Constructing Attractors of Nonlinear Dynamical Systems by State Space Decomposition

Laurent Fribourg\textsuperscript{1}, Ulrich Kühne\textsuperscript{2}, and Romain Soulat\textsuperscript{1}

\textsuperscript{1} LSV, ENS Cachan & CNRS
Cachan, France
\textsuperscript{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 \( f_u : \mathbb{R}^n \rightarrow \mathbb{R}^n \). Given a time step \( \tau \), a sampled switched system \( \Sigma \) is a dynamical system governed by an equation of the form \( x(t+\tau) = f_\sigma(x(t)) \), where \( \sigma \) 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 \( \text{Post}_{f_u}(X) \), the set \( \{x' \in \mathbb{R}^n \mid f_u(x) = x' \text{ for some } x \in X\} \). Given a pattern \( \pi \) of the form \( (u_1 \cdot u_2 \cdot \ldots \cdot u_k) \),
Constructing Attractors of Nonlinear Dynamical Systems

We will consider a system governed by a dynamical system \( f \) that is a polynomial. The set \( \{\mathcal{V}_i, \pi_i\}_{i \in I} \) is a decreasing nested sequence and the set \( R^*_\Delta = \bigcap_{j \geq 0} R^j_\Delta \) is well-defined. Furthermore, \( R^*_\Delta \) is an attractor set of \( f \), i.e.:  
1. \( \forall x \in R, \; d(\text{Post}_\Delta^j(x), R^*_\Delta) \to 0 \) as \( j \) tends to \( \infty^1 \) (attractivity). 
2. \( \forall x \in R \), \( d(\text{Post}_\Delta^j(x), R^*_\Delta) \to 0 \) as \( j \) tends to \( \infty \) (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 \( f \) are linear or affine. However, in the case where \( f \) 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 \( \pi_i \) associated to a subregion \( \mathcal{V}_i \), is now just an integer indicating the number of times the (local linearization of) \( f \) should be applied when the state is in \( \mathcal{V}_i \).

\(^{1}\) \( d \) 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 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

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

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

\[
x(t + \tau) = f_{\text{lin}}(x(t)) + \epsilon
\]

where:
- \( f_{\text{lin}} \) is an affine function defined by \( f_{\text{lin}}(x) = Ax + b \) with \( A \in \mathbb{R}^{n \times k}, b \in \mathbb{R}^n \)
- \( \epsilon \) is a disturbance vector belonging to a rectangle region \( \Lambda = [-\epsilon_1, +\epsilon_1] \times \cdots [-\epsilon_n, +\epsilon_n] \) of \( \mathbb{R}^n \), with \( \epsilon_i \geq 0 \) for all \( i \).

Since \( \Lambda \) is a product of intervals centered in 0, it can be written as a zonotope

\[
Z_\Lambda = <0, G_\Lambda> \quad \text{with} \quad G_\Lambda = \begin{pmatrix}
\epsilon_1 & 0 & \cdots & 0 \\
0 & \epsilon_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \epsilon_n
\end{pmatrix}
\]

**Lemma 4.** Consider a zonotope \( Z = <c, G> \) with \( G \) a square matrix, a box \( \Lambda = [-\epsilon_1, +\epsilon_1] \times \cdots [-\epsilon_n, +\epsilon_n] \) of \( \mathbb{R}^n \), and a function \( f \) defined by:

\[
f(x) = Ax + b + \epsilon, \quad \text{with} \quad \epsilon \in \Lambda.
\]

We have: \( \text{Post}_f(Z) < Ac + b, AG + G_\Lambda> \).

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:

\[
f(x) = f_{\text{lin}}(x) + P(x),
\]

where \( f_{\text{lin}}(x) \) 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 \( \Lambda \) of \( P(x) \).

---

2 The Minkowski of two sets \( A, B \) is defined by \( A + B = \{ a + b \mid a \in A, b \in B \} \)
Constructing Attractors of Nonlinear Dynamical Systems

Lemma 5. Consider a function \( f \) defined by: \( f(x) = f_{\text{lin}}(x) + P(x) \), where \( f_{\text{lin}}(x) \) 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:

\[
\text{Post}_{f}(Z) \subset \text{Post}_{f_{\text{lin}}}(Z) + Z_{\Lambda}
\]

with:

- \( - \text{Post}_{f_{\text{lin}}}(Z) = < f(c), AG > \)
- \( Z_{\Lambda} = < 0, \begin{pmatrix} \varepsilon_{1}(Z) & 0 & \ldots & 0 \\ 0 & \varepsilon_{2}(Z) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \varepsilon_{n}(Z) \end{pmatrix} > \)

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 \( \Delta \), and construct an attractor related to \( R_{\Delta}^{*} \).

4 Case studies

These examples are taken from [2]. Given a zonotope \( Z := c, G \), we explain how to compute \( \text{Post}_{f_{\text{lin}}}(Z) \) and \( Z_{\Lambda} \) 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 \( \text{Post}_{f_{\text{lin}}}(Z) = \begin{pmatrix} 1 & \tau \\ -\tau & 1 + \tau \end{pmatrix} Z + \begin{pmatrix} 0 \\ -c_{1}^{2}c_{2}\tau \end{pmatrix} = \begin{pmatrix} 1 & \tau \\ -\tau & 1 + \tau(1 - c_{1}^{2}) \end{pmatrix} x(0) \).

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 \(|c_{1}^{2} - (c_{1} + G_{1,2} + G_{2,2})^{2}| \) 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., \( Z_{\Lambda} := 0, \begin{pmatrix} 0 \\ (C_{1}^{2} - (C_{1} + G_{1,2} + G_{2,2})^{2})\tau \end{pmatrix} > \)).

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
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 $R^*_\Delta$ 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:

$$x(\tau) = \begin{pmatrix} 1 + \tau & -\tau \\ 0.08 \tau & -0.0064 \tau + 1 \end{pmatrix} x(0) + \begin{pmatrix} -(x_1(0))^2/3 + 0.875 \tau \\ 0.056 \tau \end{pmatrix}$$

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

$$x(\tau) = \begin{pmatrix} 1 + \tau & -\tau \\ 0.08 \tau & -0.0064 \tau + 1 \end{pmatrix} x(0) + \begin{pmatrix} -c_1^2 \tau/3 + 0.875 \tau \\ 0.056 \tau \end{pmatrix}$$

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

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 $R^*_\Delta$ 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.
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.

### 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 $\pi$ (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 $\Delta$ is a $k$-invariant decomposition of $R$. 

---

**Figure 2** Decomposition for the FitzHugh-Nagumo Neuron (left) : $R^j_\Delta$ for $j = 30$ (right).
Algorithm 1: Decomposition\textup{(}W,R,D,K\textup{)}

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

1. $(\pi, b) := \text{Find Pattern}(W, R, K)$
   2. if $b = \text{True}$ then
      3. \; \text{return} $\langle\{(W, \pi)\}, \text{True}\rangle$
   4. else
      5. \; \text{if $D = 0$ then}
         6. \; \text{return} $\langle, \text{False}\rangle$
      7. else
         8. \; \text{Divide equally $W$ into $(W_1, W_2, W_3, W_4)$} \quad \text{/* (case $n = 2$) */}
      9. \; $(\Delta_1, b_1) := \text{Decomposition}(W_1, R, D - 1, K)$
     10. \; $(\Delta_2, b_2) := \text{Decomposition}(W_2, R, D - 1, K)$
     11. \; $(\Delta_3, b_3) := \text{Decomposition}(W_3, R, D - 1, K)$
     12. \; $(\Delta_4, b_4) := \text{Decomposition}(W_4, R, D - 1, K)$
     13. \; \text{return} $\langle\Delta_1 \cap \Delta_2 \cap \Delta_3 \cap \Delta_4, b_1 \land b_2 \land b_3 \land b_4\rangle$

Algorithm 2: Find\_Pattern\textup{(}W,R,K\textup{)}

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

1. \textbf{for} $i = 1 \ldots K$ \textbf{do}
2. \quad $\Pi := \text{set of patterns of length } i$
3. \quad \textbf{while} $\Pi$ \text{ is non empty} \textbf{do}
4. \quad \quad \text{Select } \pi \text{ in } \Pi
5. \quad \quad $\Pi := \Pi \setminus \{\pi\}$
6. \quad \quad \textbf{if} $\text{Post}_{\pi}(W) \subseteq R$ \textbf{then}
7. \quad \quad \quad \text{return} $\langle\pi, \text{True}\rangle$
8. \textbf{return} $\langle, \text{False}\rangle$
References


