Inner approximated reachability analysis

Eric Goubault, Olivier Mullier, Sylvie Putot, Michel Kieffer

Inner Approximated Reachability Analysis

Eric Goubault, Olivier Mullier, Sylvie Putot
CEA LIST, CEA Saclay Nano-INNOV
91 191 Gif-sur-Yvette, France
firstname.lastname@cea.fr

Michel Kieffer
L2S - CNRS - Supélec - Univ Paris-Sud
91192 Gif-sur-Yvette
kieffer@lss.supelec.fr

ABSTRACT

Computing a tight inner approximation of the range of a function over some set is notoriously difficult, way beyond obtaining outer approximations. We propose here a new method to compute a tight inner approximation of the set of reachable states of non-linear dynamical systems on a bounded time interval. This approach involves affine forms and Kaucher arithmetic, plus a number of extra ingredients from set-based methods. An implementation of the method is discussed, and illustrated on representative numerical schemes, discrete-time and continuous-time dynamical systems.

Categories and Subject Descriptors

F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs—*invariants,mechanical verification*; G.1.0 [**Numerical Analysis**]: General—*Interval arithmetic,Numerical algorithms*; G.4 [**Mathematical Software**]: Reliability and robustness

General Terms

Theory, Verification

Keywords

Inner approximation; modal intervals; affine arithmetic

1. INTRODUCTION

Analyzing the reachability of dynamical systems is essential to many areas of computer science, numerical analysis, and control theory. For the validation of computer programs for instance, determining an outer approximation of the set of states that can be reached by a program, can help to prove that it cannot reach erroneous states. On the other hand, inner approximations are useful to prove the reachability of some desired states. Combined, outer and inner approximations provide an indication of the precision of the estimates of the exact reachability region, as shown in [17].

HSCC'14, April 15–17, 2014, Berlin, Germany.

http://dx.doi.org/10.1145/2562059.2562113.

In numerical analysis and control theory, reachability problems using outer and inner approximations are also of primary importance for similar reasons, both for discrete-time and continuous-time dynamical systems [26]. Inner approximations can also be useful for viability problems [2], to prove that there exists a controller for which the system under study behaves in a satisfying way. In the linear case, inner approximations are useful for the design of controllers [13].

In this paper, we propose a method to produce a tractable inner approximation of the set of reachable states of a non-linear dynamical system, at each (discrete or continuous) time step, over a bounded time interval. The main contributions are the following :

- We introduce a generalization of zonotopic abstractions (by vectors of affine forms, as will be recalled in the preliminaries), that produces implicit representations of inner (and outer) approximations (Section 3). The method introduced in [17] can be seen as a special zeroth-order case of the approximation presented here.
- We show how we can extract from these generalized affine vectors, either inner approximations of each component of the vector-valued dynamical system (Section 4.1), using ideas from modal or Kaucher arithmetic, or, if necessary, a joint inner range (Section 4.3). Note that this joint range is more costly but does not have to be computed at each time step.
- Section 5 presents results obtained with our implementation, first for the convergence study of numerical algorithms, where no joint range is necessary, then for discrete and finally continuous-time dynamical systems. The inner approximation for a small hybrid system, requiring the extra ingredient of the interpretation of guard conditions, is finally quickly exemplified.

Related work.

Many methods have been proposed to evaluate outer approximations of reachability sets of linear discrete or continuous-time dynamical systems. They are generally based on interval methods, zonotopes [12], support functions [20], ellipsoids [26], etc. Evaluation of inner approximations has been considered in the linear case in [1], by inner approximating the exponential of a matrix, or using ellipsoidal techniques [26]. Outer approximations of reachability sets have also been obtained for non-linear systems, albeit more recently, e.g., for polynomials systems [7, 29]. Nevertheless, methods to evaluate inner approximations of such sets are

far less developed, since most methods in the non-linear case rely on conservative linearizations, which necessarily produce outer approximations. Under-approximate bounded vertex representation of polyhedra have been proposed for the analysis of Simulink/Stateflow models [24], but they are restricted to linear systems. Hybrid system falsification [28] relies on simulation-based local inner approximations. There exist few methods to compute global inner approximations of the image of non-linear vector-valued functions, mostly based on bisections of the input domain, see for instance [15], later extended by the authors in [16], or inner approximating sets of (semi-algebraic) constraints [21]. But these bisections are very costly if an accurate approximation is needed, and they are not directly applicable to the problem of inner reachability of dynamical systems. For the case of discrete-time dynamical systems for instance, this would require to apply these methods separately to each iterate, with a very costly symbolic representation. To the best of our knowledge, the abstraction described here, generalizing and improving over [17], is the only one to propose such kind of direct inner approximation in a general setting.

2. PRELIMINARIES

Let us first introduce the ingredients that will be instrumental in the computation of inner approximations. In Section 2.1, we formulate the problem of computing an inner or an outer approximation of the image of a function in terms of quantified expressions, for which partial solutions can be given using generalized intervals and Kaucher arithmetic. Section 2.2 introduces affine vectors (also called affine sets in some references [11, 19]) which extend (classical) interval arithmetic to improve the accuracy of outer approximation computations. The rest of the paper mixes these two notions to obtain tight inner approximations in a general setting.

2.1 Generalized intervals for outer and inner approximations

The results and notations quickly introduced in this section are mostly based on the work of Goldsztejn *et al.* on modal intervals [14].

Interval extensions, outer and inner approximations.

Classical intervals are used in many situations to rigorously compute with interval domains instead of reals, usually leading to outer approximations of function ranges over boxes. The set of classical intervals is denoted by $\mathbb{IR} = \{[a,b],\ a \in \mathbb{R}, b \in \mathbb{R}, a \leqslant b\}$. In what follows, intervals are in bold. An outer approximating extension of a function $f : \mathbb{R}^n \to \mathbb{R}$ is a function $\boldsymbol{f} : \mathbb{IR}^n \to \mathbb{IR}$ such that $\forall \boldsymbol{x} \in \mathbb{IR}^n, \text{range}(f, \boldsymbol{x}) = \{f(x), x \in \boldsymbol{x}\} \subseteq \boldsymbol{f}(\boldsymbol{x})$. The natural interval extension consists in replacing real operations by their interval counterparts in the expression of the function. A generally more accurate extension relies on the mean-value theorem, linearizing the function to compute.

Classical interval computations can be interpreted as quantified propositions. Consider for example $f(x) = x^2 - x$. Its natural interval extension, evaluated on $[2,3]$, is $\boldsymbol{f}([2,3]) = [2,3]^2 - [2,3] = [1,7]$, which can be interpreted as the proposition

$$(\forall x \in [2,3])\,(\exists z \in [1,7])\,(f(x) = z).$$

The mean-value extension gives $f(2.5) + \boldsymbol{f}'([2,3]) \times ([2,3] - 2.5) = [1.25, 6.25]$, and can be interpreted similarly.

The drawback of these extensions is that the ranges they yield can be pessimistic, i.e., largely over-estimate the actual range. Inner approximations can be used to evaluate this pessimism, by determining a set of values proved to belong to the range of the function over some input box. The fact that some $\boldsymbol{z} \in \mathbb{IR}$ satisfies $\boldsymbol{z} \subseteq \text{range}(f, \boldsymbol{x})$, i.e., is an inner approximation of the range of f over $\boldsymbol{x}$, can again be written using quantifiers :

$$(\forall z \in \boldsymbol{z})\,(\exists x \in \boldsymbol{x})\,(f(x) = z).$$

Modal intervals and generalized intervals.

A modal interval [9] is an interval supplemented by a quantifier. Extensions of modal intervals were proposed in the framework of generalized intervals, and called AE extensions because universal quantifiers (All) always precede existential ones (Exist) in the interpretations. They give rise to a generalized interval arithmetic which coincides with Kaucher arithmetic [25].

Let us first introduce generalized intervals, i.e., intervals whose bounds are not ordered. The set of generalized intervals is denoted by $\mathbb{IK} = \{[a,b],\ a \in \mathbb{R}, b \in \mathbb{R}\}$. Considering a set of real numbers $\{x \in \mathbb{R},\ a \leqslant x \leqslant b\}$, one can define two generalized intervals, $[a,b]$, which is called *proper*, and $[b,a]$, which is called *improper*. We define the operations dual $[a,b] = [b,a]$ and pro $[a,b] = [\min(a,b), \max(a,b)]$. The generalized intervals are partially ordered by inclusion which extends inclusion of classical intervals. Given two generalized intervals $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, the inclusion is defined by $\boldsymbol{x} \sqsubseteq \boldsymbol{y} \Leftrightarrow \underline{y} \leqslant \underline{x} \wedge \overline{x} \leqslant \overline{y}$. The inclusion is then related to the dual interval by $\boldsymbol{x} \sqsubseteq \boldsymbol{y} \Leftrightarrow \text{dual } \boldsymbol{x} \sqsupseteq \text{dual } \boldsymbol{y}$.

DEFINITION 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $\boldsymbol{x} \in \mathbb{IK}^n$, which we can decompose in $\boldsymbol{x}_\mathcal{A} \in \mathbb{IR}^p$ and $\boldsymbol{x}_\mathcal{E} \in (\text{dual } \mathbb{IR})^q$ with $p + q = n$. A generalized interval $\boldsymbol{z} \in \mathbb{IK}$ is $(f, \boldsymbol{x})$-interpretable if*

$$(\forall x_\mathcal{A} \in \boldsymbol{x}_\mathcal{A})\,(Q_z z \in \text{pro } \boldsymbol{z})\,(\exists x_\mathcal{E} \in \text{pro } \boldsymbol{x}_\mathcal{E}), (f(x) = z) \quad (1)$$

where $Q_z = \exists$ if $\boldsymbol{z}$ is proper, and $Q_z = \forall$ otherwise.

We will later be interested in a generalization of this definition to vector functions $f : \mathbb{R}^n \to \mathbb{R}^p$. In the present context of intervals, we can only consider each component of f independently.

When all intervals in (1) are proper, we retrieve the interpretation of classical interval computation, which gives an outer approximation of $\text{range}(f, \boldsymbol{x})$

$$(\forall x \in \boldsymbol{x})\,(\exists z \in \boldsymbol{z})\,(f(x) = z).$$

When all intervals are improper, (1) becomes an inner approximation of $\text{range}(f, \boldsymbol{x})$

$$(\forall z \in \text{pro } \boldsymbol{z})\,(\exists x \in \text{pro } \boldsymbol{x})\,(f(x) = z).$$

Kaucher arithmetic and the generalized interval natural extension.

Kaucher arithmetic [25] returns intervals that are interpretable as inner approximations in some simple cases. Kaucher addition extends addition on classical intervals by $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$ and $\boldsymbol{x} - \boldsymbol{y} = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$. We now decompose $\mathbb{IK}$ in $\mathcal{P} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \geqslant 0 \wedge \overline{x} \geqslant 0\}$, $-\mathcal{P} = \{\boldsymbol{x} =$

$\boldsymbol{x} \times \boldsymbol{y}$	$\boldsymbol{y} \in \mathcal{P}$	$\mathcal{Z}$	$-\mathcal{P}$	dual$\mathcal{Z}$
$\boldsymbol{x} \in \mathcal{P}$	$[\underline{x}\underline{y}, \overline{x}\overline{y}]$	$[\overline{x}\underline{y}, \overline{x}\overline{y}]$	$[\overline{x}\underline{y}, \underline{x}\overline{y}]$	$[\underline{x}\underline{y}, \underline{x}\overline{y}]$
$\mathcal{Z}$	$[\underline{x}\overline{y}, \overline{x}\overline{y}]$	$[\min(\underline{x}\overline{y}, \overline{x}\underline{y}), \max(\underline{x}\underline{y}, \overline{x}\overline{y})]$	$[\overline{x}\underline{y}, \underline{x}\underline{y}]$	0
$-\mathcal{P}$	$[\underline{x}\overline{y}, \overline{x}\underline{y}]$	$[\underline{x}\overline{y}, \underline{x}\underline{y}]$	$[\overline{x}\overline{y}, \overline{x}\underline{y}]$	$[\overline{x}\overline{y}, \overline{x}\underline{y}]$
dual$\mathcal{Z}$	$[\underline{x}\underline{y}, \overline{x}\underline{y}]$	0	$[\overline{x}\overline{y}, \underline{x}\overline{y}]$	$[\max(\underline{x}\underline{y}, \overline{x}\overline{y}), \min(\underline{x}\overline{y}, \overline{x}\underline{y})]$

Table 1: Kaucher multiplication

$[\underline{x}, \overline{x}]$, $\underline{x} \leqslant 0 \wedge \overline{x} \leqslant 0\}$, $\mathcal{Z} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \leqslant 0 \leqslant \overline{x}\}$, and dual $\mathcal{Z} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \geqslant 0 \geqslant \overline{x}\}$. Kaucher multiplication $\boldsymbol{x} \times \boldsymbol{y}$ is described in Table 1. In Sections 3 and 4, we will have $\boldsymbol{y} = [1, -1]$, belonging to dual $\mathcal{Z}$.

Let us interpret the result of the multiplication $\boldsymbol{z} = \boldsymbol{x} \times \boldsymbol{y}$ in one of the cases encountered when $\boldsymbol{y} \in$ dual $\mathcal{Z}$, for instance for $\boldsymbol{x} \in \mathcal{Z}$. Proposition 1 will express the fact that the result can be interpreted as in Definition 1. Interval $\boldsymbol{z}$ can a priori either be proper or improper, let us consider the improper case. We obtain an inner approximation of the range of the multiplication: according to the quantifiers in Definition 1, computing $\boldsymbol{z} = \boldsymbol{x} \times \boldsymbol{y}$ consists in finding $\boldsymbol{z}$ such that for all $x \in \boldsymbol{x}$, for all $z \in$ pro $\boldsymbol{z}$, there exists $y \in$ pro $\boldsymbol{y}$ such that $z = x \times y$. If $\boldsymbol{x}$ contains zero, which is the case when $\boldsymbol{x} \in \mathcal{Z}$, then $\boldsymbol{z}$ is necessarily 0, the result given in Table 1. Indeed, a property that holds for all $x \in \boldsymbol{x}$, holds in particular for $x = 0$, from which we deduce that for all $z \in$ pro $\boldsymbol{z}$, (there exists $y \in$ pro $\boldsymbol{y}$) $z = 0$.

Kaucher division is defined for all $\boldsymbol{y}$ such that $0 \notin$ pro $\boldsymbol{y}$ by $\boldsymbol{x}/\boldsymbol{y} = \boldsymbol{x} \times [1/\overline{y}, 1/\underline{y}]$.

When restricted to proper intervals, these operations coincide with the classical interval operations. Kaucher arithmetic defines a generalized interval natural extension [14] :

PROPOSITION 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, given by an arithmetic expression with only single occurrences of variables. Then for $\boldsymbol{x} \in \mathbb{IK}^n$, $f(\boldsymbol{x})$, computed using Kaucher arithmetic, is $(f, \boldsymbol{x})$-interpretable.*

Kaucher arithmetic can thus be used in some cases to compute an inner approximation of range$(f, \boldsymbol{x})$. But the restriction to functions f with single occurrences of variables, that is with no dependency, prevents its direct use. A mean-value extension allows us to by-pass this limitation.

Generalized interval mean-value extension.

In the general case of a differentiable function f, the mean-value theorem can be extended to define a generalized interval mean-value extension (see [14]) :

THEOREM 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable, $\boldsymbol{x} \in \mathbb{IK}^n$ and suppose that for each $i \in \{1, \ldots, n\}$, we can compute $\boldsymbol{\Delta}_i \in \mathbb{IR}$ such that*

$$\left\{ \frac{\partial f}{\partial x_i}(x),\ x \in \textit{pro}\ \boldsymbol{x} \right\} \sqsubseteq \boldsymbol{\Delta}_i. \tag{2}$$

Then, for all $\tilde{x} \in$ pro $\boldsymbol{x}$, the following interval is $(f, \boldsymbol{x})$-interpretable :

$$\tilde{f}(\boldsymbol{x}) = f(\tilde{x}) + \sum_{i=1}^{n} \boldsymbol{\Delta}_i (\boldsymbol{x}_i - \tilde{x}_i). \tag{3}$$

EXAMPLE 1. *Let f be defined by $f(x) = x^2 - x$, for which we want to compute an inner approximation of the range over $\boldsymbol{x} = [2, 3]$. Due to the two occurrences of x, $f(\boldsymbol{x})$, computed with Kaucher arithmetic, is not $(f, \boldsymbol{x})$-interpretable. The interval $\tilde{f}(\boldsymbol{x}) = f(2.5) + \boldsymbol{f}'([2, 3])(\boldsymbol{x} - 2.5) = 3.75 + [3, 5](\boldsymbol{x} - 2.5)$ given by its mean-value extension, computed with Kaucher arithmetic, is $(f, \boldsymbol{x})$-interpretable. For $\boldsymbol{x} = [3, 2]$, using the multiplication rule for $\mathcal{P} \times$ dual $\mathcal{Z}$, we get $\tilde{f}(\boldsymbol{x}) = 3.75 + [3, 5]([3, 2] - 2.5) = 3.75 + [3, 5][0.5, -0.5] = 3.75 + [1.5, -1.5] = [5.25, 2.25]$, that can be interpreted as: $\forall z \in [2.25, 5.25]$, $\exists x \in [2, 3]$, $z = f(x)$. Thus, $[2.25, 5.25]$ is an inner approximation of* range$(f, [2, 3])$.

2.2 Affine vectors for outer approximations

Affine arithmetic.

Affine arithmetic [6] is an extension of (classical) interval arithmetic, that takes into account affine correlations between variables. Affine operations are exact in affine arithmetic, so that affine forms are good candidates to define inner approximations, as we will see.

An affine form is a sum over a set of noise symbols ε_i

$$\hat{x} = x_0 + \sum_{i=1}^{n} x_i \varepsilon_i, \tag{4}$$

with $x_i \in \mathbb{R}$ for all i. Each noise symbol ε_i stands for an independent component of the total uncertainty on $\hat{x}$, its value is unknown but bounded in $[-1, 1]$. The same noise symbol can be shared by several variables, expressing correlations between these variables. The set of values represented by an affine form $\hat{x}$ is the box $\left[x_0 - \sum_{i=1}^{n} |x_i|, x_0 + \sum_{i=1}^{n} |x_i|\right]$. Conversely, the assignment of a variable x whose value is given in a range $[a, b]$, is defined as a centered form using a fresh noise symbol $\varepsilon_{n+1} \in [-1, 1]$, which indicates unknown dependency with other variables: $\hat{x} = \frac{(a+b)}{2} + \frac{(b-a)}{2} \varepsilon_{n+1}$.

The result of linear operations on affine forms is an affine form, and is thus interpreted exactly. For two affine forms $\hat{x}$ and $\hat{y}$, and a real number λ, we have $\lambda\hat{x} + \hat{y} = (\lambda x_0 + y_0) + \sum_{i=1}^{n} (\lambda x_i + y_i)\varepsilon_i$.

Affine vectors for outer approximations.

In (classical) affine arithmetic, non-affine operations are linearized, and new noise symbols are introduced to handle the approximation term. In our use, we distinguish, as detailed in [18, 19], these new noise symbols denoted by η_j noise symbols,from the ε_i. The ε_i noise symbols model uncertainty in data or parameters, while the η_j noise symbols model uncertainty coming from the analysis. For instance, a possible (simple) abstraction of the multiplication of two affine forms, defined, for simplicity, on ε_i only, writes

$$\hat{x}\hat{y} = x_0 y_0 + \sum_{i=1}^{n} (x_i y_0 + y_i x_0)\, \varepsilon_i + \frac{1}{2} \sum_{1 \leqslant i,j \leqslant n} |\ x_i y_j + x_j y_i\ |\ \eta_1.$$

More generally, non-affine operations are abstracted by an approximate affine form obtained for instance by a first-order Taylor expansion, plus an approximation term attached to a new noise symbol. Affine operations have linear complexity in the number of noise symbols, whereas non-affine operations can be evaluated with quadratic cost.

EXAMPLE 2. *Consider the arithmetic expressions $x = a * b; y = x + b$, starting from $a \in [-2, 0]$ and $b \in [1, 3]$. The assignments of a and b create two new noise symbols ε_1, ε_2: $\hat{a} = -1 + \varepsilon_1$, $\hat{b} = 2 + \varepsilon_2$. The multiplication produces a new*

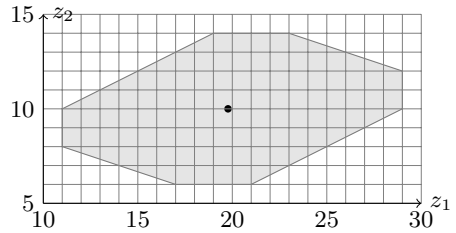

Figure 1: Zonotope $\gamma(Z)$ of Example 3.

η_1 symbol, we get $\hat{x} = -2+2\varepsilon_1-\varepsilon_2+\eta_1$. Affine expressions are handled exactly, we get $\hat{y} = 2\varepsilon_1+\eta_1$. The range of y given by $\hat{y}$ is $[-3,3]$, which is also the exact range, while the range obtained with the natural interval extension is $[-5,3]$.

In what follows, we use matrix notations to handle affine vectors, that is vectors of affine forms. We denote $\mathcal{M}(n,p)$ the space of matrices with n lines and p columns of real coefficients, and ${}^tX$ the transpose of a matrix X.

DEFINITION 2. *An affine vector is a vector of p affine forms over n noise symbols ε_i, $1 \leqslant i \leqslant n$ and m noise symbols η_j, $1 \leqslant j \leqslant m$. It is represented by a matrix $Z \in \mathcal{M}(n+m+1,p)$ decomposed in sub-matrices $Z_0 = (z_{0,k})_{1\leqslant k\leqslant p}$, $Z_\varepsilon = (z_{i,k})_{\substack{1\leqslant k\leqslant p\\ 1\leqslant i\leqslant n}}$ and $Z_\eta = (z_{j,k})_{\substack{1\leqslant k\leqslant p\\ n+1\leqslant j\leqslant n+m}}$. The set of values it represents is the zonotope*

$$\gamma(Z) = \left\{ {}^tZ_0 + {}^tZ_\varepsilon\varepsilon + {}^tZ_\eta\eta \mid (\varepsilon,\eta) \in [-1,1]^{n+m} \right\}. \qquad (5)$$

In Definition 2, the k-th component of the vector is given by the affine form $\hat{z}_k = z_{0,k} + \sum_{i=1}^{n} z_{i,k}\varepsilon_i + \sum_{j=n+1}^{n+m} z_{j,k}\eta_j$.

EXAMPLE 3. *For $n=4$ and $p=2$, the set of values represented by the affine vector $(\hat{z}_1,\hat{z}_2)$ with $\hat{z}_1 = 20-4\varepsilon_1+2\varepsilon_3+3\varepsilon_4$, and $\hat{z}_2 = 10-2\varepsilon_1+\varepsilon_2-\varepsilon_4$, is the gray zonotope of Figure 1. Of course, the range of each variable considered independently can also be computed: $\gamma(\hat{z}_1) = [20-4-2-3, 20+4+2+3] = [11,29]$ and $\gamma(\hat{z}_2) = [6,14]$.*

The affine vector introduced in Definition 2 can also be seen as a linear function of the n inputs or noise symbols to the p variables it represents, plus an uncertain part given as a linear transform of a box (the zonotope ${}^tZ_\eta\eta$). It represents a set of functions from $\mathbb{R}^n$ to $\mathbb{R}^p$:

$$\gamma_{\text{func}}(Z) = \left\{ g_Z : \mathbb{R}^n \to \mathbb{R}^p \mid \begin{array}{l} \forall \varepsilon \in [-1,1]^n, \exists \eta \in [-1,1]^m, \\ g_Z(\varepsilon) = {}^tZ_0 + {}^tZ_\varepsilon\varepsilon + {}^tZ_\eta\eta \end{array} \right\}$$

This will be instrumental in the definition of outer and inner approximations, by allowing to state Properties 1 and 2. Property 1 states that affine vectors are interpretable as over-approximations.

PROPERTY 1. *Let $f : \mathbb{R}^n \to \mathbb{R}^p$ be a function, $\boldsymbol{x} \in \mathbb{IR}^n$, and $Z \in \mathcal{M}(n+m+1,p)$ the abstraction in affine vectors of the p components of $f(x), x \in \boldsymbol{x}$. Then Z is $(f,\boldsymbol{x})$-interpretable :*

$$(\forall x \in \boldsymbol{x})\,(\exists z \in \gamma(Z)),\ (f(x) = z).$$

Equivalently, using the interpretation of affine vectors as sets of functions, Z is $(f,\boldsymbol{\varepsilon})$-interpretable :

$$(\forall \varepsilon \in \boldsymbol{\varepsilon})\,(\exists \eta \in \boldsymbol{\eta}),\ (f(x(\varepsilon)) = {}^tZ_0 + {}^tZ_\varepsilon\varepsilon + {}^tZ_\eta\eta).$$

In Definition 1, an *interval* is considered (f,x)-interpretable. In Property 1, Definition 1 is implicitly extended to zonotopic sets of values z, for vector-valued functions f. However, we will not be able to obtain sets that are (f,x)-interpretable in the same way for inner approximation.

Constrained affine vectors.

As described in [11], we can interpret conditions in these affine vectors by adding some constraints on the noise symbols ε_i. Instead of letting them vary freely into $[-1,1]$, we restrain ourselves to inputs that satisfy these constraints. This idea allows us to interpret conditions in programs, or guard conditions for hybrid systems, for outer approximations as well as inner approximations: we do not detail this in this paper, but quickly illustrate this on a simple example in Section 5.5.

3. GENERALIZED AFFINE VECTORS

We consider again the problem of finding an $(f,\boldsymbol{x})$ interpretable set as in Definition 1, but with the affine arithmetic point-of-view.

To each component $\boldsymbol{x}_i$, $i = 1,\dots,n$ of the input box $\boldsymbol{x} \in \mathbb{IK}^n$, we associate a noise symbol ε_i, by writing $\hat{x}_i(\varepsilon_i) = \frac{\underline{x}_i+\overline{x}_i}{2} + \frac{\overline{x}_i-\underline{x}_i}{2}\varepsilon_i$, where $\boldsymbol{x}_i = [\underline{x}_i, \overline{x}_i]$. Then any function $f : \mathbb{R}^n \to \mathbb{R}$, for some input $\boldsymbol{x} \in \mathbb{IK}^n$, can be seen as a function f^ε of the vector $\varepsilon = (\varepsilon_1,\dots,\varepsilon_n)$. f^ε is said to be the function induced on $\varepsilon = (\varepsilon_1,\dots,\varepsilon_n)$ by the substitution of $\hat{x}_i(\varepsilon_i)$, $i = 1,\dots,n$ in f.

As already mentionned in [17], we can now restate the generalized mean-value extension of Theorem 1 on f^ε.

THEOREM 2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, $\boldsymbol{x} \in \mathbb{IK}^n$, and $f^\varepsilon : \mathbb{R}^n \to \mathbb{R}$ the function induced on $\varepsilon = (\varepsilon_1,\dots,\varepsilon_n)$. Suppose that $\boldsymbol{\Delta}_i$ is an outer approximation of the partial derivative $\frac{\partial f^\varepsilon}{\partial \varepsilon_i}$:*

$$\left\{ \frac{\partial f^\varepsilon}{\partial \varepsilon_i}(\varepsilon),\ \varepsilon \in [-1,1]^n \right\} \sqsubseteq \boldsymbol{\Delta}_i. \qquad (6)$$

Then, $\forall (t_1,\dots,t_n) \in$ pro $\boldsymbol{\varepsilon} = [-1,1]^n$,

$$\tilde{f^\varepsilon}(\boldsymbol{\varepsilon}_1,\dots,\boldsymbol{\varepsilon}_n) = f^\varepsilon(t_1,\dots,t_n) + \sum_{i=1}^{n} \boldsymbol{\Delta}_i(\boldsymbol{\varepsilon}_i - t_i), \qquad (7)$$

is $(f,\boldsymbol{x})$-interpretable. In particular,

- *if $\tilde{f^\varepsilon}([1,-1]^n)$, computed with Kaucher arithmetic, is improper, then pro $\tilde{f^\varepsilon}([1,-1]^n)$ is an inner approximation of $\{f^\varepsilon(\varepsilon),\ \varepsilon \in [-1,1]^n\} = \{f(x),\ x \in \boldsymbol{x}\}$.*
- *if $\tilde{f^\varepsilon}([-1,1]^n)$ is a proper interval, then it is an outer approximation of $\{f(x),\ x \in \boldsymbol{x}\}$.*

Theorem 2 [17] allows us to compute outer and inner approximating intervals of the ranges of expressions. But we also have more than just ranges, as we define generalized affine forms over the noise symbols ε_i, generalized in the sense that multiplicative coefficients of the noise symbols are no longer just real numbers but represent sets of values. We will characterize the joint inner approximation defined by these forms in Section 4.

Affine vectors with interval coefficients, which we refer to as zeroth-order generalized affine vectors, are obtained by bounding the partial derivatives in intervals $\boldsymbol{\Delta}_i$. This is

what was proposed in [17], and will not be detailed here. Operations on these zeroth-order sets involve interval computations, and thus suffer from the drawbacks of interval arithmetic.

Affine vectors with affine vector coefficients are obtained when an outer approximation of the Jacobian matrix of the function is computed using affine vectors. They are called first-order generalized affine vectors, and are introduced in the next section. We will compare results of the zeroth and first-order sets in Section 5.

3.1 First-order generalized affine vectors

In this section, we start by defining the first-order generalized affine vectors and the property we expect them to satisfy (Property 2) to be able to use them for inner approximation. We then explicit the construction of such sets.

DEFINITION 3. *A first-order generalized affine vector from $\mathbb{R}^n$ to $\mathbb{R}^p$ is a triple (Z, c, J) consisting of an affine vector $Z \in \mathcal{M}(n+m+1, p)$, a vector $c \in \mathbb{R}^p$, and a vector of affine vectors $J \in (\mathcal{M}(n,p))^n$.*

In order to use Theorem 2 to derive an inner approximation of range$(f, \boldsymbol{x})$ from these first-order generalized affine vectors, we want them to satisfy the following property.

PROPERTY 2. *A first-order generalized affine vector (Z, c, J) abstracts the function $f : \mathbb{R}^n \to \mathbb{R}^p$, if $c = f^\varepsilon(0)$ and*

$$(\forall \varepsilon \in \boldsymbol{\varepsilon})\,(\exists \eta \in \boldsymbol{\eta}),\ \begin{cases} f^\varepsilon(\varepsilon) = {}^tZ_0 + {}^tZ_\varepsilon \varepsilon + {}^tZ_\eta \eta \\ \frac{\partial f^\varepsilon}{\partial \varepsilon_i}(\varepsilon) = {}^tJ_{i,0} + {}^tJ_{i,\varepsilon}\varepsilon + {}^tJ_{i,\eta}\eta, \\ \forall i = 1, \ldots, n \end{cases} \quad (8)$$

Equation 8 expresses that for a given $\varepsilon \in [-1,1]^n$, (Z, c, J) defines an outer approximation of $f^\varepsilon(\varepsilon)$ and of $(\frac{\partial f^\varepsilon}{\partial \varepsilon_i})_i(\varepsilon)$ relying on the same parametrization in the η noise symbols.

We thus now define arithmetic operations that preserve Property 2, starting from a generalized affine vector defined as in Definition 3, where to each component $\boldsymbol{x}_i$ of the input box $\boldsymbol{x}$, corresponds a noise symbol ε_i, $i = 1, \ldots, n$. Note that the k-th component of the vector represented by the affine vector Z, is given by the affine form $\hat{z}^k = z_{0,k} + \sum_{i=1}^{n} z_{i,k}\varepsilon_i + \sum_{j=n+1}^{n+m} z_{j,k}\eta_j$. Similarly, the k-th component of the affine vector J_i, noted $\hat{j}_{ik}$, is a vector that represents an affine form that outer approximates the component $\frac{\partial f_k^\varepsilon}{\partial \varepsilon_i}$. In the following, for a more compact definition of operations, we will see the affine vector matrix Z as its equivalent vector of affine forms $\hat{z}_k$, $1 \leqslant k \leqslant p$, and the vector of affine vectors J as its equivalent matrix of affine forms $\hat{j}_{i,k}$, $1 \leqslant i \leqslant n, 1 \leqslant k \leqslant p$.

The following example will illustrate the arithmetic operations on first-order generalized affine vectors.

EXAMPLE 4. *Let $x = (x_1, x_2) \in [2,3] \times [3,4]$ and*

$$f(x) = \begin{pmatrix} x_1^3 - 2x_1x_2 \\ x_2^3 - 2x_1x_2 \end{pmatrix}$$

Assignment.

The generalized affine vector $(Z', c', J') \in \mathcal{M}(n+m+1, p+1) \times \mathbb{R}^{p+1} \times (\mathcal{M}(n, p+1))^n$ for $f' : \mathbb{R}^n \to \mathbb{R}^{p+1}$ where $f_{p+1} := [a, b]$, with $a < b$ and corresponding noise symbol ε_i, is defined by :

$$\begin{cases} Z' = \begin{pmatrix} Z & \frac{a+b}{2} + \frac{b-a}{2}\varepsilon_i \end{pmatrix} \\ c' = \begin{pmatrix} c & \frac{a+b}{2} \end{pmatrix} \\ J' = \begin{pmatrix} & 0 & \\ J & \frac{b-a}{2} & \\ & 0 & \end{pmatrix} \leftarrow i\text{-th line} \end{cases}$$

EXAMPLE 5. *In Example 4, let us interpret the assignments $x_1 := [2,3]$ and $x_2 := [3,4]$. The affine forms for x_1 and x_2 are $\hat{x}_1 = \frac{5}{2} + \frac{1}{2}\varepsilon_1$ and $\hat{x}_2 = \frac{7}{2} + \frac{1}{2}\varepsilon_2$. The centers are $c_1 = \frac{5}{2}$ and $c_2 = \frac{7}{2}$. Finally, the Jacobian of the function which associates (x_1, x_2) to $(\varepsilon_1, \varepsilon_2)$ is $J = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$.*

Affine operations.

Affine operations are handled exactly; we will examplify them later on the function of Example 4, at the same time as multiplication. For $(\lambda_1, \lambda_2) \in \mathbb{R}^2$, the generalized affine vector $(Z', c', J') \in \mathcal{M}(n+m+1, p+1) \times \mathbb{R}^{p+1} \times (\mathcal{M}(n, p+1))^n$ for $f' : \mathbb{R}^n \to \mathbb{R}^{p+1}$ where $f_{p+1} = \lambda_1 f_i + \lambda_2 f_j$, is defined by :

$$\begin{cases} Z' = \begin{pmatrix} Z & \lambda_1\hat{z}_i + \lambda_2\hat{z}_j \end{pmatrix} \\ c' = \begin{pmatrix} c & \lambda_1 c_i + \lambda_2 c_j \end{pmatrix} \\ J' = \begin{pmatrix} & \lambda_1\hat{j}_{1,i} + \lambda_2\hat{j}_{1,j} & \\ J & \vdots & \\ & \lambda_1\hat{j}_{n,i} + \lambda_2\hat{j}_{n,j} & \end{pmatrix} \end{cases}$$

Multiplication.

The generalized affine vector $(Z', c', J') \in \mathcal{M}(n+m+2, p+1) \times \mathbb{R}^{p+1} \times (\mathcal{M}(n, p+1)^n$ for $f' : \mathbb{R}^n \to \mathbb{R}^{p+1}$ where $f_{p+1} = f_i \times f_j$, is defined by :

$$\begin{cases} Z' = \begin{pmatrix} Z & \hat{z}_i\hat{z}_j \end{pmatrix} \\ c' = \begin{pmatrix} c & c_i c_j \end{pmatrix} \\ J' = \begin{pmatrix} & \hat{z}_j\hat{j}_{1,i} + \hat{z}_i\hat{j}_{1,j} & \\ J & \vdots & \\ & \hat{z}_j\hat{j}_{n,i} + \hat{z}_i\hat{j}_{n,j} & \end{pmatrix} \end{cases}$$

EXAMPLE 6. *In Example 4, one needs to compute $\hat{x}_1\hat{x}_2$, $\hat{x}_1^3$ and $\hat{x}_2^3$. First, we get, using rules from Section 2.2, $\hat{x}_1\hat{x}_2 = \frac{35}{4} + \frac{7}{4}\varepsilon_1 + \frac{5}{4}\varepsilon_2 + \frac{1}{4}\eta_1$. This adds a (third) column to Z and a new center, $c_3 = \frac{35}{4}$. Moreover,*

$$\begin{array}{rcl} \frac{\partial(\hat{x}_1\hat{x}_2)}{\partial\varepsilon_1} & = & \frac{\partial\hat{x}_1}{\partial\varepsilon_1}\hat{x}_2 + \hat{x}_1\frac{\partial\hat{x}_2}{\partial\varepsilon_1} \\ & = & \frac{7}{4} + \frac{1}{4}\varepsilon_2 \end{array}$$

Similarly, $\frac{\partial(\hat{x}_1\hat{x}_2)}{\partial\varepsilon_2} = \frac{5}{4} + \frac{1}{4}\varepsilon_1$. This adds a (third) column to J: ${}^t\begin{pmatrix} \frac{7}{4} + \frac{1}{4}\varepsilon_2 & \frac{5}{4} + \frac{1}{4}\varepsilon_1 \end{pmatrix}$.
Then we compute successively $\hat{x}_1\hat{x}_1$, $\hat{x}_1(\hat{x}_1\hat{x}_1)$, $\hat{x}_2\hat{x}_2$, $\hat{x}_2(\hat{x}_2\hat{x}_2)$ adding each time a new column to Z, a new center and a new column to J (computing the outer approximations of the partial derivatives of these expressions along the ε_j). One gets

$$\begin{array}{rcl} \hat{x}_1^3 - 2\hat{x}_1\hat{x}_2 & = & -\frac{25}{16} + \frac{95}{16}\varepsilon_1 + \frac{17}{8}\eta_3 \\ \hat{x}_2^3 - 2\hat{x}_1\hat{x}_2 & = & \frac{427}{16} - \frac{7}{4}\varepsilon_1 + \frac{255}{16}\varepsilon_2 + \frac{15}{8}\eta_5 \end{array}$$

with the centers $c_1 = -\frac{15}{8}$ *and* $c_2 = \frac{203}{8}$*, and the following last two columns of the Jacobian* J

$$\begin{pmatrix} \frac{97}{16} + \frac{15}{4}\varepsilon_1 - \frac{1}{2}\varepsilon_2 + \frac{15}{8}\eta_6 & -\frac{5}{2} - \frac{1}{2}\varepsilon_1 \\ -\frac{7}{2} - \frac{1}{2}\varepsilon_2 & \frac{257}{16} - \frac{1}{2}\varepsilon_1 + \frac{21}{4}\varepsilon_2 + \frac{15}{8}\eta_7 \end{pmatrix}$$

The fact that the arithmetic operations defined above preserve Property 2 results from the property that operations on (classical) affine vectors outer approximate the concrete operations [18], combined with the rules of derivation of sum and product of functions.

4. INTERVAL AND JOINT INNER RANGE OF GENERALIZED AFFINE VECTORS

This section describes the information we can derive from the generalized affine vectors on the inner range of vector-valued functions, first component-by-component, then considering components jointly.

4.1 Interval inner approximation of the range

From a first-order generalized affine vector (Z, c, J) abstracting a function $f : \mathbb{R}^n \to \mathbb{R}^p$, a zeroth-order generalized affine vector is easily obtained as

$$\check{x}_k = c_k + \sum_{i=1}^{n} [a_{ik}, b_{ik}]\varepsilon_i, \ \forall k = 1, \ldots, p,$$

where $[a_{ik}, b_{ik}]$ is the interval concretization of the affine form $\hat{j}_{ik}$.

We can then define the following (inner) interval concretization for each variable x_k, $k = 1, \ldots, p$:

$$\underline{\gamma}(\check{x}_k) \quad = \quad \text{pro}\ (c_k + \textstyle\sum_{i=1}^{n} [a_{ik}, b_{ik}] \times [1, -1])$$

computed using Kaucher arithmetic, see Section 2.1. As noted in [17], the improper interval $[1, -1]$ being in dual $\mathcal{Z}$, the types of (proper) intervals $[a_{ik}, b_{ik}]$ that do not lead to a multiplication equal to zero can be deduced from Table 1. It must be in $\mathcal{P}$ or in $-\mathcal{P}$, that is the interval bounding the Jacobian coefficient should not contain zero.

By Theorem 2, as the intervals $[a_{ik}, b_{ik}]$ outer approximate the partial derivative of the kth component of f^{ε} with respect to ε_i, $\underline{\gamma}(\check{x}_k)$ is guaranteed to be inside the set of reachable values for x_k, i.e., of the k-th projection of the image of f.

EXAMPLE 7. *With the first-order inner approximating set of Example 6, we get the following concretization in terms of inner approximating forms of order zero :*

$$\begin{array}{rcl} \check{x}_1 & = & -1.875 + [-0.0625, 12.1875]\varepsilon_1 + [-3, -2]\varepsilon_2 \\ \check{x}_2 & = & 25.375 + [-4, -3]\varepsilon_1 + [8.4375, 23.6875]\varepsilon_2 \end{array}$$

and the interval concretizations using Kaucher arithmetic: $\underline{\gamma}(\check{x}_1)$ = *pro* $(-1.875 + [-0.0625, 12.1875] \times [1, -1] + [-3, -2] \times [1, -1])$ = *pro* $(-1.875 + [2, -2]) = [-3.875, 0.125]$ *and* $\underline{\gamma}(\check{x}_2)$ = *pro* $(25.375 + [3, -3] + [8.437, -8.437]) = [13.937, 36.812]$.

On this example, we get better interval concretizations with the direct computation of zeroth-order forms of [17],

$$\begin{array}{rcl} \check{x}'_1 & = & -1.875 + [2, 10.5]\varepsilon_1 + [-3, -2]\varepsilon_2 \\ \check{x}'_2 & = & 25.375 + [-4, -3]\varepsilon_1 + [10.5, 22]\varepsilon_2 \end{array}$$

that give $\underline{\gamma}(\check{x}'_1) = [-5.875, 2.125]$ *and* $\underline{\gamma}(\check{x}'_2) = [11.875, 38.875]$. *There is no general rule about the relative precision of the interval concretizations of zeroth-order and first-order generalized affine forms. Nevertheless, the joint range will be better with first-order forms. This result is similar to that obtained when comparing interval arithmetic to affine arithmetic. When the considered function* f *is more involved, see e.g., Section 5, interval concretizations of first-order generalized affine forms are usually much more precise than the ones obtained with zeroth-order generalized affine forms.*

The next section describes a tool for the inner approximation of vector-valued functions, which will allow us in Section 4.3 to characterize an inner approximation of the joint inner range of first-order generalized affine forms.

4.2 Inner range of vector-valued functions

This section recalls the main result of [15] to evaluate an inner approximation of the range of a function with domain and co-domain of the same dimension. We refer the reader to [16] for the extension of this method to functions from $\mathbb{R}^n$ to $\mathbb{R}^p$ with p not necessarily equal to n.

THEOREM 3. *(Corollary 3.1 of [15]) Let* $\mathbf{x} \in \mathbb{IR}^n$ *and* $f : \mathbf{x} \to \mathbb{R}^n$ *be a continuous function, continuously differentiable in the interior of* $\mathbf{x}$, $\text{int}(\mathbf{x})$. *Consider* $\mathbf{y} \in \mathbb{IR}^n$ *and* $\tilde{x} \in \mathbf{x}$ *such that* $f(\tilde{x}) \in \mathbf{y}$. *Consider also an interval matrix* $\mathbf{J} \in \mathbb{IR}^{n \times n}$ *such that* $f'(x) \in \mathbf{J}$ *for all* $x \in \mathbf{x}$. *Assume that* $0 \notin \mathbf{J}_{ii}$ *for all* $i \in [1, \ldots, n]$. *Calling* $\text{Diag}\,\mathbf{J}$ *the diagonal part of* $\mathbf{J}$ *and* $\text{OffDiag}\,\mathbf{J}$ *its off diagonal part, consider*

$$H(\mathbf{J}, \tilde{x}, \mathbf{x}, \mathbf{y}) = \tilde{x} + (\text{Diag}^{-1}\,\mathbf{J})\Big(\mathbf{y} - f(\tilde{x}) - (\text{OffDiag}\,\mathbf{J})(\mathbf{x} - \tilde{x})\Big). \tag{9}$$

If $H(\mathbf{J}, \tilde{x}, \mathbf{x}, \mathbf{y}) \sqsubseteq \text{int}(\mathbf{x})$ *then* $\mathbf{y} \sqsubseteq \text{range}(f, \mathbf{x})$.

This theorem provides an efficient test for a box $\mathbf{y}$ to be a subset of the range of a vector-valued function, see Figure 2 for an illustration. The restriction on f having the same dimension of domain and co-domain comes from the matrix inversion of $\text{Diag}\,\mathbf{J}$ in (9).

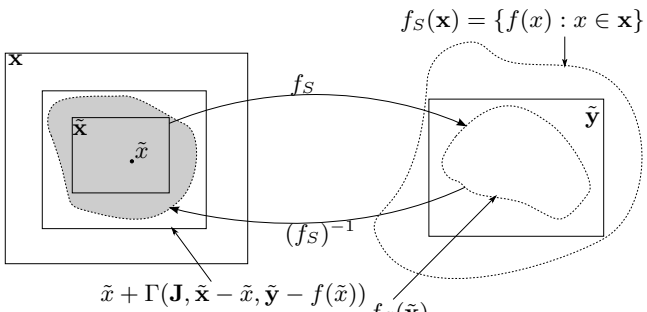

Figure 2: Sets and functions involved in Corollary 3 for inner approximation.

The algorithm relying on Corollary 3 to obtain an inner approximation of range$(f, \boldsymbol{x})$ is as follows. The algorithm starts with a function f, a box $\mathbf{x} \in \mathbb{IR}^n$ on which an inner approximation of the range of f must be evaluated, and a parameter ϵ, which will determine the precision for the inner approximation of range$(f, \boldsymbol{x})$. We start by using an outer approximation of the image of f on $\mathbf{x}$, called $\mathbf{y}$ (using interval analysis, and the mean-value theorem as in Section 2.1). If condition (9) on $\mathbf{x}$ and $\mathbf{y}$ is satisfied, then $\mathbf{y}$ is in the image of f. Otherwise, we bisect $\mathbf{y}$ and carry on testing condition (9) on each of the generated sub-boxes, until they are proven to be in range$(f, \boldsymbol{x})$, or their width is less than ϵ, so as to ensure termination of the algorithm.

Hence this method gives a paving of boxes proven to be inside the image of f on $\mathbf{x}$. We illustrate this on the particular case of a vector-valued function given by a generalized affine vector in the next section.

4.3 Inner range of generalized affine vectors

Corollary 3 gives a criterion to prove that boxes belong to range$(f, \boldsymbol{x})$, by only evaluating an outer approximation of the Jacobian of f on sub-boxes of $\mathbf{x}$. Both zeroth-order and first-order generalized affine vectors compute an outer approximation of this Jacobian. A joint concretization Γ is thus calculated using the algorithm presented in Section 4.2, using the center $c_1, \ldots, c_p$ for $f(\tilde{x})$ and J in place of the exact Jacobian of f.

EXAMPLE 8. *The joint concretization of the first-order generalized affine vector for Example 4 is represented in Figure 3 and compared to the exact range. The joint concretization for the zeroth-order generalized affine vector only contains the point* $(-1.875, 25.375)$ *here, because it does not track dependencies between the coefficients of the Jacobian.*

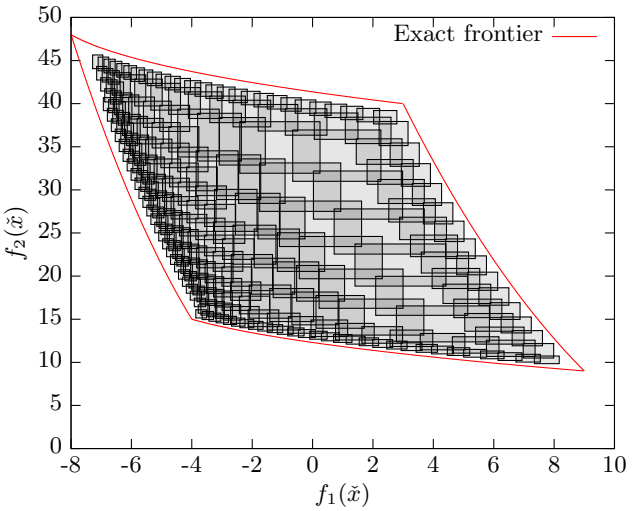

Figure 3: Inner concretization of first-order affine vector and exact function border for Example 8: 269 proved inner boxes in 0.03s with ϵ=0.1.

5. EXPERIMENTS

5.1 Implementation

A C library[1] has been written, that implements both zeroth-order and first-order generalized affine vectors; these abstractions are interfaced to the Apron [23] library of abstract domains, and rely on the Taylor1+ [10] implementation for the outer approximating affine vectors.

In the previous sections, we relied on real numbers to compute the generalized affine vectors. In the implementation, one important point is to ensure guaranteed results while using finite-precision numbers. We use the multi-precision floating-point library MPFR [8] which allows us to increase precision if necessary. To get a sound and tractable implementation, we make sure the outer approximation of the variables and the Jacobian of the function by affine vectors or intervals is sound with respect to finite-precision. For intervals, this is obvious using outward rounding (rounding towards $-\infty$ for the first bound and towards $+\infty$ for the second bound). For affine vectors, this can also be achieved quite easily, see [10] for instance. Then, the center c_k of the generalized affine form for a variable x_k, can be soundly computed by a small interval with outward rounding, $[\underline{c_k}, \overline{c_k}]$. Finally, we want a sound interval inner approximation using Kaucher arithmetic

$$\underline{\gamma}(\tilde{x}_k) = \text{pro}\ ([\underline{c_k}, \overline{c_k}] + \sum_{j=1}^{n} [a_{ik}, b_{ik}] \times [1, -1]),$$

$[\underline{c_k}, \overline{c_k}]$ is a proper interval while each $[a_{ik}, b_{ik}] \times [1, -1]$ is an improper interval. We get an inner approximation of the range of x_k if the addition of these generalized intervals is an improper interval $\boldsymbol{r}$: computing this addition again with outward rounding ensures correctness: indeed if the finite precision approximation $\boldsymbol{r}_p$ of $\boldsymbol{r}$, is such that $\boldsymbol{r} \sqsubseteq \boldsymbol{r}_p$, then dual $(\boldsymbol{r}_p) \sqsubseteq$ dual $(\boldsymbol{r})$, that is we obtain a smaller (thus correct) inner approximation than the one that would be computed with real numbers.

Note that, in the extreme case, if too much precision is lost in the computation due to the use of finite precision compared to the width of the inner approximation given by $[a_{ik}, b_{ik}] \times [1, -1]$, then $[\underline{c_k}, \overline{c_k}] + \sum_{j=1}^{n} [a_{ik}, b_{ik}] \times [1, -1]$ may become proper, so that we no longer get any inner approximation.

5.2 Convergence of numerical schemes

Inner approximations are useful to state properties of numerical algorithms, as shown below.

5.2.1 A Newton iteration

We consider the (non-linear) iteration of the Newton algorithm $x(k+1) = 2x(k) - ax(k)^2$, for $a \in [1.95, 2.]$. If we take $x(0)$ not too far away from the inverse of a, this iteration converges to $1/a$. We start here from $x(0) = 0.6$.

Figure 4 represents 10 iterates of this Newton algorithm, computed with the outer approximating affine vectors (Taylor1+ [10] Apron implementation), and the zeroth-order and first-order inner approximating affine vectors, all with double precision. While the zeroth-order inner approximation quickly tends to a unique point, the first-order inner aproximation remains very close to the outer approximation (actually so close we do not see the difference on the figure). Let us now show how we can usefully combine the information from the inner and outer approximation on this simple scheme. If we ask to iterate this scheme until $|x(k+1) - x(k)| < 5.\,10^{-4}$, we can prove, thanks to the outer approximation, that the stopping criterion of the loop is always satisfied after 4 iterations (we have $|x(4) - x(3)| \subseteq [-2.6\,10^{-4}, 2.6\,10^{-4}]$). While the inner approximation of $x(k+1) - x(k)$ proves that there exist some inputs for which the criterion is not satisfied for the first 3 iterations (for instance, $[-7.7\,10^{-4}, -4.1\,10^{-4}] \subseteq x(3) - x(2)$). The inner and outer approximation can be used to prove that when the criterion is satisfied, $[.4999244, .5127338] \subseteq x(4) \subseteq [0.499831, 0.512906]$, which is actually quite tight.

[1] available at `http://www.lix.polytechnique.fr/Labo/Sylvie.Putot/hscc14.html`

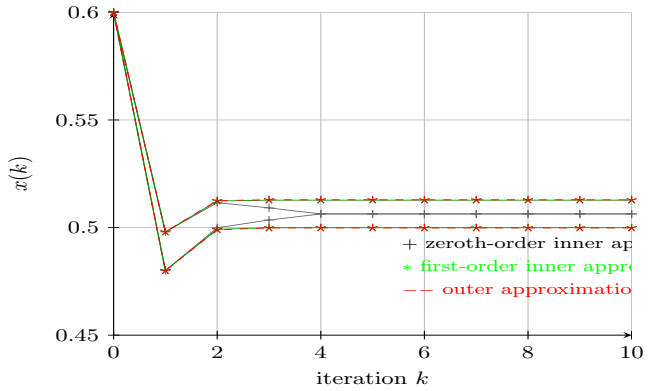

Figure 4: Inner and outer approximations for the Newton iterates

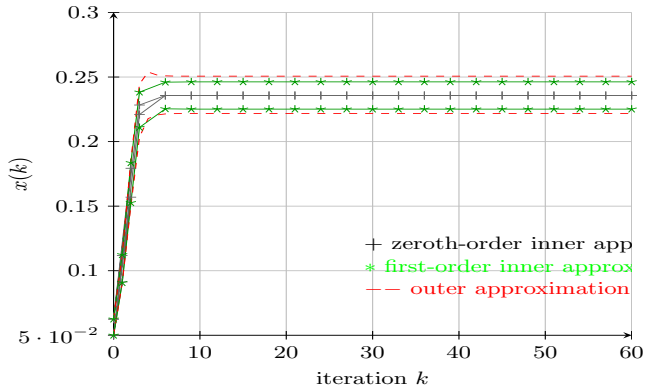

Figure 5: Inner and outer approximations for the Householder iterates

5.2.2 *A Householder iteration*

Consider the following Householder scheme

$$x(k+1) = x(k) + x(k)\left(\frac{1}{2}h(k) + \frac{3}{8}h(k)^2\right)$$

with $h(k) = 1 - ax(k)^2$ and $a \in [16, 20]$, starting from $x(0) = [\frac{1}{20}, \frac{1}{16}]$. The results by inner and outer approximation are presented Figure 5. It is even clearer here than on the Newton example that the zeroth-order inner approximation is not accurate enough for such a non-linear scheme, while the first-order inner approximation manages to remain stable along iterations and not far from the outer approximation.

We represent in Figure 6 the execution times for the three methods: not surprisingly, the outer approximation is the fastest, as an inner approximation needs the evaluation of an outer approximation. The cost of the zeroth-order inner approximation remains close to that of the outer approximation. The first-order inner approximation is naturally more costly, as it involves outer approximation of every component of the Jacobian matrix of the function composed of every elementary sub-expressions involved in the scheme (the inner approximation is built inductively on the arithmetic expressions, in order to be automatically computed on any program). As expected, the cost remains almost linear compared to the cost of the outer approximation. Note that a study of the cost and behaviour of this outer approximation

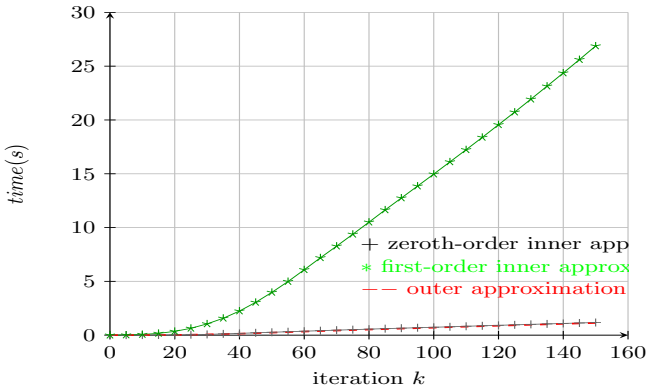

Figure 6: Comparison of execution times for the Householder scheme

was reported in [10], where its good tradeoff between cost and accuracy was demonstrated.

5.3 Reachability of discrete dynamical systems: FitzHugh-Nagumo neuron model

This polynomial discrete-time dynamical system is derived from a continuous-time dynamical system modeling the electrical activity of a neuron, using an Euler time-discretization scheme, see [29]:

$$\begin{cases} x_1(k+1) = x_1(k) + h\left(x_1(k) - \frac{x_1(k)^3}{3} - x_2(k) + I\right) \\ x_2(k+1) = x_2(k) + h\left(0.08(x_1(k) + 0.7 - 0.8x_2(k))\right) \end{cases}$$

where $h = 0.2$, $I = \frac{7}{8}$, and the initial set is the bounding box $[1, 1.25] \times [2.25, 2.5]$

Using first-order affine vectors, we obtain for instance, at iteration 100 (in 11.54 seconds), the inner and outer ranges : $[-.737783, -.716137] \subseteq x_1 \subseteq [-.857537, -.595651]$, and $[.450016, .506109] \subseteq x_2 \subseteq [.429873, .542796]$. Figure 7 represents both approximations, for the 100 iterations: note that we do not present the joint range defined in Section 4.3, but only the interval ranges in both coordinates: at each iteration, each interval corresponding to the two coordinates for the inner approximating boxes (in plain lines) is guaranteed to be in the reachability set for each of the two variables x_1 and x_2. The boxes themselves are not guaranteed to be within the reachability set. Of course, the outer approximating boxes (in dotted lines), that enclose the zonotopes that were actually computed, are guaranteed to be an outer approximation of the reachable values. We see that, even for 100 iterations of a non-trivial non-linear dynamical system, we get outer and inner approximations which are quite close to each other, demonstrating the quality of the analysis performed - the reader may also compare these results with the very similar corresponding figure in [29].

5.4 Reachability of ODEs: Brusselator

Our inner approximation basically relies on a calculation of an outer approximation of a Jacobian (plus a center). For ODEs, it is simple to derive an ODE that gives the evolution over time of each entry of the Jacobian of the solution with respect to the initial conditions. We can then use any method (here, Taylor models) to outer approximate solutions of this derived system of ODEs, and use the result as a starting point for our inner approximation.

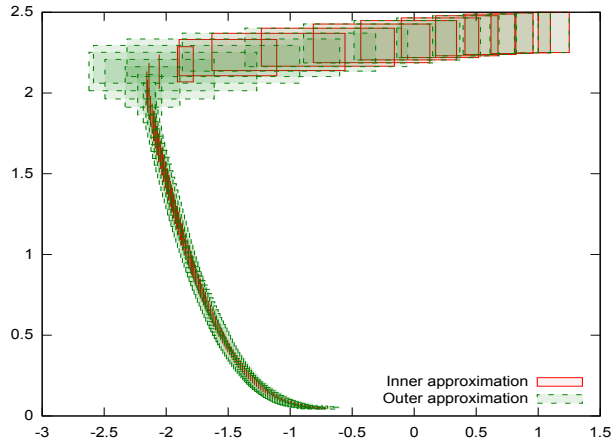

Figure 7: Outer and inner approximations of reachable sets for 100 iterations of the FitzHugh-Nagumo model

This is now detailed on the generic system

$$\dot{x}_i = f_i(x_1, \ldots, x_n) \text{ for } i = 1, \ldots, n \tag{10}$$

with initial condition $x(0) = (x_1(0), \ldots, x_n(0)) \in \mathbf{x_0} \sqsubseteq \mathbb{R}^n$. Under some regularity conditions on the functions f_i, $i = 1, \ldots, n$, the Cauchy-Lipschitz theorem asserts the existence of a unique solution $g : \mathbb{R}^{n+1} \to \mathbb{R}^n$ to (10) on a time interval $[0, \tau]$, for some positive τ. We could derive from set-integration methods (see, e.g., Lohner's method [27]), a method that would directly use our inner approximating arithmetic, in the style of what has been done in [3] using zonotopes. This needs too many developments for an inclusion in this paper, we choose instead to use a Taylor model of the dynamics [4] and of the dynamics of its Jacobian with respect to initial conditions.

When the functions f_i in (10) are at least continuously differentiable in all their arguments, we can write down a new "derived" system of ordinary differential equations describing the evolution in time of a solution g of (10), and its Jacobian $J_g = \left(g_{i,j} = \frac{\partial g_i}{\partial x_j(0)}\right)$. The functions $g_{i,j}$ and g_l satisfy the following systems of ordinary differential equations:

$$\left\{ \begin{array}{rcl} \dot{y}_{i,j} & = & \sum_{k=1}^{n} \frac{\partial f_i}{\partial x_k} y_{k,j} \\ \dot{x}_l & = & f_l(x_1, \ldots, x_n) \end{array} \right.$$

for $i = 1, \ldots, n$, $j = 1, \ldots, n$ and $\ell = 1, \ldots, n$, with $y_{i,i}(0) = 1$ and $y_{i,j}(0) = 0$ for $i \neq j$. Using Taylor models, the derived ODE is now modeled by polynomials P_k in $x_1(0), \ldots, x_n(0)$ and in t (the current time), and a box remainder $\mathbf{I}_k$ for a time interval $[t_k, t_{k+1}]$, $k = 0, \ldots, l$, $t_0 = 0$. These polynomials and interval remainders are such that $P_k([t_k, t_{k+1}]) + \mathbf{I}_k$ are guaranteed to contain the solutions to the derived ODE (hence the Jacobian of the solutions of the initial ODE) for all initial conditions within $\mathbf{x_0}$. We can now use our outer approximation methods using affine arithmetic to get an inner approximation in terms of first-order generalized affine forms.

Example 9. *Consider the Brusselator studied e.g. in [3], [4]:*

$$\left\{ \begin{array}{rcl} \dot{x}_1 & = & 1 + x_1^2 x_2 - 2.5 x_1 \\ \dot{x}_2 & = & 1.5 x_1 - x_1^2 x_2 \end{array} \right.$$

with $x_1(0) \in [0.9, 1]$ *and* $x_2(0) \in [0, 0.1]$.

We use the tool Flow [5] to derive the Taylor models of order 5, up to time* $t = 4$*, with fixed steps equal to 0.1 (hence we get 40 Taylor models) and remainder estimation parameter equal to 0.1 (we will get a rather coarse estimate of the flowpipe, given that we allow for quite large box remainders) for the derived system of ODEs.*

We now look at the inner approximation derived from the zonotopic outer approximations of the Taylor model $(P_{39}, \mathbf{I}_{39})$*, i.e., describing the solutions for the Brusselator in the time interval* $[3.9s, 4s]$*. As an indication, the Taylor model derived by Flow* for* x_1*, within the latter time interval is a fifth-order polynomial in the initial conditions of the ODE and in time* t*, composed of 56 monomials. We find* $(x_1, x_2) \in [0.700684, 0.763468] \times [1.851165, 1.894451]$ *with centers* $c_1 = 0.732$ *and* $c_2 = 1.873$*, and the Jacobian is outer approximated by forms which are within the interval matrix:*

$$\frac{1}{20} \begin{pmatrix} [0.1347, 0.1624] & [0.2427, 0.2963] \\ [-0.0049, 0.1091] & [0.03129, 0.2039] \end{pmatrix}$$

Hence an inner approximation of the range of x_1 *and* x_2 *are respectively pro* $(0.732 + \frac{1}{20}[0.1347, 0.1624] \times [1, -1] + \frac{1}{20}[0.2427, 0.2963] \times [1, -1]) = [0.7132, 0.751]$ *and pro* $(1.873 + \frac{1}{20}[-0.0049, 0.1091] \times [1, -1] + \frac{1}{20}[0.03129, 0.2039] \times [1, -1]) = [1.87124, 1.87437]$*. We see that for* x_1*, we get a tight inner approximation with respect to the outer approximation.*

5.5 Guard conditions in hybrid automata

We briefly illustrate in this section how to interpret in our framework guard conditions in dynamical systems, such as the ones defining mode changes in hybrid automata. We concentrate here on a particular case, exemplified below. The general case is out of the scope of this paper.

We consider a simple 1D mass-spring system, where the mass decreases linearly over time, until some minimum mass is reached. As a hybrid automaton, this can be modeled by two modes m_1 and m_2 with transition from m_1 to m_2 if $k > k_m$ and from m_2 to m_1 if $k \leqslant k_m$. Each of the modes is normally governed by an ODE, but to keep things simple, we discretize them using a simple Euler scheme, with time constant $h = 0.04$. In all modes $x(n+1) = x(n) - hk(n)(x(n) - x_c)$, but in m_1, $k(n+1) = k(n) - hg_k$ $(k(n+1) > k_m)$, and in m_2, $k(n+1) = k(n)$ $(k(n+1) \leqslant k_m)$. The initial values for k and x are $k(0) \in [2, 2.5]$ (i.e. $k(0) = 2.25 + 0.25\varepsilon_1$), $x(0) \in [10, 11]$ (i.e. $x(0) = 10.5 + 0.5\varepsilon_2$), and constants $g_k = 2$, $k_m = 1$, $x_c = 8$.

The system starts in mode m_1, and some states can change mode only from iterate 13 on, as a simple analysis by our inner and outer approximations show. At iterate 13, the first-order generalized affine form (which coincides here with the outer approximation) of $k(13)$ is $1.21 + 0.25\varepsilon_1$. Therefore the states that will stay in mode m_1 at iterate 13 are exactly those for which $1.21 + 0.25\varepsilon_1 > 1$, i.e. $\varepsilon_1 > -0.84$, that is for initial state $k(0) \in [2.04, 2.5]$. At iterate 13, the Jacobian of $x(13)$ is outer approximated by: $J = (-0.006650 - 0.0009165\varepsilon_1 - 0.002021\varepsilon_2 + 0.0729218\eta_1 \;\; 0.3863 - 0.05388\varepsilon_2 + 0.001809\eta_4)$. The outer approximation of $x(13)$ is $7.7003 - 0.005278\varepsilon_1 - 0.4234\varepsilon_2 + 1.3824\eta_3$. It proves that $x(13) \in [5.8891, 9.5114]$ if we ignore the mode change, and if not, that is if we take into account $\varepsilon_1 > -0.84$, we get a slightly tighter value: $x(13) \in [5.8891, 9.5106]$. From the first-order generalized affine form, we also deduce that the inner interval range for $x(13)$ is in all cases $[7.369694594, 8.030894409]$:

this is still a fairly wide inner approximation even in the case of the mode change.

In this example, we have been able to handle the guard condition exactly as a restriction of the values that ε_1 can take: $1 \geq \varepsilon_1 > -0.84$ instead of $1 \geq \varepsilon_1 > -1$. In the general case, the guard condition will be expressed as a set of inequalities involving several noise symbols, for which we will have to compute inner boxes, or joint interval inner approximations, for instance using the work of Isshii *et al.* [22] on an interval-based projection method for under-constrained systems, that relies on similar ideas as described in Section 4.2 for the computation of boxes guaranteed to be in the image of a vector-valued function.

6. CONCLUSION

The method we developed for inner approximating reachable sets of dynamical systems can still be improved in several directions. First, we can directly integrate the first-order generalized affine forms arithmetic in the Picard operator approach to solving continuous-time ODEs. This would relieve us from the preliminary step of obtaining a Taylor model of the ODE. Second, we can use more general set-based methods for representing the Jacobian we need at each step of our method, in particular, we would like to investigate the use of higher Taylor methods.

7. ACKNOWLEDGEMENTS

This work was partly supported by the Digiteo SANSCRIT and ANR-12-INSE-0007-02 CAFEIN projects.

Michel Kieffer is partly supported by the Institut Universitaire de France, 75005 Paris.

8. REFERENCES

[1] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of linear systems with uncertain parameters and inputs. In *IEEE CDC*, 2007.

[2] J. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhäuser, Boston, 1990.

[3] O. Bouissou, A. Chapoutot, and A. Djoudi. Enclosing temporal evolution of dynamical systems using numerical methods. In *NASA Formal Methods*, 2013.

[4] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *RTSS*, pages 183–192, 2012.

[5] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *CAV*, pages 258–263, 2013.

[6] J. L. D. Comba and J. Stolfi. Affine arithmetic and its applications to computer graphics. *SIBGRAPI*, 1993.

[7] T. Dang and R. Testylier. Hybridization domain construction using curvature estimation. In *HSCC*, pages 123–132, 2011.

[8] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. Mpfr: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw*, 33(2):13, 2007.

[9] E. Gardeñes, M. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, and A. Trepat. Model intervals. *Reliable Computing*, 7(2):77–111, 2001.

[10] K. Ghorbal, E. Goubault, and S. Putot. The zonotope abstract domain taylor1+. In *CAV'09*, volume 5643 of *LNCS*, pages 627–633. Springer, 2009.

[11] K. Ghorbal, E. Goubault, and S. Putot. A logical product approach to zonotope intersection. In *CAV'10*, volume 6174 of *LNCS*, 2010.

[12] A. Girard. Reachability of uncertain linear systems using zonotopes. In *HSCC'05*. Springer, 2005.

[13] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *HSCC 2006*, volume 3927 of *LNCS*, pages 257–271. Springer, 2006.

[14] A. Goldsztejn, D. Daney, M. Rueher, and P. Taillibert. Modal intervals revisited: a mean-value extension to generalized intervals. In *QCP'05*, 2005.

[15] A. Goldsztejn, L. Jaulin, et al. Inner approximation of the range of vector-valued functions. *Reliable Computing*, 14:1–23, 2010.

[16] E. Goubault, M. Kieffer, O. Mullier, and S. Putot. General inner approximation of vector-valued functions. *Reliable Computing*, 18:117–143, 2013.

[17] E. Goubault and S. Putot. Under-approximations of computations in real numbers based on generalized affine arithmetic. In *SAS*, pages 137–152, 2007.

[18] E. Goubault and S. Putot. A zonotopic framework for functional abstractions. *CoRR*, abs/0910.1763, 2009.

[19] E. Goubault, S. Putot, and F. Vedrine. Modular static analysis with zonotopes. In *SAS'12*, volume 7460 of *LNCS*, pages 24–40. Springer, 2012.

[20] C. L. Guernic and A. Girard. Reachability analysis of hybrid systems using support functions. In *CAV*, 2009.

[21] D. Henrion and C. Louembet. Convex inner approximations of nonconvex semialgebraic sets applied to fixed-order controller design. *International Journal of Control*, 85(8):1083–1092, 2012.

[22] D. Ishii, A. Goldsztejn, and C. Jermann. Interval-based projection method for under-constrained numerical systems. *Constraints*, 17(4):432–460, 2012.

[23] B. Jeannet and A. Miné. Apron: A library of numerical abstract domains for static analysis. In *CAV'09*, pages 661–667. Springer, 2009.

[24] A. Kanade, R. Alur, F. Ivančić, S. Ramesh, S. Sankaranarayanan, and K. C. Shashidhar. Generating and analyzing symbolic traces of simulink/stateflow models. In *CAV'09*. Springer, 2009.

[25] E. Kaucher. Interval analysis in the extended interval space IR. *Comput. (Supplementum) 2*, 1980.

[26] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *HSCC '00*, pages 202–214. Springer, 2000.

[27] R. J. Lohner. Enclosing the solutions of ordinary initial and boundary value problems. In *Computer Arithmetic: Scientific Computation and Programming Languages*, pages 255–286. Wiley-Teubner, 1987.

[28] T. Nghiem, S. Sankaranarayanan, G. Fainekos, F. Ivancić, A. Gupta, and G. J. Pappas. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. In *HSCC'10*. ACM, 2010.

[29] M. A. B. Sassi, R. Testylier, T. Dang, and A. Girard. Reachability analysis of polynomial systems using linear programming relaxations. In *ATVA*, 2012.