_, False\rangle$ when no pattern maps W into R

for $i = 1 \ldots K$ **do**
  Π := set of patterns of length i
  while Π *is non empty* **do**
    Select π in Π
    $\Pi := \Pi \setminus \{\pi\}$
    if $Post_\pi(W) \subseteq R$ **then**
      return $\langle \pi, True\rangle$
return $\langle \_, False\rangle$

References

1 M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *CDC*, pages 4042–4048. IEEE, 2008.

2 M. Amin Ben Sassi, R. Testylier, T. Dang, and A. Girard. Reachability analysis of polynomial systems using linear programming relaxations. In *ATVA*, pages 137–151, 2012.

3 Eugene Asarin, Thao Dang, and Antoine Girard. Reachability analysis of nonlinear systems using conservative approximation. In Oded Maler and Amir Pnueli, editors, *HSCC*, volume 2623 of *Lecture Notes in Computer Science*, pages 20–35. Springer, 2003.

4 Thao Dang, Colas Le Guernic, and Oded Maler. Computing reachable states for nonlinear biological models. In *CMSB*, pages 126–141, 2009.

5 G. Frehse. PHAVer: algorithmic verification of hybrid systems past HyTech. *STTT*, 10(3):263–279, 2008.

6 L. Fribourg and R. Soulat. Limit cycles of controlled switched systems: Existence, stability, sensitivity. In *Proc. 3rd NCMIP 2013 IOP Publishing "Journal of Physics: Conference Series"*, May 2013. To appear.

7 Laurent Fribourg and Romain Soulat. Finite controlled invariants for sampled switched systems. Research Report LSV-13-09, Laboratoire Spécification et Vérification, ENS Cachan, France, April 2013. 27 pages.

8 W. Kühn. Zonotope dynamics in numerical quality control. *Mathematical Visualization*, pages 125–134, 1998.

9 Minimator Team. Minimator Web page. http://www.lsv.ens-cachan.fr/ soulat/minimator/, 2013.

10 Mohamed Amin Ben Sassi and Antoine Girard. Controller synthesis for robust invariance of polynomial dynamical systems using linear programming. *CoRR*, abs/1107.1580, 2011.

11 Mohamed Amin Ben Sassi and Antoine Girard. Computation of polytopic invariants for polynomial dynamical systems using linear programming. *Automatica*, 48(12):3114–3121, 2012.

12 Mohamed Amin Ben Sassi and Antoine Girard. Control of polynomial dynamical systems on rectangles. In *European Control Conference*, 2013.