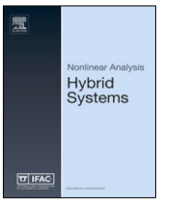
A higher order method for input-affine uncertain systems

Nonlinear Analysis: Hybrid Systems 47 (2023) 101266

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs



A higher order method for input-affine uncertain systems

Sanja Zivanovic Gonzalez [a,*], Luca Geretti [b], Davide Bresolin [d], Tiziano Villa [b], Pieter Collins [c]

[a] Barry University, Miami Shores, FL, USA
[b] Università di Verona, Verona, Italy
[c] Maastricht University, Maastricht, The Netherlands
[d] Università di Padova, Padova, Italy

ARTICLE INFO

Article history:
Received 10 December 2021
Received in revised form 20 June 2022
Accepted 18 August 2022
Available online 24 September 2022

Keywords:
Differential inclusions
Nonlinear systems
Rigorous numerics

ABSTRACT

Uncertainty is unavoidable in modeling dynamical systems and it may be represented mathematically by differential inclusions. In the past, we proposed an algorithm to compute validated solutions of differential inclusions; here we provide several theoretical improvements to the algorithm, including its extension to piecewise constant and sinusoidal approximations of uncertain inputs, updates on the affine approximation bounds and a generalized formula for the analytical error. The approach proposed is able to achieve higher order convergence with respect to the current state-of-the-art. We implemented the methodology in Ariadne, a library for the verification of continuous and hybrid systems. For evaluation purposes, we introduce ten systems from the literature, with varying degrees of nonlinearity, number of variables and uncertain inputs. The results are hereby compared with two state-of-the-art approaches to time-varying uncertainties in nonlinear systems.

1. Introduction

In this paper we present a method for computing rigorous solutions of uncertain nonlinear dynamical systems in continuous-time, which is an important sub-problem in the verification of uncertain nonlinear hybrid systems. Uncertainty in the system arises due to environmental disturbances and modeling discrepancies. The former include input and output disturbances, and noise on sensors and actuators; the latter account for the unavoidable approximation of a model with respect to the real system due to unmodeled phenomena, order reduction and parameter variations over changes of the environment and variations over time of the modeled system. These forms of uncertainty and imprecision may be appropriately modeled by differential inclusions.

Differential inclusions are a generalization of differential equations having multivalued right-hand sides

$$\dot{x}(t) \in F(x(t)), \;\; x(0) = x_0, \tag{1}$$

see [1–3]. As well as being an important class of uncertain system models, they can also be used to model differential equations with discontinuities, by taking the closed convex hull of the right-hand side as proposed by Filippov [4], Even more importantly, they arise from the analysis of complex or large-scale systems using model-order reduction or

* Corresponding author.
E-mail addresses: szivanovic@barry.edu (S. Zivanovic Gonzalez), luca.geretti@univr.it (L. Geretti), davide.bresolin@unipd.it (D. Bresolin), tiziano.villa@univr.it (T. Villa), pieter.collins@maastrichtuniversity.nl (P. Collins).

https://doi.org/10.1016/j.nahs.2022.101266
1751-570X/

compositional analysis. When applying model-order reduction techniques to replace a high-order system of differential equations $\dot{x} = f(x)$ by a low-order system, the differential inclusion form $\dot{z} \in h(z) + [-\epsilon, \epsilon]$ captures the unmodeled influences in the reduced-order system. When components of a complex system depend on one another, we can decouple them by replacing inputs from other components with noise that varies over the range of possible values, resulting in smaller but uncertain subsystems (see [5]). Ability to identify these subsystems becomes essential in compositional analysis of complex systems represented by hybrid automata (see [6]).

Another important application area for differential inclusions is control theory. Assume a control system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \tag{2}$$

where $u(t) \in U$ is not exactly known. Then, one may need to compute reachable sets corresponding to all admissible inputs which, under certain assumptions, is equivalent to computing the reachable set of a differential inclusion, see Theorem 1. A recent book [7] gives more insight on the application of differential inclusions in control theory.

One of the first algorithms for obtaining solution sets of a differential inclusion was given in [8,9]. In [8] they used viability kernels and in [9] they considered Lipschitz differential inclusions, giving a polyhedral method for obtaining an approximation of the solution set to an arbitrary known accuracy. In the case where F is only upper-semicontinuous with compact, convex values, it is possible to compute arbitrarily accurate over-approximations to the solution set, as shown in [10].

In recent years, the focus of approximating reachable set shifted to providing rigorous solutions, i.e. over-approximations of the solution set, and several algorithms have been proposed. Interval Taylor models were used in [11,12]; exponential enclosure technique was used in [13]; an algorithm based on comparison theorems was given in [14]; support vector machines were used in [15]; a Lohner-type algorithm was used in [16,17]; conservative linearization was used in [18]; a set-oriented method in [19], and polynomialization was used in [20,21].

In [22] ellipsoidal enclosures were developed to provide inner and outer (thick) enclosures to the reachable sets of uncertain systems. This technique involves temporal series expansion with assumption that uncertain input is piece-wise constant. Nevertheless, for comparison purposes, more suitable are [20,21], and [12], since [20] provides convergence analysis, [21] is higher-order method, and [12] uses the same function calculus as our method (Taylor models). However, only [12,21] are implemented in state-of-the-art tools similar to the tool we use, Ariadne, i.e. CORA and Flow*, respectively.

Finding correct balance between speed and accuracy is a challenging issue that depends on the application domain. While for online applications speed is crucial, accuracy may be a matter of life and death in cases such as a robot performing laser incision on a patient, see [23–25], or simply be critical for cost and effectiveness as in generic robotic automation [26]. As noted in these papers, the model of the system takes the form of a hybrid system, consisting of a discrete control part that operates in a continuous environment whose dynamics exhibits uncertainties.

Regarding the application to hybrid systems, there are three main components in the analysis of their dynamics: applying the discrete transitions, computing the continuous behavior, and resolving the guard conditions governing the interaction between the continuous and discrete dynamics. When extending the capabilities of a reachability analysis tool from deterministic to nondeterministic systems, applying the discrete transitions is relatively straightforward: one simply parametrizes the set of possible successor states, as one would parametrize a set of initial states. Similarly, resolving the guard conditions involves the introduction of an additional parameter for the crossing time. In both cases, the resulting set remains finite dimensional. However, when computing the continuous behavior, since the disturbance is time-dependent, the set of disturbances is infinite-dimensional, so a direct parametrization is infeasible. For this reason, developing methods computing the continuous behavior is the critical part of the extension, and requires additional theory. It is this theory which we address in this paper.

Our objective in particular is to provide an over-approximation of the reachable set of an input-affine differential inclusion of the form

$$\dot{x}(t) = f(x(t)) + \sum_{i=1}^{m} g_i(x(t))v_i(t); \quad x(t_0) = x_0, \tag{3}$$

where $x : \mathbb{R} \to \mathbb{R}^n$, $v_i(\cdot) \in [-V_i, V_i]$ is a bounded measurable function for $i = 1, \ldots, m$ and $V_i > 0$ for all i. Our method focuses on creating an auxiliary system

$$\dot{y}(t) = f(y(t)) + \sum_{i=1}^{m} g_i(y(t))w_i(t); \quad y(t_0) = y_0, \quad w(\cdot) \in W, \tag{4}$$

by finding appropriate functions $w(t)$ and a set W, such that the difference between solutions $\|x(t) - y(t)\|$ is as small as desired.

Numerical results given in this paper were obtained using the function calculus implemented in Ariadne [27], a tool for reachability analysis and verification of cyber physical systems. In particular, we use *Taylor models* for the rigorous approximation of continuous functions. A Taylor model expresses approximations to a function in the form of a polynomial (defined over a suitably small domain) plus an interval remainder, see [28]. While *Taylor model* calculus already provides us with over-approximations when performing calculations such as antiderivation, direct application of it to the system

(3) is not possible since $v(\cdot)$ belongs to an infinite dimensional space. Instead, we propose to define an auxiliary system, whose time-varying inputs are finitely parameterized, and to which we can apply *Taylor model* calculus to obtain over-approximations, compute the difference between the two systems, and add this difference (analytical error) to achieve an over-approximation of the reachable set. Moreover, we desire to achieve third-order error in a single step approximation.

In our previous papers [29,30], the algorithm for obtaining an over-approximation in such a way was presented, derivation in the one-dimensional additive case with its corresponding error formula was given, cases of affine, step and sinusoidal auxiliary functions were revealed and some computational results were showcased. Here, we provide full derivation of the local error for a general input-affine system and extract formulas for the error in several cases. Namely, we present errors of $O(h)$, $O(h^2)$, and $O(h^3)$ explicitly with suitable $w(t)$. Formulas for the local error are obtained based on Lipschitz constants, logarithmic norm and bounds on higher-order derivatives. Computational results are more thorough providing insights on dependency on the simplification and noise levels. In particular, we demonstrate efficiency and accuracy of our algorithm by testing ten nonlinear systems of different sizes and inputs. Comparison of reachable sets is made with the ones produced by Flow* and CORA.

The paper is organized as follows. In Section 2, we give key ingredients of the theory used. In Section 3, we give a mathematical setting for obtaining over-approximations of the reachable sets of input-affine differential inclusions; we derive the local error, and we give formulas for obtaining the error of second and third orders. Implementation aspects are presented in Section 4 and numerical testing of the algorithm and its comparison to other tools is presented in Section 5. Some proofs and numerical results were left out of this paper for space reasons: the interested reader can refer to the arXiv document [31] for additional details.

2. Preliminaries

By a solution of (1), given F is a continuous set-valued map with compact and convex values, we mean an absolutely continuous function $x : [0, T] \to \mathbb{R}^n$ such that, for almost all $t \in [0, T]$, $x(\cdot)$ is differentiable at t and $\dot{x}(t) \in F(x(t))$. The solution set $S_T(x_0) \subset C([0, T], \mathbb{R}^n)$ is defined as

$$S_T(x_0) = \{x(\cdot) \in C([0, T], \mathbb{R}^n) \mid x(\cdot) \text{ is a solution of (1)}\}.$$

The *reachable set* at time t, $R(x_0, t) \subset \mathbb{R}^n$, is defined as

$$R(x_0, t) = \{x(t) \in \mathbb{R}^n \,|x(\cdot) \in S_t(x_0)\}.$$

The following theorem states conditions under which the solution sets of control system and differential inclusion coincide.

Theorem 1. *Let $f : X \times U \to X$ be continuous where U is a compact separable metric space and assume that there exists an interval I and an absolutely continuous $x : I \to \mathbb{R}^n$, such that for almost all $t \in I$,*

$$\dot{x}(t) \in f(x(t), U).$$

Then there exists a Lebesgue measurable $u : I \to U$ such that for almost all $t \in I$, $x(\cdot)$ satisfies

$$\dot{x}(t) = f(x(t), u(t)).$$

The theorem and the proof can be found in [1, Corollary 1.14.1]. For further work on the theory of differential inclusions see [1–3].

Here, we use the supremum norm for the vector norm in $\mathbb{R}^n$, i.e., for $x \in \mathbb{R}^n$, $\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$. We often use abbreviation $\text{conv}\{x(t), y(t)\}$ to denote convex full between $x(t)$ and $y(t)$. Specifically, we mean a line between $x(t)$ and $y(t)$ for each $t \in [a, b]$, i.e. all $z(t) = sx(t) + (1 - s)y(t)$, $s \in [0, 1]$ and all $t \in [a, b]$. For a function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ the norm used is $\|f\|_\infty = \sup_{x \in D} \|f(x)\|_\infty$.

The corresponding matrix norm instead is

$$\|Q\|_\infty = \max_{k=1,\ldots,n} \left\{ \sum_{i=1}^{n} |q_{ki}| \right\}.$$

Given a square matrix Q and a matrix norm $\|\cdot\|$, the *logarithmic norm* is defined by

$$\lambda(Q) = \lim_{h \to 0^+} \frac{\|I + hQ\| - 1}{h}.$$

There are explicit formulas for the logarithmic norm for several matrix norms, see [32,33]. The formula for the logarithmic norm corresponding to the matrix norm we use is

$$\lambda_\infty(Q) = \max_k \{q_{kk} + \sum_{i \neq k} |q_{ki}|\}.$$

Logarithmic norm was introduced independently in [32], and [34] in order to derive error estimates to initial value problems, see also [35]. Using the logarithmic norm is advantageous over the use of the Lipschitz constant since it can

have negative values, and thus, one can distinguish between forward and reverse time integration, and stable and unstable systems. We, then, use the following theorem to give an estimate between a solution of a differential equation and an almost solution.

Theorem 2. *Let $x(t)$ satisfy the differential equation $\dot{x}(t) = f(t, x(t))$ with $x(t_0) = x_0$, where f is Lipschitz continuous. Suppose that there exist functions $l(t)$, $\delta(t)$ and ρ such that $\lambda(Df(t, z(t))) \leq l(t)$ ($Df(\cdot)$ denotes the Jacobian matrix) for all $z(t) \in \text{conv}\{x(t), y(t)\}$ and $\|\dot{y}(t) - f(t, y(t))\| \leq \delta(t)$, $\|x(t_0) - y(t_0)\| \leq \rho$. Then for $t \geq t_0$ we have*

$$\|y(t) - x(t)\| \leq e^{\int_{t_0}^{t} l(s)ds} \left(\rho + \int_{t_0}^{t} e^{-\int_{t_0}^{s} l(r)dr} \delta(s)ds \right).$$

The theorem is presented in [33].

Numerical computations of reachable sets of time-varying systems require a rigorous way of computing with sets and functions in Euclidean space. A suitable calculus is given by the *Taylor models* defined in [28].

Definition 1. Let $f : D \subset \mathbb{R}^v \to \mathbb{R}$ be a function that is $(n+1)$ times continuously partially differentiable on an open set containing the domain D. Let x_0 be a point in D and P the nth order Taylor polynomial of f around x_0. Let I be an interval such that

$$f(x) - P(x - x_0) \in I \text{ for all } x \in D$$

Then the pair (P, I) is an nth order Taylor model of f around x_0 on D.

A full description of Taylor models as used in Ariadne is given in [36].

3. Analytical error

In this section, we present theoretical grounds for construction of the auxiliary system presented in Eq. (4) with formulas for the local error on the difference between the true system and its auxiliary counterpart.

Let $[0, T]$ be an interval of existence of (3). Let $0 = t_0, t_1, \ldots, t_{n-1}, t_n = T$ be a partition of $[0, T]$, and let $h_k = t_{k+1} - t_k$. For $x \in \mathbb{R}^n$ and $v(\cdot) \in L^\infty([t_k, t_{k+1}]; \mathbb{R}^m)$, define $\phi(x_k, v(\cdot)) = x(t_{k+1})$ which is the solution of (3) at time t_{k+1} with $x(t_k) = x_k$. At each time step we want to compute an over-approximation R_{k+1} to the set

$$\text{reach}(R_k, t_k, t_{k+1}) = \{\phi(x_k, v(\cdot)) \mid x_k \in R_k \text{ and } v(\cdot) \in L^\infty([t_k, t_{k+1}]; \mathbb{R}^m)\},$$

where $L^\infty([t_k, t_{k+1}]; \mathbb{R}^m)$ is the space of essentially bounded measurable functions from interval $[t_k, t_{k+1}]$ into $\mathbb{R}^m$, i.e., the functions are bounded except on a set of measure zero. Since L^∞ is infinite-dimensional, we aim to approximate the set of all solutions by restricting the disturbances to a finite-dimensional space by creating an auxiliary system (Eq. (4)). Let a set of functions $W_k \subset C([t_k, t_{k+1}]; \mathbb{R}^m)$ be parameterized as $W_k = \{w(a_k, \cdot) \mid a_k \in A \subset \mathbb{R}^p\}$. For example, W_k can be a set of all linear functions of the form $w(a_k, t) = a_{0k} + a_{1k}t$. We then need to find an error bound ϵ_k such that

$$\forall\, v_k \in L^\infty([t_k, t_{k+1}]; V),\ \exists\, a_k \in A \text{ s.t. } \|\phi(x_k, v_k(\cdot)) - \phi(x_k, w(a_k, \cdot))\| \leq \epsilon_k. \tag{5}$$

Note that we do not need to find explicitly infinitely many a_k's. Instead we need to choose the correct dimension ($\mathbb{R}^p$) and provide bounds to get a desired error ϵ_k.

3.1. Error derivation

Let

- $f : \mathbb{R}^n \to \mathbb{R}^n$ is a C^p function,
- each $g_i : \mathbb{R}^n \to \mathbb{R}^n$ is a C^p function,
- each $v_i(\cdot)$ is a measurable function such that $v_i(t) \in [-V_i, +V_i]$ for some $V_i > 0$.

Here, $p \geq 1$ depends on the desired order and will be precisely defined later. We write for simplicity $w_i(t) = w_i(a_k, t)$ $i = 1, \ldots, m$ and assume that they are continuously differentiable real-valued functions. The error representing the difference between the exact solution and the auxiliary system is derived mostly using integration by parts until a desired order (e.g., $O(h^3)$) is achieved. In what follows, Df denotes the Jacobian matrix, $D^2 f$ denotes the Hessian matrix, and $\lambda(\cdot)$ denotes the logarithmic norm of a matrix defined in Section 2. For convenience of notation, we write $h_k = t_{k+1} - t_k$, $t_{k+1/2} = t_k + h_k/2 = (t_k + t_{k+1})/2$, and $\hat{q}(t) = \int_{t_k}^{t} q(s)\, ds$.

The one-step error in the difference between x_{k+1} and y_{k+1} is derived as follows. Writing (3) and (4) as integral equations, we obtain:

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} f(x(t)) + \sum_{i=1}^{m} g_i(x(t))v_i(t)\,dt; \tag{6a}$$

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(y(t)) + \sum_{i=1}^{m} g_i(y(t))w_i(t)\,dt. \tag{6b}$$

Without loss of generality, we assume that $x(t_k) = y(t_k)$ for all $k \geq 0$. To be precise, initially, we assume $x(t_0) = y(t_0)$. After obtaining an over-approximation R_1 to the solution set at time t_1, we use R_1 as the set of initial points of both the original system (3) and the auxiliary one (4) for the next time step. Thus we have $x(t_1) = y(t_1) \in R_1$. We compute R_2, and consider it to be the set of initial points for both equations at time t_2. Proceeding like this, we have $x(t_k) = y(t_k)$, for all $k \geq 0$. Therefore, the difference between the two systems in (6) becomes

$$x(t_{k+1}) - y(t_{k+1}) = \int_{t_k}^{t_{k+1}} f(x(t)) - f(y(t))\,dt \tag{7a}$$

$$+ \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} g_i(x(t))v_i(t) - g_i(y(t))w_i(t)\,dt. \tag{7b}$$

Integrating by parts the term (7a), we obtain

$$\begin{aligned}
(7a) \;&= \Big[(t - t_{k+1/2})\big(f(x(t)) - f(y(t))\big)\Big]_{t_k}^{t_{k+1}} \\
&\quad - \int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\frac{d}{dt}\big(f(x(t)) - f(y(t))\big)dt \\
&= (h_k/2)\big(f(x(t_{k+1})) - f(y(t_{k+1}))\big) \\
&\quad - \int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\big(Df(x(t))\dot{x}(t) - Df(y(t))\dot{y}(t)\big)dt.
\end{aligned}$$

There are two ways that we deal with term (7b). First we rewrite the term inside the integral as

$$g_i(x(t))v_i(t) - g_i(y(t))w_i(t) = (g_i(x(t)) - g_i(y(t)))\,w_i(t) + g_i(x(t))\,(v_i(t) - w_i(t)),$$

and then integrate by parts the second term to obtain

$$\begin{aligned}
(7b) &= \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} (g_i(x(t)) - g_i(y(t)))\,w_i(t)\,dt \\
&\quad + \sum_{i=1}^{m} \Big[g_i(x(t))(\hat{v}_i(t) - \hat{w}_i(t))\Big]_{t_k}^{t_{k+1}} - \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} \frac{d}{dt}\Big(g_i(x(t))\Big)\,(\hat{v}_i(t) - \hat{w}_i(t))\,dt \\
&= \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} (g_i(x(t)) - g_i(y(t)))\,w_i(t)\,dt && (8a) \\
&\quad + \sum_{i=1}^{m} g_i(x(t_{k+1}))(\hat{v}_i(t_{k+1}) - \hat{w}_i(t_{k+1})) && (8b) \\
&\quad - \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} Dg_i(x(t))\,\dot{x}(t)\,(\hat{v}_i(t) - \hat{w}_i(t))\,dt && (8c)
\end{aligned}$$

The second derivation is obtained just by integrating by parts,

$$\begin{aligned}
(7b) &= \sum_{i=1}^{m} \Big[g_i(x(t))\hat{v}_i(t) - g_i(y(t))\hat{w}_i(t)\Big]_{t_k}^{t_{k+1}} \\
&\quad - \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} \frac{d}{dt}\Big(g_i(x(t))\Big)\hat{v}_i(t) - \frac{d}{dt}\Big(g_i(y(t))\Big)\hat{w}_i(t)\;dt \\
&= \sum_{i=1}^{m} g_i(x(t_{k+1}))\hat{v}_i(t_{k+1}) - g_i(y(t_{k+1}))\hat{w}_i(t_{k+1}) && (9a) \\
&\quad - \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} Dg_i(x(t))\hat{v}_i(t)\dot{x}(t) - Dg_i(y(t))\hat{w}_i(t)\dot{y}(t)\;dt && (9b)
\end{aligned}$$

Eqs. (8) and (7a) can be used to derive second-order local error estimates. By applying the mean value theorem we obtain

$$f(x(t_{k+1})) - f(y(t_{k+1})) = \int_0^1 Df(z(s))ds\ \big(x(t_{k+1}) - y(t_{k+1})\big).$$

Here, $z(s) = x + sh$ is a line between x and $x + h$ (for $x, x + h \in V$, $L(x; x + h) \subseteq V$), Df denotes Jacobian matrix of f, and integration is understood component-wise. Hence,

$$\begin{aligned}
(7a)\ &= (h_k/2) \int_0^1 Df(z(s))ds\ \big(x(t_{k+1}) - y(t_{k+1})\big) && (10a)\\
&\quad - \int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\big(Df(x(t))\dot{x}(t) - Df(y(t))\dot{y}(t)\big)\, dt. && (10b)
\end{aligned}$$

Separate the second part of the integrand in (10b) as

$$\begin{aligned}
Df(x(t))\,\dot{x}(t) - Df(y(t))\,\dot{y}(t) &= Df(x(t))\,\big(\dot{x}(t) - \dot{y}(t)\big) && (11a)\\
&\quad + \big(Df(x(t)) - Df(y(t))\big)\,\dot{y}(t). && (11b)
\end{aligned}$$

The first term of the right-hand-side can be expanded using

$$\begin{aligned}
\dot{x}(t) - \dot{y}(t) = f(x(t)) - f(y(t)) &+ \sum_{i=1}^{m} \big(g_i(x(t)) - g_i(y(t))\big) w_i(t)\\
&+ \sum_{i=1}^{m} g_i(x(t))\big(v_i(t) - w_i(t)\big).
\end{aligned}$$

Hence, we obtain

$$\begin{aligned}
(7a) &= (h_k/2) \int_0^1 Df(z(s))ds\,(x(t_{k+1}) - y(t_{k+1})) && (12a)\\
&\quad - \int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\ Df(x(t))\ (f(x(t)) - f(y(t)))\,dt && (12b)\\
&\quad - \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\ Df(x(t))\ (g_i(x(t)) - g_i(y(t)))w_i(t)\,dt && (12c)\\
&\quad - \sum_{i=1}^{m} \int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\ Df(x(t))\ g_i(x(t))\ (v_i(t) - w_i(t))\,dt, && (12d)\\
&\quad - \int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\ (Df(x(t)) - Df(y(t)))\ \dot{y}(t)dt && (12e)
\end{aligned}$$

where (12a) is (10a), (12b)–(12d) comes from (11a), and (12e) comes from (11b). Note that for any C^1-function $h(x)$ we can write

$$|h(x(t)) - h(y(t))| \le \|Dh(z(t))\| \cdot |x(t) - y(t)|$$

where $z(t) \in \overline{\text{conv}}\{x(t), y(t)\}$, i.e. closure of the convex hull of $\{x(t), y(t)\}$. This will allow us to obtain third-order bounds for terms ((12b), (12c), (12e)). In order to obtain a third-order estimate for the term (12d), a further integration by parts is needed. We obtain:

$$\begin{aligned}
(12d) = -\sum_{i=1}^{m} \Big[Df(x(t))\, g_i(x(t)) \int_{t_k}^{t} (s - t_{k+1/2})(v_i(s) - w_i(s))ds \Big]_{t_k}^{t_{k+1}}\\
+ \int_{t_k}^{t_{k+1}} \big(D^2 f(x(t))\, g_i(x(t)) + Df(x(t))Dg_i(x(t))\big)\,\dot{x}(t)\\
\int_{t_k}^{t} (s - t_{k+1/2})(v_i(s) - w_i(s))ds\ dt.
\end{aligned} \tag{13d}$$

Using a derivation similar to the one used for (12), again using the mean value theorem and integration by parts, we obtain

$$(9a) + (9b) = \sum_{i=1}^{m} \int_0^1 Dg_i(z(s))ds\ \big(x(t_{k+1}) - y(t_{k+1})\big)\,\hat{w}_i(t_{k+1}) \tag{14a}$$

$$+\sum_{i=1}^{m} g_i(x(t_{k+1}))\big(\hat{v}_i(t_{k+1}) - \hat{w}_i(t_{k+1})\big) \tag{14b}$$

$$-\sum_{i=1}^{m}\int_{t_k}^{t_{k+1}} \big(Dg_i(x(t)) - Dg_i(y(t))\big)\,\dot{y}(t)\,\hat{w}_i(t)dt \tag{14c}$$

$$-\sum_{i=1}^{m}\int_{t_k}^{t_{k+1}} Dg_i(x(t))\,\big(f(x(t)) - f(y(t))\big)\,\hat{w}_i(t)\,dt \tag{14d}$$

$$-\sum_{i=1}^{m}\int_{t_k}^{t_{k+1}} Dg_i(x(t))f(x(t))\,\big(\hat{v}_i(t) - \hat{w}_i(t)\big) \tag{14e}$$

$$-\sum_{i=1}^{m}\sum_{j=1}^{m}\int_{t_k}^{t_{k+1}} Dg_i(x(t))\,\big(g_j(x(t)) - g_j(y(t))\big)\,w_j(t)\,\hat{w}_i(t)\,dt \tag{14f}$$

$$-\sum_{i=1}^{m}\sum_{j=1}^{m}\int_{t_k}^{t_{k+1}} Dg_i(x(t))\,g_j(x(t))\,\big(v_j(t)\hat{v}_i(t) - w_j(t)\hat{w}_i(t)\big)\,dt. \tag{14g}$$

The term (14e) can be further integrated by parts to obtain

$$\begin{aligned}(14e) = &-\sum_{i=1}^{m}\Big[Dg_i(x(t))f(x(t))\int_{t_k}^{t}(\hat{v}(s) - \hat{w}(s))ds\Big]_{t_k}^{t_{k+1}}\\ &+\sum_{i=1}^{m}\int_{t_k}^{t_{k+1}}\big(D^2g_i(x(t))f(x(t)) + Dg_i(x(t))\,Df(x(t))\big)\dot{x}(t)\,(\hat{\hat{v}}_i(t) - \hat{\hat{w}}_i(t))\,dt\end{aligned} \tag{15e}$$

and the term (14g) to obtain

$$\begin{aligned}(14g) = &-\sum_{i=1}^{m}\sum_{j=1}^{m}\Big[Dg_i(x(t))\,g_j(x(t))\int_{t_k}^{t}\big(v_j(s)\hat{v}_i(s) - w_j(s)\hat{w}_i(s)\big)ds\Big]\\ &+\sum_{i=1}^{m}\sum_{j=1}^{m}\int_{t_k}^{t_{k+1}}\big(D^2g_i(x(t))\,g_j(x(t)) + Dg_i(x(t))\,Dg_j(x(t))\big)\,\dot{x}(t)\\ &\qquad\int_{t_k}^{t}\big(v_j(s)\hat{v}_i(s) - w_j(s)\hat{w}_i(s)\big)ds\;dt.\end{aligned} \tag{15g}$$

Eqs. (12)–(15) can be used to derive third-order local error estimates.

3.2. Error formulas

We proceed to give formulas for the local error having different assumptions on functions $f(\cdot)$, $g_i(\cdot)$ and $w_i(\cdot)$. We present necessary and sufficient conditions for obtaining local errors of $O(h)$, $O(h^2)$, $O(h^3)$, and give a methodology for obtaining even higher-order errors. Moreover, we give formulas for the error calculation in several cases.

Assume that we have a bounding box B on the solutions of (3) and (4) for all $t \in [0, T]$. This is easily achievable using the Euler method on the initial set subject to the system dynamics. Then, we can obtain constants r, V_i, K, K_i, L, L_i, H, Λ such that

$$\begin{aligned}&|v_i(t)| \le V_i,\ \ |w_i(t)| \le rV_i\ \ \|f(z(t))\| \le K,\ \ \|g_i(z(t))\| \le K_i\ \ \lambda(Df(z(t))) \le \Lambda,\\ &\|Df(z(t))\| \le L,\ \ \|Dg_i(z(t))\| \le L_i,\ \ \|D^2f(z(t))\| \le H,\ \ \|D^2g_i(z(t))\| \le H_i,\end{aligned} \tag{16}$$

for each $i = 1, \ldots, m$, and for all $t \in [0, T]$, and $z(\cdot) \in B$. We also set

$$K' = \sum_{i=1}^{m} V_i\,K_i,\quad L' = \sum_{i=1}^{m} V_i\,L_i\quad H' = \sum_{i=1}^{m} V_i\,H_i.$$

When possible we estimate the difference of the solutions using the logarithmic norm rather than the Lipschitz constant. To obtain the actual error value, we replace variables and functions by their bounds from Eq. (16). In each of the cases, $w_i(a, \cdot)$ is a real valued finitely-parameterized function with $a \in A \subset \mathbb{R}^N$. In general, the number of parameters N depends on the number of inputs and the order of error desired. In what follows, we denote $\varphi(x) = (e^x - 1)/x$.

3.2.1. Local error of $O(h)$

Theorem 3. *For any $k \geq 0$, and all $i = 1, \ldots, m$, if*

- *$f(\cdot)$ is a Lipschitz continuous vector function,*
- *$g_i(\cdot)$ are continuous vector functions, and*
- *$w_i(t) = 0$ on $[t_k, t_{k+1}]$,*

then the local error is of $O(h)$. Moreover, a formula for the error is:

$$\left|x(t_{k+1}) - y(t_{k+1})\right| \leq h_k\, K'\, \varphi(\Lambda h_k). \tag{17}$$

Alternatively, we can use

$$\left|x(t_{k+1}) - y(t_{k+1})\right| \leq h_k \left(2K + K'\right). \tag{18}$$

Proof. Since $w_i(t) = 0$, we have $\dot{y}(t) = f(y(t))$. Using the bounds given in (16), we can take $l(t) = \Lambda$ in Theorem 2 and since

$$\left\| \dot{y}(t) - \left(f(y(t)) + \sum_{i=1}^{m} g_i(y(t)) v_i(t) \right) \right\| = \left\| \sum_{i=1}^{m} g_i(y(t)) v_i(t) \right\| \leq \sum_{i=1}^{m} K_i\, V_i = K',$$

we can take $\delta(t) = K'$. Hence the formula (17) is obtained directly from Theorem 2. Note that $\varphi(\Lambda h_k) = 1 + \Lambda h_k/2 + \cdots$ is $O(1)$, so the local error is of $O(h)$. Eq. (18) can be obtained by noting that $\sup_{t \in [t_k, t_{k+1}]} \|f(x(t)) - f(y(t))\| \leq 2K$. □

3.2.2. Local error of $O(h^2)$

In order to obtain $O(h^2)$ error, we need $w_i(\cdot)$ functions to satisfy

$$\int_{t_k}^{t_{k+1}} v_i(t) - w_i(t)\, dt = 0 \tag{19}$$

on $[t_k, t_{k+1}]$. Simplest way is to set w_i to be a constant $w_i = \frac{1}{h_k} \int_{t_k}^{t_{k+1}} v_i(t) dt$.

Theorem 4. *For any $k \geq 0$, and all $i = 1, \ldots, m$, if*

- *$f(\cdot)$, $g_i(\cdot)$ are C^1 vector functions, and*
- *$w_i(t)$ are real valued, constant functions defined on $[t_k, t_{k+1}]$ by $w_i = \frac{1}{h_k} \int_{t_k}^{t_{k+1}} v_i(t) dt$,*

then a formula for calculation of the local error is given by

$$\|x(t_{k+1}) - y(t_{k+1})\| \leq h_k^2 \left(\left(K + K'\right) L'/3 + 2\, K' \left(L + L'\right) \varphi(\Lambda h_k) \right). \tag{20}$$

Proof. To derive (20), we obtain $\|x(t_{k+1}) - y(t_{k+1})\|$ from Eqs. (7a) and (8). Using the bounds given in (16), it is immediate that $\|\dot{x}\| \leq K + K'$, and straightforward to show that $|w_i(t)| \leq V_i$ and $|\hat{v}_i(t) - \hat{w}_i(t)| \leq 2V_i\, h_k$ for $t \in [t_k, t_{k+1}]$. However, we can get a slightly better bound $|\hat{v}_i(t) - \hat{w}_i(t)| \leq V_i\, h_k/2$ by considering the following: Without loss of generality, assume $t \in [0, h]$, and let

$$a_i(t) = \frac{1}{t} \int_0^t v_i(s)\, ds, \quad b_i(t) = \frac{1}{h - t} \int_t^{h-t} v_i(s)\, ds$$

and define

$$w_i(t) = (t\, a_i(t) + (h - t)\, b_i(t))/h.$$

Then, $w_i = w_i(t)$ is constant for all $t \in [0, h]$. Notice that $\hat{v}_i(t) = ta(t)$ and $\hat{w}_i(t) = (t/h)(ta(t) + (h - t)b(t))$. Hence, we have

$$\hat{v}_i(t) - \hat{w}_i(t) = t(h - t)(a(t) - b(t))/h,$$
$$|\hat{v}_i(t) - \hat{w}_i(t)| = t(h - t)|a(t) - b(t)|/h \leq V_i\, h/2.$$

Additionally, we can prove that $\int_{t_k}^{t_{k+1}} |\hat{v}_i(t) - \hat{w}_i(t)|\, dt \leq V_i\, h_k^2/3$. Take $z(t)$ to satisfy the differential equation $\dot{z}(t) = f(z(t))$. From Theorem 2, we have

$$\|x(t) - z(t)\|, \|y(t) - z(t)\| \leq h_k\, K'\, \varphi(\Lambda h_k)$$

and hence

$$\|x(t) - y(t)\| \leq 2\, h_k\, K'\, \varphi(\Lambda h_k)$$

for $t \in [t_k, t_{k+1}]$. Taking the norm of the Eqs. (7a), (8a), (8c) we obtain the desired formula (20). □

Remark 1. Note that as $\Lambda \to 0$, then $\frac{e^{\Lambda h}-1}{\Lambda h} \to 1$. This is also consistent with Theorem 2. In fact, if $\Lambda = 0$, we get

$$\|x(t) - y(t)\| \leq 2\, h_k \left(\sum_{i=1}^{m} K_i\, V_i\right)$$

and therefore,

$$\|x(t_{k+1}) - y(t_{k+1})\| \leq h_k^2 \left(\left(K + K'\right) L'/3 + 2\, K' \left(L + L'\right)\right), \tag{21}$$

which is still of $O(h^2)$. Further, we will not give explicit formulas for the error when $\Lambda = 0$.

Remark 2. Computation of the local error is complicated by the fact that $|v_i(t) - w_i(t)|$ is not uniformly small. This means that the terms $g(x)(v_i - w_i)$ must be integrated over a complete time step in order to be able to use the fact that $\int_{t_k}^{t_{k+1}} v_i(t)\, dt = \int_{t_k}^{t_{k+1}} w_i(t)\, dt$, and this must be done *without* first taking norms inside the integral. As a result, we cannot apply results on the logarithmic norm directly. Instead, we "bootstrap" the procedure by applying a first-order estimate for $\|x(t) - y(t)\|$ valid for any $t \in [t_k, t_{k+1}]$.

3.2.3. Local error $O(h^2) + O(h^3)$

We can attempt to improve the error bounds by allowing $w_i(t)$ to have two independent parameters. In the general case, we shall see that this gives rise to a local error estimate containing terms of $O(h^2)$ and $O(h^3)$, rather than the anticipated pure $O(h^3)$ error.

We seek two-parameter $w_i(t)$ functions which satisfy the following pair of equations

$$\int_{t_k}^{t_{k+1}} v_i(t) - w_i(t)\, dt = 0; \tag{22a}$$

$$\int_{t_k}^{t_{k+1}} (t - t_{k+1/2})\, (v_i(t) - w_i(t))\, dt = 0. \tag{22b}$$

Among the various possibilities, we found that the following three representations for $w_i(t)$ have good theoretical properties:

(a) Step-function representation in the form:

$$w_i(t) = \begin{cases} a_{i,0} & \text{if } t_k \leq t < t_{k+1/2} \\ a_{i,1} & \text{if } t_{k+1/2} \leq t \leq t_{k+1}, \end{cases}$$

where $t_{k+1/2} = t_k + h/2$.

(b) Affine function given as:

$$w_i(t) = a_{i,0} + a_{i,1}(t - t_{k+1/2})/h_k$$

(c) Sinusoidal function in the form of:

$$w_i(t) = a_{i,0} + a_{i,1}\, \sin\left(\gamma\, (t - t_{k+1/2})/h_k\right)$$

for $\gamma = 4.1632$.

To obtain appropriate sets of input functions $w_i(t)$, we aim to match the moments of $v_i(t)$:

$$\mu_{i,0} = \tfrac{1}{h} \textstyle\int_{t_k}^{t_{k+1}} v_i(t)\, dt;$$
$$\mu_{i,1} = \tfrac{4}{h^2} \textstyle\int_{t_k}^{t_{k+1}} (t - t_{k+1/2}) v_i(t)\, dt.$$

These satisfy $|\mu_{i,0}| \leq V_i$ and $|\mu_{i,1}| \leq (1 - \mu_{i,0}^2/V_i^2)V_i$, hence they can be parameterized as

$$\mu_{i,0} = c_{i,0},$$
$$\mu_{i,1} = (1 - c_{i,0}^2/V_i^2)c_{i,1}$$

for $|c_{i,0}|, |c_{i,1}| \leq V_i$. If $w_i(\cdot)$ are step-functions in the form presented in (a), then $a_{i,0} = \mu_{i,0} - \mu_{i,1}$ and $a_{i,1} = \mu_{i,0} + \mu_{i,1}$. To obtain the exact set for parameters $a_{i,0},\ a_{i,1}$ take

$$v_i(t) = \begin{cases} -V_i \text{ for } t \in [t_k, t_k + \tau) \\ +V_i \text{ for } t \in [t_k + \tau, t_{k+1}]. \end{cases}$$

Then we get

$$\mu_{i,0} = (1 - 2\tau/h)V_i$$
$$\mu_{i,1} = (4\tau/h - 4\tau^2/h^2)V_i$$

and hence

$$a_{i,0} = (1 - 6\tau/h + 4\tau^2/h^2)V_i$$
$$a_{i,1} = (1 + 2\tau/h - 4\tau^2/h^2)V_i,$$

for which we find the bounds

$$|a_{i,0}|, |a_{i,1}| \leq 5V_i/4 \text{ and } |w_i(t)| \leq 5V_i/4.$$

We can further re-parameterize $a_{i,0}$ and $a_{i,1}$ by taking

$$a_{i,0} = V_i\big(c_{i,0} - (1 - c_{i,0}^2)c_{i,1}\big)$$
$$a_{i,1} = V_i\big(c_{i,0} + (1 - c_{i,0}^2)c_{i,1}\big),$$

where $c_{i,0}, c_{i,1} \in [-V_i, +V_i]$. This yields precisely the parameter values corresponding to an actual input $v_i(t)$.

If $w_i(\cdot)$ are affine functions, then solving (22) yields $a_{i,0} = \mu_{i,0}$ and $a_{i,1} = 3\mu_{i,1}$. To provide exact bounds for $w_i(t)$, for a given $a_{i,0}$, we can maximize $a_{i,1}$ which gives us $a_{i,1} = 3(1 - a_{i,0}^2/V_i^2)$ yielding the constraint

$$a_{i,0}^2 + |a_{i,1}|/3 \leq 1.$$

Re-parameterizing, we can set $a_{i,0} = c_{i,0}$ and $a_{i,1} = 3(1 - c_{i,0}^2/V_i^2)c_{i,1}$ with $c_{i,0}, c_{i,1} \in [-V_i, +V_i]$, which then gives

$$w_i(t) = c_{i,0} + 3(1 - c_{i,0}^2/V_i^2)c_{i,1}\,(t - t_{k+1/2})/h_k. \tag{23}$$

Hence,

$$|a_{i,0}| \leq V_i, \;\; |a_{i,1}| \leq 3V_i(1 - (a_{i,0}/V_i)^2) \text{ and } |w_i(t)| \leq 5V_i/3. \tag{24}$$

Alternatively, if $w_i(t)$ are sinusoidal functions in the form given in (c), then $a_{i,0} = \mu_{i,0}$ and $a_{i,1} = p(\gamma)\mu_{i,1}$ where

$$p(2\gamma) = \tfrac{1}{2}\gamma/\big(\sin(\gamma)/\gamma - \cos(\gamma)\big),$$

and the maximum value of $|w_i|$ is $(p(\gamma) + 1/4p(\gamma))V_i$. To obtain the smallest possible maximum value we minimize $p(\gamma) + 1/4p(\gamma)$ which yields $\gamma \approx 4.163152$ with $p(\gamma) \approx 1.146311$, $p(\gamma) + 1/4p(\gamma) \approx 1.364402$. Hence

$$w_i(t) = c_{i,0} + (1 - c_{i,0}^2/V_i^2)\,c_{i,1}\,\sin(4.1632(t - t_{k+1/2}));$$
$$|c_{i,0}|, |c_{i,1}| \leq V_i; \quad |w_i| \leq 1.3645\,V_i.$$

In all cases (a–c) we see that $|w_i| \leq r\,V_i$, where r is a constant obtained depending on the choice of the $w_i(\cdot)$ functions. The bound for the local error is then given by the following theorem:

Theorem 5. *For any $k \geq 0$, and all $i = 1, \ldots, m$, if*

- *$f(\cdot)$ is a C^2 vector function,*
- *$g_i(\cdot)$ are non-constant C^2 functions, and*
- *$w_i(t)$ are real-valued functions defined on $[t_k, t_{k+1}]$ which satisfy Eqs. (22) with $|w_i(t)| \leq r\,V_i$ for some constant $r \in \mathbb{R}$,*

then an error of $O(h^2)$ is obtained. The formula for the error is given by

$$\big(1 - L(h_k/2) - h_k\,r\,L'\big)\,\|x(t_{k+1}) - y(t_{k+1})\| \leq (h_k^2/4)(1 + r^2)\,L'\,K'$$
$$+ (h_k^3/4)(1 + r)\,K'\,\big((2rH' + H)(K + rK') + L^2 + \big(3rL + 2r^2L'\big)\,L'\big)\,\varphi(\Lambda h_k)$$
$$+ (h_k^3/24)(1 + r)\,\big(K + K'\big)\,\big(3(H\,K' + L\,L') + 4(H'K + LL')\big).$$

Proof. To get the desired formula we have to provide bounds to equations in (13d) and (15e). With the assumptions of the theorem, we can improve terms (12d) and (14e) such that they become (13d) and (15e), which are of $O(h^3)$. In addition, we use

$$\|\dot{x}(t)\| \leq K + K', \quad \|\dot{y}(t)\| \leq K + rK'$$
$$\|x(t) - y(t)\| \leq h_k\,(1 + r)\,K'\,\varphi(\Lambda h_k),$$

and bound the rest of the terms in Eqs. (12) and (14). Formula for the error $\|x(t_{k+1}) - y(t_{k+1})\|$ is then easily obtained. □

We now show that with the assumptions of theorem we cannot in general obtain an error of $O(h^3)$. Specifically, if $w_i(t)$ are two-parameter functions satisfying Eqs. (22), the following counterexample gives a system for which only $O(h^2)$ local error is possible.

Example 1. Consider the following input-affine system which satisfies assumptions in Theorem 5:

$$\dot{x}_1 = x_2 + v_1 + x_1 v_2; \quad \dot{x}_2 = x_1 + v_2; \quad x(t_k) = x_k.$$

Take inputs

$$v_1(t) = \sin\left(\frac{2\pi}{h_k}(t - t_k)\right), \qquad v_2(t) = \cos\left(\frac{2\pi}{h_k}(t - t_k)\right).$$

Using (22), we get $w_2(t) = 0$, and $w_1(t)$ is nonzero ($w_1(t)$ can be explicitly calculated for all three functions but we do not need it), hence the auxiliary systems looks like

$$\dot{y}_1 = y_2 + w_1; \quad \dot{y}_2 = y_1.$$

As shown in the previous section, the only term which might not have order h_k^3 is the term in (14g) which is reduced to

$$\sum_{i=1}^{2} \int_{t_k}^{t_{k+1}} Dg_2(x(t))g_i(x(t))\, v_i(t)\hat{v}_2(t)dt,$$

since $Dg_1 = 0$. When $i = 2$, the term above is of $O(h^3)$ since $\frac{1}{2}\frac{d}{dt}(\hat{v}_i^2(t)) = v_i(t)\hat{v}_i(t)$ and we can integrate by parts once more. Therefore, we are left with

$$\int_{t_k}^{t_{k+1}} Dg_2(x(t))g_1(x(t))\, v_1(t)\hat{v}_2(t)dt = -\frac{h_k^2}{4\pi}\ [1\ \ 0]^T,$$

a term of $O(h^2)$.

3.2.4. Local error of $O(h^3)$

We showed that for a general input-affine system, a local error of order $O(h^3)$ cannot be obtained using two-parameter approximate inputs $w_i(a_{0,i}, a_{1,i}, t)$. However if, in addition, we assume that $g_i(\cdot)$ are constant functions or if we have a single input then we can obtain a local error of $O(h^3)$. If $g_i(\cdot)$ are constant functions, then the error calculation is equivalent to the error calculation of an even simpler case, the so called additive noise case. The equation is then given by

$$\dot{x}(t) = f(x(t)) + v(t). \tag{25}$$

Here, $v(t) = (v_1(t), \ldots, v_n(t))$ is vector-valued.

Corollary 1. *For any $k \geq 0$,*

- *if the system has additive noise,*
- *$f(\cdot)$ is a C^2 function, and*
- *$w_i(t)$ are real-valued functions defined on $[t_k, t_{k+1}]$ which satisfy Eqs. (22) with $|w_i(t)| \leq r\, V_i$, for all $i = 1, \ldots, n$ and some constant $r \in \mathbb{R}$*

then an error of $O(h^3)$ is obtained:

$$\begin{aligned}\left(1 - (h_k/2)L\right)\|x(t_{k+1}) - y(t_{k+1})\| \leq\ & \frac{h_k^3}{8}\,(1+r)\,K'\,H\,(K+K') \\ & + \frac{h_k^3}{4}\,(1+r)\,K'\left(L^2 + H\,(K + rK')\right)\varphi(\Lambda h_k).\end{aligned} \tag{26}$$

The formula for the error in the additive noise case is simplified because $L' = H' = 0$. If we write $\|v(t)\| = K'$, then the result follows directly from Theorem 5.

Corollary 2. *For any $k \geq 0$, if*

- *the input-affine system has a single input, i.e., $m = 1$ in (3)*
- *$f(\cdot)$ and $g(\cdot)$ are C^2 functions, and*
- *$w_i(t)$ are real-valued functions defined on $[t_k, t_{k+1}]$ which satisfy Eqs. (22) with $|w_i(t)| \leq r\, V_i$, for all $i = 1, \ldots, n$ and some constant $r \in \mathbb{R}$*

then an error of $O(h^3)$ is obtained. The formula for the local error is given by

$$\begin{aligned}&\left(1 - L(h_k/2) - h_k\, r\, L'\right)\|x(t_{k+1}) - y(t_{k+1})\| \leq \\ &(h_k^3/4)(1+r)K'\left((2rH' + H)(K + rK') + L^2 + \left(3rL + 2r^2L'\right)L'\right)\varphi(\Lambda h_k) \\ &+ (h_k^3/24)\left(K + K'\right)\left((1+r)(3(HK' + LL') + 4(H'K + LL')\right. \\ &\qquad \left. + 8(1 + r^2)(H'K' + (L')^2)\right).\end{aligned}$$

Table 1
Total number of parameters needed depending on the number of inputs m in the system. If $w_i(\cdot)$ are polynomials, the highest degree needed for at least one $w_i(\cdot)$ is given.

# of inputs $= m$	# of equations = total # of parameters $= m(m+3)/2$	Highest degree d of a $w_i = \lceil (m+1)/2 \rceil$
1	2	1
2	5	2
3	9	2
4	14	3
5	20	3
6	27	4
10	65	5

Observing the error given by Eqs. (12) and (14), we see that if in addition to satisfying equations given in (22), the functions $w_i(\cdot)$ also satisfy

$$\int_{t_k}^{t_{k+1}} v_i(t)\hat{v}_j(t) - w_i(t)\hat{w}_j(t)\, dt \;=\; 0, \tag{27}$$

then we can get an error of $O(h^3)$. Since, functions $w_i(\cdot)$ cannot be computed independently any more, the number of parameters of each $w_i(\cdot)$ will depend on the number of inputs. In Table 1, we present the total number of parameters needed depending on the number of inputs in the system. In addition, if $w_i(\cdot)$ are polynomials, we disclose the highest degree required for at least one $w_i(\cdot)$ so that a local error of $O(h^3)$ is obtained. However, further investigation in obtaining suitable functions $w_i(t)$ is not desirable as it does not seem to be computationally feasible.

4. Implementation

The algorithm used for computing the reachable set of (3) is:

Algorithm 1. Let $R_k = \{h_k(s) + [-\varepsilon_k, \varepsilon_k]^n \mid s \in [-1, +1]^{p_k}\}$ be an over-approximation of the set $R(X_0, t_k)$. To compute an over-approximation R_{k+1} of $R(X_0, t_{k+1})$:

1. Create the auxiliary system

$$\dot{y}(t) = f(y(t), w(a_k, t)),\;\; x(t_k) = x_k = y_k, \quad t \in [t_k, t_{k+1}], y_k \in R_k, a_k \in A.$$

2. Compute the necessary bounds as presented at the beginning of Section 3.1
3. Compute the uniform error bound ϵ_k which represents the distance between the two solutions, i.e., $\|\phi(x_k, v_k(\cdot)) - \phi(x_k, w(a_k, \cdot))\| \leq \epsilon_k$
4. Compute the flow of the auxiliary system via *Taylor model* integration, i.e., obtain $(h(s_k) + [-\varepsilon_k, \varepsilon_k]^n, a_k)$ that represents an over-approximation of the solution set (see Section 2 on computation in Ariadne).
5. Compute the set R_{k+1} which over-approximates $R(x_0, t_{k+1})$ as $R_{k+1} = \{(h(s_k) + [-\varepsilon_k, \varepsilon_k]^n, a_k) + [-\epsilon_k, \epsilon_k]^n\}$, i.e., the Taylor model obtained in step 5 $\pm$ the analytical error obtained in step 3.
6. Simplify parameters (if necessary).

Step 4 of the algorithm produces an approximated flow $\phi(x_k, w(a_k, \cdot))$ which is guaranteed to be valid for all $x_k \in R_k$. In practice, we cannot represent ϕ exactly, and instead use Taylor model approximation with guaranteed error bound. In Step 3, we compute the uniform error bound ϵ_k and in Step 5 we add it to the computed flow to obtain an over-approximation, R_{k+1}. Step 6 is crucial for the efficiency and accuracy of the algorithm, as explained below.

According to the theoretical framework, the approximation error is reduced by decreasing the step size h. However, when an actual implementation is concerned, other numerical aspects contribute to the quality of representation of the sets and the resulting over-approximations. In particular, the computational error, i.e., the error due to implementation of the algorithm in Ariadne, contributes towards over-approximation of the solution set in two ways. One is due to the *Taylor model* calculus used and the other due to simplification of the parameters.

In order to prevent the eventual blow-up of the number of polynomial terms used in the Taylor model, small and/or high-order terms must be "swept" into the uniform error bound e. For this purpose, Ariadne introduces a *sweep threshold* σ_{thr} constant that represents the minimum coefficient that a term needs in order to avoid being swept into e. As already discussed, an additional contribution to e is the error originating from the inputs approximation, which is added to the model for each variable. Therefore, over time, e becomes relatively large, ultimately causing the bounds of the represented set to diverge; to address this issue, we need to extract periodically a new parameter for each variable, thus originating n new independent parameters ("uniform error reconditioning"). In particular, our experience with the implementation showed that significantly more accurate results are obtained by parameter extraction at each evolution step, introducing n new parameters at each step. At the same time, each step of the proposed algorithm introduces ℓm additional parameters

into the description of the flow, where ℓ is the number of parameters required for each $w_{i,k}$: $\ell = 0$ for the zero case, $\ell = 1$ for the constant case, and $\ell = 2$ for the affine, sinusoidal and piecewise constant cases. Summarizing, after k steps we end up introducing $k(n + \ell m)$ new parameters.

Therefore it is apparent that a critical requirement for the feasibility of the algorithm is to simplify periodically the representation of the reached sets. For the purposes of this paper, we rely on the following basic simplification policy: after a number of steps N_s we keep a number of parameters equal to a multiple β_s of the parameters introduced between two simplifications. To decide which parameters to keep after the simplification, we sum the coefficients of the terms where a parameter is present: the parameters with the lowest sum are considered to have the least impact on the set representation and their terms are simplified into e ("Kuhn reconditioning" [37]). Increasing β_s increases the *average* number of parameters during evolution, while increasing N_s increases the *variance* of such number. Consequently, techniques that reduce the number of parameters are essential for scalability purposes.

Note that our method only guarantees a local error of high order at the sequence of rational points $\{t_k\}$ which is *a priori* chosen. If one is trying to estimate the error at times $t_k < t < t_{k+1}$ for any k along a *particular* solution, an $O(h)$ formula should be used as given in Theorem 3.

5. Numerical results

In this Section we present the results of the implementation of our approach within Ariadne, followed by a comparison with Flow* and CORA 2018. Before that, the first Subsection explains the evaluation criteria, followed by the values chosen for the numerical parameters of the three tools and by the description of the systems to be used for evaluation.

5.1. Evaluation setup

In order to evaluate the quality of the reachable set of a system, we introduce the *volume score* (from here on simply *score*) Σ_V as

$$\Sigma_V = \frac{1}{\sqrt[n]{\Pi_{i=1}^{n} |B_i|}} \tag{28}$$

where B is the bounding box of a set. Given a set, the formula over-approximates it into a box for simplicity, evaluates its volume and normalizes on the number of variables. In particular, halving the set on each dimension yields twice the score. Without extra notation, we evaluate Σ_V on the final set of evolution to measure the quality of the whole trace. It must be noted that since a bounding box returns an over-approximation, this measure is not entirely reliable when used for comparisons: given two different sets with equal exact bounds, a slightly larger box may be obtained for the set having the more complex representation. Still, it is an intuitive and affordable measure that can be used across tools with different internal representations.

In addition to the volume score, we evaluate the performance in terms of execution time t_x in seconds. In particular, the execution times are obtained using a macOS 10.14.6 laptop with an Intel Core i7-6920HQ processor, using AppleClang 10.0.1 as a compiler in the case of Ariadne and Flow* executables, or running on MATLAB 2018b in the case of CORA. Finally, all the score and execution time values in the following are rounded to the nearest least significant digit.

In Table 2 we provide numerical parameters used for evaluation in the benchmark. For simplicity we used fixed reasonable values for Ariadne. For Flow* and CORA we asked the respective developers to collaborate in order to identify good values. In the case of Flow*, the values identified were actually kept fixed for all systems (since small variations did not show any significant difference in behavior), while for CORA those have been specified tailored to the system.

We evaluate ten different systems taken from the literature, with varying nonlinearity, whose summary is provided in Table 3. In addition to synthetic information such as the number of variables, number of inputs, average order $\bar{O}$ of dynamics and additivity of inputs, the step size h and the evolution time T_e are presented for each system. For quick reference we also show the number of steps involved in the evolution $h \times T_e$.

Details on systems dynamics, input ranges, and overridden tool parameters used by CORA instead are presented in Table 4. With respect to [14], input range widths for the CR system have been divided by 100 since none of the three tools were able to analyze the system otherwise. With respect to [20], the DC system has been rewritten in its equivalent input-affine form in order to be analyzed using Ariadne.

5.2. Results

This subsection on results starts by evaluating the quality of approximation with and without simplification of the parameters that represent a set. After assessing the quality at the default noise levels, we analyze the effect of varying the noise levels. The next subsection will compare these results with those obtained using CORA and Flow*.

Given the large size of the benchmark suite and lack of space, figures are omitted. Instead we will rely on quantitative tabular data based on the metrics that were previously introduced.

In Table 5 we show results in terms of score Σ_V and execution time t_x for distinct setups while using the following approximations: Z for zero, C for constant, A for affine, S for sinusoidal and P for piecewise-constant. In order to evaluate

Table 2
Numerical parameters used for evaluation in the benchmark.

Ariadne	Flow*	CORA
• Sweep threshold σ_{thr}: 10^{-8} • Number of steps between simplifications N_s: 12 • Number of parameters to be kept after a simplification β_s: 6	• Mantissa precision: 53 bits • Taylor model fixed order: 6 • Cutoff threshold: 10^{-10} • Remainder estimation: 0.1	• `zonotopeOrder`: 100 • `tensorOrder`: 3 • `errorOrder`: 25 • `intermediateOrder`: 100 • `taylorTerms`: 5 • `advancedLinErrorComp`: 0 • `reductionInterval`: inf • `reductionTechnique`: 'girard' • `maxError`: as large as possible to avoid splitting

Table 3
Summary information on systems tested.

Name	Alias	Ref	n	m	$\bar{O}$	+	h	T_e	Steps
Higgins-Sel'kov	HS	[5]	2	3	3	N	1/50	10	500
Chemical Reactor	CR	[14]	4	3	2	N	1/16	10	160
Lotka–Volterra	LV	[14]	2	2	2	N	1/50	10	500
Jet Engine	JE	[12]	2	2	2	Y	1/50	5	250
PI Controller	PI	[12]	2	1	2	Y	1/32	5	160
Jerk Eq. 21	J21	[38]	3	1	5/3	N	1/16	10	160
Lorenz Attractor	LA	[39]	3	1	5/3	N	1/256	1	256
Rössler Attractor	RA	[39]	3	1	5/3	Y	1/128	12	1536
Jerk Eq. 16	J16	[38]	3	1	4/3	Y	1/16	10	160
DC-DC Converter	DC	[20]	2	2	1	N	1/10	5	50

the complete benchmark suite we compare best static approximation obtained with two different dynamic evaluations. Best static displays best results from the five approximations, each one being used 100% of the time. Since this approach shows that even if we focus on approximations using two parameters, the best result may largely depend on the system under analysis, we should check all available approximations and choose the best one. Our framework allows for this choice to be performed at each integration step. However, this *tight* approach incurs in a significant cost in terms of execution time, slightly lower than the sum of the costs in static evaluation. Consequently we defined a *loose* approach for choosing the best approximation: a counter k_a is associated with a given approximation a, with $k_a = 1 \, \forall a$ at the beginning of evolution; if an approximation is not the best one, the value of k_a is doubled and a will be checked again after k_a steps; each time a it is the best one, instead we reset $k_a = 1$. Such exponential delay in checking a less-than-optimal approximation allows to focus on the best approximation(s).

Since dynamic choice will, in general, yield a mix of approximations, we show a "a%" column that summarizes the frequency of choosing a given approximation, i.e., A93P7 means that the affine approximation was the best one on 93% of the steps while the piecewise-affine approximation was chosen on the remaining 7%. We see that a tight dynamic choice yields better results than the best static choice; our evaluation showed that the best approximation changes infrequently and we can identify sections of the evolution where a given approximation is always chosen. Therefore such behavior is compatible with a loose dynamic choice of the best approximation: as shown in the third column of Table 5, the score Σ_V is very close to the one coming from a tight approximation, while the execution time t_x is not particularly higher than the one coming from the best static approximation. Still, the execution time remains significantly high, preventing completion for some of the systems.

Table 6 presents equivalent data to Table 5 while periodic simplification of parameters is applied. Simplification period is $N_s = 12$, and $\beta_s = 6$ times the number of parameters introduced between simplification events. On the first column we also tabulate the best loose dynamic result from Table 5 for comparison purposes. Here, we notice that the volume score metric, being inaccurate, can sometimes result in unexpected behaviors, such as for LV a loose dynamic score higher than the tight dynamic score, or for JE a tight dynamic score worse than the best static score. Comparison with the first column shows that in some cases (i.e., at least CR and J21, if we do not consider the improvement from timeout in the HS, LV and RA cases) simplification yields a better score. This is especially true for a tight dynamic choice of the approximation, but again a loose dynamic choice allows for significantly shorter execution times with very small losses of accuracy.

We must remark that while the C approximation is very infrequently chosen in these Tables, such frequencies are very sensitive to numerical settings related to integration. In our analysis of the benchmark suite during development of the Ariadne library, small variations yielded significant changes in the mix of approximations, which further motivated the use of this dynamic approach.

In Table 7 we display some values for the uniform error ϵ and the volume $V = \Pi_{i=1}^{n} |B_i|$ of the bounding box of the set for the HS system, across its 500 integration steps. For each step, all approximations are computed while the

Table 4
Detailed information on systems tested.

Alias	System	Inputs	CORA parameters overriding defaults
HS	$\dot{S} = v_0 - Sk_1P^2$ $\dot{P} = Sk_1P^2 - k_2P$	$v_0 = 1 \pm 0.0002$ $k_1 = 1 \pm 0.0002$ $k_2 = 1.00001 \pm 0.0002$	• `zonotopeOrder`: inf • `tensorOrder`: 2
CR	$\dot{x}_A = -u_3x_Ax_B - 0.4x_Ax_C + 0.05u_1 - 0.1x_A$ $\dot{x}_B = -u_3x_Ax_B + 0.05u_2 - 0.1x_B$ $\dot{x}_C = u_3x_Ax_B - 0.4x_Ax_C - 0.1x_C$ $\dot{x}_D = 0.4x_Ax_C - 0.1x_D$	$u_1 = 1 \pm 0.001$ $u_2 = 0.9 \pm 0.001$ $u_3 = 30 \pm 0.2$	• `tensorOrder`: 2
LV	$\dot{x} = u_1\, x(1-y)$ $\dot{y} = u_2\, y(x-1)$	$u_1 = 3 \pm 0.01$ $u_2 = 1 \pm 0.01$	• `zonotopeOrder`: 10 • `tensorOrder`: 2 • `reductionInterval`: 50
JE	$\dot{x} = -y - 1.5x^2 - 0.5x^3 - 0.5 + u_1$ $\dot{y} = 3x - y + u_2$	$u_1 = \pm 0.005$ $u_2 = \pm 0.005$	• `zonotopeOrder`: 200 • `intermediateOrder`: 200 • `advancedLinErrorComp`: 1
PI	$\dot{v} = -0.101(v-20) + 1.3203(x - 0.1616) - 0.01v^2$ $\dot{x} = 0.101(v-20) - 1.3203x + 0.2134 + 0.01v^2 + 3(20-v) + u$	$u = \pm 0.1$	• `zonotopeOrder`: 200 • `advancedLinErrorComp`: 1
J21	$\dot{x} = y$ $\dot{y} = z$ $\dot{z} = -z^3 - yx^2 - ux$	$u = 0.25 \pm 0.01$	• `zonotopeOrder`: 300 • `intermediateOrder`: 200 • `errorOrder`: 50 • `advancedLinErrorComp`: 1
LA	$\dot{x} = 10(y-x)$ $\dot{y} = x(u-z) - y$ $\dot{z} = xy - 8z/3$	$u = 28 \pm 0.01$	• `zonotopeOrder`: 300
RA	$\dot{x} = -y - z$ $\dot{y} = x + 0.1y$ $\dot{z} = z(x-6) + u$	$u = 0.1 \pm 0.001$	
J16	$\dot{x} = y$ $\dot{y} = z$ $\dot{z} = -y + x^2 + u$	$u = -0.03 \pm 0.001$	
DC	$\dot{x} = -0.018\,x - 0.066\,y + u_1(\frac{1}{600}x + \frac{1}{15}y) + u_2$ $\dot{y} = 0.071\,x - 0.00853\,y + u_1(-\frac{1}{14}x - \frac{20}{7}y)$	$u_1 = \pm 0.002$ $u_2 = \frac{1}{3} \pm \frac{1}{15}$	• `taylorTerms`: 20 • `tensorOrder`: 2

Table 5
Volume score Σ_V and execution times t_x in seconds for each system and various setups, when not simplifying the number of parameters; the first one picks the best approximation statically; the second one comes from dynamically evaluating each approximation at each step and selecting the best one; the third one comes from dynamically evaluating each approximation with a frequency proportional to its quality. The best Σ_V for a given system is emphasized in bold. A timeout (T.O.) is obtained if completion is not achieved within 8 h of execution.

	Best static			Tight dynamic			Loose dynamic		
	Σ_V	t_x	a	Σ_V	t_x	a%	Σ_V	t_x	a%
HS	32.16	11 143	C	T.O.			T.O.		
CR	323.3	640	A	**324.0**	3894	A93P7	323.6	683	A91P9
LV	12.08	7 674	C	T.O.			T.O.		
JE	16.13	166	Z	**16.16**	1887	Z82P18	16.13	171	Z100
PI	5.959	105	S	**5.962**	580	S44P56	5.960	151	S24P76
J21	23.41	292	A	**23.94**	1433	C2A86S4P8	23.41	295	A100
LA	12.15	1 152	S	**12.22**	6398	A61S24P15	12.20	1429	A48S37P15
RA	T.O.			T.O.			T.O.		
J16	**26.86**	165	A	**26.86**	901	A96P4	**26.86**	176	A96P4
DC	**1.920**	1 130	A	**1.920**	6534	A97P3	**1.920**	1268	A96P4

one with the best volume is used (i.e., tight dynamic choice); simplification of the parameters is performed, purely for efficiency reasons. The integration step k values are chosen specifically to identify some of the switches between the A and P approximations. Despite the fact that ϵ is consistently better for the P approximation, we see that the best volume sometimes changes to A. While the difference-per-step is usually low, in some cases smaller than the displayed precision, we know from Table 6 that a dynamic choice of the parameter is better than the best static one.

Table 6
Score Σ_V and execution time t_x in seconds for each system and various setups, when simplifying the parameters; the first one represents the loose selection; the second one represents the best selection when simplification is performed; the third one comes from dynamically evaluating each approximation at each step and selecting the best one; the fourth one comes from dynamically evaluating each approximation with a frequency proportional to its quality. The best score for a given system is emphasized in bold, while the best score when simplifying the parameters is emphasized through underlining, if not already the absolute best score.

	Loose dynamic (no simpl.)			Best static			Tight dynamic			Loose dynamic		
	Σ_V	t_x	a%	Σ_V	t_x	a	Σ_V	t_x	a%	Σ_V	t_x	a%
HS	T.O.			48.40	38	A	**49.49**	242	A88P12	48.91	39	A94P6
CR	323.6	683	A91P9	502.3	21	A	**504.5**	181	A91P9	502.4	26	A91P9
LV	T.O.			14.53	60	A	14.53	366	A95P5	**14.54**	62	A94P6
JE	**16.13**	171	Z100	15.47	25	Z	14.39	155	Z78P21	15.47	28	Z100
PI	**5.960**	151	S24P76	5.492	7.8	P	5.493	30	S15P85	5.492	8.8	P100
J21	23.41	295	A100	23.23	15	P	**23.77**	63	C1A86P13	23.10	14	A100
LA	**12.20**	1429	A48S37P15	9.045	18	P	9.080	70	A58S6P36	9.070	18	A46S4P50
RA	T.O.			**120.0**	36	P	117.7	143	A96P4	113.8	27	A100
J16	**26.86**	176	A96P4	23.78	6.2	S	23.77	29	A96P4	23.77	6.1	A96P4
DC	**1.920**	1268	A96P4	1.906	5.9	A	1.906	36	A71P29	1.906	7.7	A88P12

Table 7
Uniform error ϵ and volume of the bounding box of the set V for the HS system, starting from the initial set for each integration step and each approximation. For each step, the minimum volume is underlined.

k	Z		C		A		S		P	
	ϵ	V	ϵ	V	ϵ	V	ϵ	V	ϵ	V
0	161e−3	1169e−4	123e−7	4576e−7	104e−8	4567e−7	914e−9	4568e−7	870e−9	4559e−7
1	160e−3	1160e−4	122e−7	5069e−7	102e−8	5060e−7	901e−9	5061e−7	857e−9	5052e−7
32	863e−4	3638e−5	361e−8	2554e−7	167e−9	2552e−7	148e−9	2552e−7	141e−9	2552e−7
33	853e−4	3551e−5	350e−8	2463e−7	160e−9	2460e−7	142e−9	2461e−7	135e−9	2461e−7
37	818e−4	3242e−5	312e−8	2139e−7	134e−9	2137e−7	119e−9	2138e−7	113e−9	2137e−7
210	457e−4	1000e−5	808e−9	6490e−8	168e−10	6489e−8	149e−10	6492e−8	142e−10	6488e−8
400	670e−4	2496e−4	147e−8	5414e−7	414e−10	5413e−7	367e−10	5416e−7	349e−10	5415e−7
487	496e−4	1363e−4	793e−9	2757e−7	162e−10	2757e−7	144e−10	2758e−7	137e−10	2756e−7
499	487e−4	1311e−4	798e−9	2508e−7	163e−10	2507e−7	145e−10	2508e−7	138e−10	2507e−7

Table 8
Volume score Σ_V and execution times t_x in seconds for each system, varying the noise level with respect to the nominal value.

	x 1/4		x 1/2		Nominal		x 2		x 4	
	Σ_V	t_x	Σ_V	t_x	Σ_V	t_x	Σ_V	t_x	Σ_V	t_x
HS	109.1	22	76.77	27	48.91	39	23.36	107	11.49	296
CR	1573	13	943.3	19	502.4	26	217.6	53	60.87	223
LV	69.31	12	32.70	26	14.54	62	5.947	206	1.165	5032
JE	29.21	13	21.85	19	15.47	28	9.368	50	4.953	15
PI	12.82	7.1	8.849	7.5	5.492	8.8	3.085	10	1.664	15
J21	36.31	6.8	30.47	9.2	23.10	14	15.41	27	8.807	73
LA	33.48	9.5	17.64	11	9.070	18	4.574	35	2.255	71
RA	385.6	18	221.5	20	113.8	27	58.85	48	29.12	80
J16	58.56	3.6	39.67	4.3	23.77	6.1	13.11	10	6.570	22
DC	4.877	3.9	3.816	5.4	1.906	7.7	0.944	15	0.464	23

5.2.1. Dependency on the noise level

Since the auxiliary system and the local error depend on the range of the inputs, it is interesting to study the relation between executions time, quality of the results, and the range of inputs. If we interpret inputs as noise sources, this correspond to study how the noise level affects performance. Table 8 evaluates each system using a loose dynamic choice of the best approximation while simplifying the parameters. The noise level ranges from 1/4 the nominal value to 4 times the nominal value. Results show the expected decay in volume score when noise increases. Results also show that the execution time increases: this is due to the fact that the corresponding increase in volume of the evolved set implies a more complex polynomial representation of the set.

5.3. Comparison with other tools

In this subsection we finally compare our results with those from CORA and Flow*. However, since CORA performs approximate rounding, its numerical results cannot be rigorous even when using interval arithmetics. For this reason, in the following Table the actual comparison is between Ariadne and Flow*, while CORA is used as a reference.

Table 9
Comparison with CORA and Flow* for different noise levels. For each approach and each system, the score Σ_V is shown. Since the execution time t_x is the same for Flow* regardless of the noise level, it is shown only for the nominal noise. The highest score between Ariadne and Flow* for each system and each noise level is emphasized in bold. Where CORA produces the best result, it is underlined for reference.

Setup			System									
Noise	Tool		HS	CR	LV	JE	PI	J21	LA	RA	J16	DC
$\times\frac{1}{4}$	Ariadne	Σ_V	**109.1**	**1573**	**69.31**	**29.21**	**12.82**	**36.31**	**33.48**	**385.6**	**58.56**	4.877
		t_x	22	13	12	13	7.1	6.8	9.5	18	3.6	3.9
	CORA	Σ_V	16.92	2539	14.39	18.40	11.53	7.459	11.08	264.0	51.47	7.605
		t_x	4.0	1.0	2.5	3.8	2.5	3.3	4.0	2.7	3.7	0.26
	Flow*	Σ_V	71.78	762.1	2.242	23.18	11.10	15.75	17.14	263.5	52.96	**7.559**
$\times\frac{1}{2}$	Ariadne	Σ_V	**76.77**	**943.3**	**32.70**	**21.85**	**8.849**	**30.47**	**17.64**	**221.5**	**39.67**	**3.816**
		t_x	27	19	26	19	7.5	9.2	11	20	4.3	5.4
	CORA	Σ_V	13.62	1632	5.970	15.55	8.420	6.803	8.983	177.3	38.30	3.827
		t_x	3.9	1.0	2.6	3.8	2.3	6.5	4.1	4.0	3.5	0.26
	Flow*	Σ_V	56.97	384.5	N/A	19.01	7.994	14.28	12.33	174.6	39.11	3.804
$\times1$	Ariadne	Σ_V	**48.91**	**502.4**	**14.54**	**15.47**	**5.492**	**23.10**	**9.070**	**113.8**	23.77	**1.906**
		t_x	39	26	62	28	8.8	14	18	27	6.1	7.7
	CORA	Σ_V	8.162	930.2	1.680	11.81	5.472	5.710	6.543	110.4	25.20	1.915
		t_x	3.9	1.0	3.5	3.7	2.5	6.2	4.1	4.0	3.3	0.26
	Flow*	Σ_V	37.78	169.9	N/A	13.87	5.107	11.99	8.113	107.5	**25.49**	1.902
		t_x	29	19	13	7.4	3.7	19	12	81	2.5	0.24
$\times2$	Ariadne	Σ_V	**23.36**	**217.6**	**5.947**	**9.368**	**3.085**	**15.41**	4.574	58.85	13.11	**0.944**
		t_x	107	53	206	50	10	27	35	48	10	15
	CORA	Σ_V	0.675	433.9	0.807	7.862	3.218	4.235	3.911	63.67	14.76	0.952
		t_x	4.0	1.0	127	3.6	2.3	6.2	4.1	4.0	3.3	0.26
	Flow*	Σ_V	17.49	50.50	N/A	8.828	2.931	8.948	**4.857**	**61.42**	**14.76**	**0.944**
$\times4$	Ariadne	Σ_V	**11.49**	**60.87**	**1.165**	**4.953**	**1.664**	**8.807**	2.255	29.12	6.570	0.464
		t_x	296	223	5032	185	15	73	71	80	22	23
	CORA	Σ_V	N/A	146.0	N/A	4.517	1.763	1.704	2.450	33.50	7.825	0.465
		t_x	N/A	1.0	N/A	3.6	2.2	6.1	4.0	4.0	3.4	0.75
	Flow*	Σ_V	N/A	N/A	N/A	4.827	1.577	5.599	**2.670**	**32.23**	**8.322**	**0.465**

Table 9 evaluates the quality of our approach while varying the noise level and using a fixed step size. The rationale here is that as the level increases, the impact of a more accurate input approximation increases. Systems are presented in decreasing order of nonlinearity from left to right. For mostly-linear systems CORA has the best results due to its kernel relying on linearization of the dynamics; Flow* has similar benefits due to specific optimizations on low-order polynomial representations. On the other hand, it is apparent that Flow* and CORA suffer when the nonlinearity is high, to the point of being unable to complete evolution. An N/A result in Flow* is due to failing convergence of the flow set over-approximation, while for CORA this is specifically due to a diverging number of split sets required to bound the flow set. Since Ariadne maintains a larger number of parameters when handling higher noise values, the computation time increases with the noise, while the computation times of Flow* and CORA do not depend on the noise (Table 9 shows execution times only for the nominal noise). Summarizing, in this setup Ariadne consistently gives better bounds for systems with medium and high nonlinearity, with comparable computation times with respect to Flow* for low noise levels, while also avoiding failure for high noise levels.

6. Conclusions

Here, we have given a numerical method for computing rigorous over-approximations of the reachable sets of differential inclusions. The method introduces high-order error bounds for single-step approximations. By providing improved control of local errors, the method allows for accurate computation of reachable sets over longer time intervals.

We have also presented theorems for obtaining local errors of different orders. It is easy to see that higher order errors (improved accuracy) require approximations that have a larger number of parameters (reduced efficiency). The growth of the number of parameters is an issue, in general. Sophisticated methods for handling these are at least as important as the single-step method. Nonetheless, in our evaluation of the methodology, we found that Ariadne yields tighter set bounds, as the nonlinearity increases, compared with the state-of-the-art tool Flow* and CORA. Although no analysis of the order of the method is given in [12], we believe that Flow* has a local error $O(h^2)$, so the global error is intrinsically first-order. Hence a higher quality is to be expected from Ariadne, since the proposed methodology is able to achieve third-order local errors. On the other hand, our approach introduces extra parameters at each step in the representation of the evolved set, causing a growth in complexity, whereas Flow* and CORA have a fixed complexity of the set representations. As a result, the computational cost increases with the noise level. Still, the comparison with the state-of-the-art showed that for high noise levels our approach is the only one capable of providing sufficient bounds for highly nonlinear systems. Currently, we are working towards component-wise derivations of the local error, in order to better address systems

whose variables have scaling of different orders of magnitude. Some of the other extensions on differential inclusions that we plan to accomplish are outlined in our paper [40]. These include constraint set representation of uncertainties including representation via affine and more general convex constraints. Further, an extension to nonlinearity in the inputs will be obtained, in order to maximize the expressiveness in terms of system dynamics.

While in this work we address uncertainty for continuous dynamics, we understand it is equally important to embed it within a hybrid systems framework, which we plan to address in our future work. Here, the main challenges are ensuring that invariants are never violated along a trajectory, even temporarily, and that urgent events occur as soon as possible. Our approach of computing parametrized sets of solutions with small error bounds should be well-suited to addressing these issues by allowing accurate detection of crossings of guard and invariant boundaries.

CRediT authorship contribution statement

Sanja Zivanovic Gonzalez: Theory, Writing – original draft. **Luca Geretti:** Experimental evaluation. **Davide Bresolin:** Reviewing and editing. **Tiziano Villa:** Reviewing and editing. **Pieter Collins:** Theory, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by MIUR, Project "Italian Outstanding Departments, 2018–2022" and by INDAM, GNCS 2020, "Strategic Reasoning and Automatic Synthesis of Multi-Agent Systems".

The authors would like to thank Xin Chen and Matthias Althoff for the support on setting up their respective softwares and tuning the systems for comparison.

References

[1] J. Aubin, A. Cellina, Differential Inclusions. Set-Valued Maps and Viability Theory, in: Fundamental Principles of Mathematical Sciences, vol. 264, Springer-Verlag, 1984.
[2] K. Deimling, Multivalued Differential Equations, in: Nonlinear Analysis and Applications, de Gruyter, 1992.
[3] G.V. Smirnov, Introduction to the Theory of Differential Inclusions, in: Graduate Studies in Mathematics, vol. 41, American Mathematical Society, 2002.
[4] A.F. Filippov, Differential Equations with Discontinuous Righthand Sides, in: Mathematics and its Applications (Soviet Series), vol. 18, Kluwer Academic, 1988.
[5] X. Chen, S. Sankaranarayanan, Decomposed reachability analysis for nonlinear systems, in: 2016 IEEE Real-Time Systems Symposium, RTSS, 2016, pp. 13–24.
[6] D. Bresolin, P. Collins, L. Geretti, R. Segala, T. Villa, S. Gonzalez Zivanovic, A computable and compositional semantics for hybrid automata, in: Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control, HSCC '20, Association for Computing Machinery, New York, NY, USA, 2020, http://dx.doi.org/10.1145/3365365.3382202.
[7] Z. Han, X. Cai, J. Huang, Theory of Control Systems Described by Differential Inclusions, in: Springer Tracts in Mechanical Engineering, Springer-Verlag, 2016.
[8] H. Frankowska, M. Quincampoix, Viability kernels of differential inclusions with constraints: Algorithms and applications, J. Math. Syst. Estim. Control 1 (1991) 371–388.
[9] A. Puri, V. Borkar, P. Varaiya, ϵ-Approximation of differential inclusions, in: Proc. of the 34th IEEE Conference on Decision and Control, IEEE, New Orleans, LA, USA, 1995, pp. 2892–2897, http://dx.doi.org/10.1109/CDC.1995.478581, URL http://ieeexplore.ieee.org/document/478581/.
[10] P. Collins, D.S. Graca, Effective computability of solutions of differential inclusions the ten thousand monkeys approach, J. UCS 15 (6) (2009) 1162–1185.
[11] Y. Lin, M.A. Stadtherr, Validated solutions of initial value problems for parametric ODEs, Appl. Numer. Math. 57 (10) (2007) 1145–1162.
[12] X. Chen, Reachability Analysis of Non-Linear Hybrid Systems Using Taylor Models (Ph.D. thesis), Aachen University, 2015.
[13] A. Rauh, R. Westphal, H. Aschemann, E. Auer, Exponential enclosure techniques for the computation of guaranteed state enclosures in ValEncIA-IVP, Reliab. Comput. 19 (2013) 66–90.
[14] S. Harwood, P. Barton, Efficient polyhedral enclosures for the reachable set of nonlinear control systems, Math. Control Signals Systems 28 (8) (2016) http://dx.doi.org/10.1007/s00498-015-0153-2.
[15] M. Rasmussen, J. Rieger, K. Webster, Approximation of reachable sets using optimal control and support vector machines, J. Comput. Appl. Math. 311 (2017) 68–83.
[16] T. Kapela, P. Zgliczyski, A Lohner-type algorithm for control systems and ordinary differential inclusions, Discrete Contin. Dyn. Syst. Ser. B 11 (2) (2009) 365–385.
[17] M. Rungger, G. Reissig, Arbitrarily precise abstractions for optimal controller synthesis, in: 2017 IEEE 56th Annual Conference on Decision and Control, CDC, 2017, pp. 1761–1768.
[18] M. Althoff, O. Stursberg, M. Buss, Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization, in: 2008 IEEE 47th Annual Conference on Decision and Control, CDC, 2008, pp. 4042–4048.
[19] M. Dellnitz, S. Klus, A. Ziessler, A set-oriented numerical approach for dynamical systems with parameter uncertainty, SIAM J. Appl. Dyn. Syst. 16 (1) (2017) 120–138.
[20] M. Rungger, M. Zamani, Accurate reachability analysis of uncertain nonlinear systems, in: Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (Part of CPS Week), HSCC '18, 2018, pp. 61–70.

[21] M. Althoff, Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets, in: Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control, HSCC '13, ACM, New York, NY, USA, 2013, pp. 173–182, http://dx.doi.org/10.1145/2461328.2461358, URL http://doi.acm.org/10.1145/2461328.2461358.
[22] A. Rauh, A. Bourgois, L. Jaulin, J. Kersten, Ellipsoidal enclosure techniques for a verified simulation of initial value problems for ordinary differential equations, in: 2021 International Conference on Control, Automation and Diagnosis, ICCAD, 2021, pp. 1–6.
[23] A. Geraldes, L. Geretti, D. Bresolin, R. Muradore, P. Fiorini, L. Mattos, T. Villa, Formal verification of medical CPS: A laser incision case study, ACM Trans. Cyber-Phys. Syst. 2 (2018) 35:1–35:29, http://dx.doi.org/10.1145/3140237.
[24] D. Bresolin, L. Geretti, R. Muradore, P. Fiorini, T. Villa, Formal verification of robotic surgery tasks by reachability analysis, Microprocess. Microsyst. 39 (8) (2015) 836–842, http://dx.doi.org/10.1016/j.micpro.2015.10.006, URL http://www.sciencedirect.com/science/article/pii/S014193311500160X.
[25] D. Bresolin, L. Geretti, R. Muradore, P. Fiorini, T. Villa, Formal verification applied to robotic surgery, in: J. van Schuppen, T. Villa (Eds.), Coordination Control of Distributed Systems, in: Lecture Notes in Control and Information Sciences, vol. 456, Springer International Publishing, 2015, pp. 347–355.
[26] L. Geretti, R. Muradore, D. Bresolin, P. Fiorini, T. Villa, Parametric formal verification: the robotic paint spraying case study, in: Proceedings of the 20th IFAC World Congress, 2017, pp. 9248–9253.
[27] Ariadne: an open library for formal verification of cyber-physical systems, 2020, http://www.ariadne-cps.org.
[28] K. Makino, M. Berz, Taylor models and other validated functional inclusion methods, Int. J. Pure Appl. Math. 4 (4) (2003) 379–456.
[29] S. Zivanovic, P. Collins, Numerical solutions to noisy systems, in: 49th IEEE Conf. on Decision and Control, CDC, 2010, pp. 798–803, http://dx.doi.org/10.1109/CDC.2010.5717780.
[30] L. Geretti, S. Zivanovic Gonzalez, P. Collins, D. Bresolin, T. Villa, Rigorous continuous evolution of uncertain systems, in: 12th International Workshop on Numerical Software Verification (NSV'19), in: LNCS, vol. 11652, 2019, pp. 60–75.
[31] S.Z. Gonzalez, P. Collins, L. Geretti, D. Bresolin, T. Villa, Higher order method for differential inclusions, 2020, https://arxiv.org/pdf/2001.11330.
[32] G. Dahlquist, Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations, in: Transactions of the Royal Institute of Technology, Almqvist and Wiksells, 1959.
[33] E. Hairer, S.P. Norsett, G. Wanner, Solving Ordinary Differential Equations. I. Nonstiff Problems, in: Springer Series in Computational Mathematics, vol. 8, Springer-Verlag, 1987.
[34] S. Lozinskii, Error Estimates for the Numerical Integration of Ordinary Differential Equations, I, in: STL Trans. Series, Space Technology Laboratories, 1962, URL https://books.google.com/books?id=qCxDHQAACAAJ.
[35] G. Söderlind, The logarithmic norm. History and modern theory, BIT Numer. Math. 46 (3) (2006) 631–652, http://dx.doi.org/10.1007/s10543-006-0069-9.
[36] P. Collins, M. Niqui, N. Revol, A taylor function calculus for hybrid system analysis: Validation in COQ, in: NSV-3: Third International Workshop on Numerical Software Verification, 2010.
[37] W. Kühn, Rigorously computed orbits of dynamical systems without the wrapping effect, Computing 61 (1) (1998) 47–67.
[38] J.C. Sprott, Some simple chaotic jerk functions, Amer. J. Phys. 65 (6) (1997) 537–543.
[39] S.H. Strogatz, Nonlinear Dynamics and Chaos (Second Edition), in: Studies in Nonlinearity, CRC Press, 2014.
[40] L. Geretti, D. Bresolin, P. Collins, S. Zivanovic Gonzalez, T. Villa, Ongoing work on automated verification of noisy nonlinear systems with ariadne, in: Proc. of the 29th International Conference on Testing Software and Systems, ICTSS, 2017, pp. 313–319.