Reach-Avoid Verification for Time-varying Systems with Uncertain Disturbances

Ruiqi Hu, Kairong Liu and Zhikun She
SKLSDE, LMIB, and School of Mathematical Sciences, Beihang University, Beijing, China
{ by1809102, krliu, zhikun.she } @buaa.edu.cn

***Abstract*—In this work, we investigate the reach-avoid problem of a class of time-varying analytic systems with disturbances described by uncertain parameters. Firstly, by proposing the concepts of maximal and minimal reachable sets, we connect the avoidability and reachability with maximal and minimal reachable sets respectively. Then, for a given disturbance parameter, we introduce the evolution function for exactly describing the reachable set, and find a series representation of this evolution function with its Lie derivatives, which can also be regarded as a series function w.r.t. the uncertain parameter. Afterward, based on the partial sums of this series, over- and under-approximations of evolution function are constructed, which can be realized by interval arithmetics with designated precision. Further, we propose sufficient conditions for avoidability and reachability and design a numerical quantifier elimination based algorithm to verify these conditions; moreover, we improve the algorithm with a time-splitting technique. Finally, we implement the algorithm and use some benchmarks with comparisons to show that our methodology is both efficient and promising.**

***Index Terms*—Reachable set; Evolution function; Parametric disturbances; Reach-avoid verification.**

I. Introduction

Reach-avoid problem is a crucial issue in various realistic applications. It usually involves rigorously proving that a dynamical system, subject to initial conditions and uncertain disturbances, will eventually reach a region of the state-space while avoiding some unsafe set of states. Verification of real-world dynamical systems, such as autonomous cars [1] and unmanned aerial vehicles [2], is challenging for many reasons. First, all possible system behaviors, which might be an uncountable number of trajectories due to uncountable initial states, must be accounted for. This makes most simulation-based approaches insufficient, and thus formal methods are needed. Second, many practical systems evolve with complex nonlinear dynamics and even jumps, and are usually additionally affected by unknown but bounded uncertainties [3]. Finally, these systems often have high-dimensional state spaces [4], which is perhaps the most difficult challenge of all.

In this paper, we investigate the reach-avoid problem of a class of time-varying analytic systems with disturbances described by uncertain parameters. Specifically, we first connect the notion of the reachable set of systems with given disturbance parameters to the notions of minimal and maximal reachable sets, which were also introduced in [5], [6], [7]. Then, we introduce our reach-avoid problem of verifying whether the system can cover the target set while avoiding the obstacle set, which comes from the motivation to simultaneously and completely destroy the military targets while avoiding infrastructures and even interceptions. Note that our reach-avoid problem is defined as sweep-avoid property in [8] and different from most of the studies [9], [10], which consider whether the reachable set can be a subset of the target set while avoiding the obstacle set; moreover, our obstacle set and target set can be time-varying. Afterward, inspired by [11], we introduce the evolution function for exactly describing the reachable set with given parameter and find a series representation of the evolution function with its Lie derivatives, which can also be regarded as a series function w.r.t. the uncertain parameter. Using interval arithmetics, the remainder of partial sums for the series representation can be consecutively estimated, and if one remainder satisfies the designated precision, we can build over- and under-approximations with the range of this remainder and the corresponding partial sum. Moreover, based on algebraic representations for over-approximations of the maximal reachable set and under-approximations of the minimal reachable set, we propose sufficient conditions for avoidability and reachability, and design an algorithm based on numerical quantifier elimination [12] for verifying these conditions. We also improve the algorithm with time-splitting technique and the corresponding simplification procedure. Finally, five benchmarks, including non-polynomial systems and time-varying obstacle set, are presented to show the performance of our computational methodology with comparisons to **CORA** [13] and **Flow*** [14]; especially, we also compare our method with the method from [8] for Example 2.

The contributions of this paper can be summarized as:

1. We introduce the evolution function for the system with given disturbance parameter, whose Taylor expansion w.r.t. time can be used to exactly describe the reachable set, and connect it to the concepts of maximal and minimal reachable set of the uncertain system.
2. By using partial sums of the series expansion of evolution function, we provide a remainder estimation (RE) based methodology to obtain over- and under-approximations of evolution function. In this way, we can handle systems with rational and even trigonometric functions.
3. We design a numerical quantifier elimination based method for verifying the reach-avoid problem, which can handle time-varying obstacle set and target set (see Exam-

This work was supported by Beijing Natural Science Foundation under Grant Z180005. Zhikun She is the corresponding author.

ple 3). In addition, if the system is robustly (un)avoidable and (un)reachable, our algorithm can definitely terminate with "true(false)".

Related Work

Reachability analysis, which involves computing reachable sets, plays a fundamental role in reach-avoid verification of systems. Since the exact reachable sets are in general very hard to be obtained for nonlinear systems, over- and under-approximations are often computed for reach-avoid verification. During recent two decades, numerous methods have been proposed to over-approximate reachable sets [15], [16], [17], [18] and some methods have been proposed for under-approximations [19], [20], [21]. These methods often use Lagrangian method [22], [23], which follows the flow of system, or Eulerian method [24], which models the dynamics of system by looking at how it flows through fixed sets.

The Lagrangian methods generally use various set representations to compute the approximations of reachable set, such as zonotopes [25], polyhedra [26], sub-polyhedral [27] and ellipsoids [28]. These methods have also been applied to deal with a variety of uncertain dynamical systems. For example, based on ellipsoidal calculus, [29] studies the ellipsoidal estimates of reachable sets of the control systems which contain quadratic type of nonlinearities and uncertain matrix parameters; and [30] first uses mean-value extensions for computing under-approximations of ranges of vector fields under adversarial uncertainties with boxes, and then applies this box based method to under-approximate robust reachable sets of dynamical systems under time-varying uncertainties. Moreover, some of these methods have been further extended to deal with delay differential equations, for example, [31] proposes a scheme to compute over- and under- approximating flowpipes for delay differential equations with uncertain initial states and parameters, and [32] explores reachable set computation for a class of delay differential equations featuring a local homeomorphism property.

The Eulerian methods, generally based on the Hamilton-Jacobi equation [33], [34], are known to be more expressive to solve generalized reachability problems, and can be applied to handle differential games [35]. While Hamilton-Jacobi reachability is applicable to many small practical systems, the curse of dimensionality prevents its direct application to many larger systems. Fortunately, certain significant leaps have been made recently in overcoming this challenge, for example, [4] proposes a technique that decomposes the dynamics of a general class of nonlinear systems into subsystems, and [36] proposes to simplify system dynamics by treating state variables as disturbances. It is noteworthy that the Taylor method [15], [17], [19], as a typical Eulerian and efficient method, can be transformed into a Lagrangian one.

The above Lagrangian methods are mainly used for safety verification (dually, avoidability verification) in the literature, although they can also be used for set-based estimation, invariant set computation, controller synthesis, and so on [37]. For example, [38] uses ellipsoidal and support vector set representations to synthesize permissive safety preserving control laws for linear systems of larger state dimension; [7] uses an approximation based on ellipsoidal techniques to study the continual reachability set, and [39] further generalizes this approximation to characterize robust reachable sets or viability kernels and uses them to prove safety/viability of the system. Meanwhile, some of the above Eulerian methods have also been used for reach-avoid verification. For example, [40] proposes a set-boundary based method to verify reach-avoid problem of non-linear systems with parametric uncertainty under the assumption that the initial set is a compact set; [41] proposes a modified Hamilton-Jacobi equation in the form of a double-obstacle variational inequality, and proves that the zero sub-level set of its viscosity solution characterizes the capture basin for the target under the state constraints.

II. Problem Description

In this paper, we consider a class of time-varying analytic systems with disturbances described by uncertain parameters, i.e., systems of the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t, \mathbf{u}),\ \mathbf{x} \in \mathbb{D}, \mathbf{u} \in \prod_{i=1}^{m}[a_i, b_i], t \in [t_0{-}T, t_0{+}T], \quad (1)$$

where $\mathbf{x}$ is an n-dimensional vector, $\mathbf{f}(\mathbf{x}, t, \mathbf{u}) : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^n$ is an analytic real function, $\mathbb{D} \subseteq \mathbb{R}^n$ is a domain, $\mathbf{u}$ is an m-dimensional bounded disturbance parameter defined in $\mathcal{U} = \prod_{i=1}^{m}[a_i, b_i] \subseteq \mathbb{R}^m$, t is the time variable and the time interval $[t_0{-}T, t_0{+}T] \subseteq \mathbb{R}$ with $T > 0$ is well-defined. Letting $\mathbf{X}_0 \subseteq \mathbb{D}$ be an initial set, the *solution* of system (1) with given disturbance parameter $\mathbf{u_0} \in \mathbb{U}$ starting from the initial state $\mathbf{x}_0 \in \mathbf{X}_0$ and initial instant t_0 is denoted as $\phi_{\mathbf{u_0}}(t; (\mathbf{x}_0, t_0))$.

For system (1), we define the *reachable set* as follows.

Definition 1. *For system* (1) *with given disturbance parameter* $\mathbf{u_0} \in \mathbb{U}$*, we define the* reachable set *from the initial set* $\mathbf{X}_0$ *with the initial instant* t_0 *at instant* $t \in [t_0, t_0 + T]$ *as*

$$Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \{\mathbf{x} \in \mathbb{D} \mid \exists \mathbf{x}_0 \in \mathbf{X}_0, \mathbf{x} = \phi_{\mathbf{u_0}}(t; (\mathbf{x}_0, t_0))\}.$$

Moreover, the reachable set *from the initial set* $\mathbf{X}_0$ *with the initial instant* t_0 *within time interval* $[t_0, t_0 + T]$ *is defined as*

$$\boldsymbol{Reach}^{\mathbf{u_0},T}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \bigcup_{t\in[t_0,t_0+T]} Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mathbf{X}_0,t_0)}.$$

We also define $\boldsymbol{Reach}^{\mathbf{u_0},-T}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \bigcup_{t\in[t_0-T,t_0]} Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$.

Since we cannot specify the disturbance $\mathbf{u_0}$, following [5], we further define maximum and minimum reachable sets of uncertain system (1) as follows.

Definition 2. *For given system* (1)*, the maximal reachable set* $Reach^{max\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$ *from the initial set* $\mathbf{X}_0$ *with initial time* t_0 *at instant* t *is defined as*

$$Reach^{max\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \{\mathbf{x} \in \mathbb{D} \mid \exists \mathbf{u} \in \mathcal{U}, \exists \mathbf{x}_0 \in \mathbf{X}_0, \mathbf{x} = \phi_{\mathbf{u}}(t; (\mathbf{x}_0, t_0))\};$$

the minimal reachable set $Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$ *from* $\mathbf{X}_0$ *with initial time* t_0 *at instant* t *is defined as*

$$Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \{\mathbf{x} \in \mathbb{D} \mid \forall \mathbf{u} \in \mathcal{U}, \exists \mathbf{x}_0 \in \mathbf{X}_0, \mathbf{x} = \phi_{\mathbf{u}}(t; (\mathbf{x}_0, t_0))\}.$$

Moreover, the maximal reachable sets from $\mathbf{X}_0$ *within time interval* $[t_0, t_0+T]$ *is defined as:* $\boldsymbol{Reach}^{max\ T}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \bigcup_{t\in[t_0,t_0+T]} Reach^{max\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$. *Similarly, we can also define:* $\boldsymbol{Reach}^{max\ -T}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \bigcup_{t\in[t_0-T,t_0]} Reach^{max\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$.

It is clear that $Reach^{max\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \bigcup_{\mathbf{u_0}\in\mathcal{U}} Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$ and $Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \bigcap_{\mathbf{u_0}\in\mathcal{U}} Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$. Based on Definition 2, we propose the reach-avoid problem for uncertain system (1) with given target set RA_T and obstacle set RA_O.

Definition 3. *For system* (1) *with given initial set* $\mathbf{X}_0$ *and initial time* t_0*, the obstacle set* RA_O *is avoidable within time interval* $[t_0, t_0+T]$ *iff*

$$\boldsymbol{Reach}^{max\ T}_{\mathbf{f},(\mathbf{X}_0,t_0)} \bigcap RA_O = \emptyset;$$

and the target set RA_T *is reachable within time interval* $[t_0, t_0+T]$ *iff there exists a* $t \in [t_0, t_0+T]$ *such that*

$$Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} \supseteq RA_T.$$

Remark 1. *If the obstacle set* $RA_O(t)$ *and target set* $RA_T(t)$ *are time-varying, the obstacles set* $RA_O(t)$ *is avoidable within time interval* $[t_0, t_0+T]$ *iff the constraint* $\forall t \in [t_0, t_0+T][Reach^{max\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} \bigcap RA_O(t) = \emptyset]$ *holds; and the target set* $RA_T(t)$ *is reachable within time interval* $[t_0, t_0+T]$ *iff the constraint* $\exists t \in [t_0, t_0+T][Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} \supseteq RA_T(t)]$ *holds. However, it is necessary to mention that we may encounter the case that there exists a* t *such that* $Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)} = \emptyset$.

Obviously, our reach-avoid problem is different from the most studies [40], [9], [10] since we consider covering all of the target set while avoiding the obstacle set, which is naturally useful in path-planning problems [8], e.g., designing paths for simultaneously and completely destroying the military targets while avoiding infrastructures and even interceptions. The key issue for verifying the reach-avoid problem is to compute $Reach^{max\ T}_{\mathbf{f},(\mathbf{X}_0,t_0)}$ and $Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$. By Definitions 1 and 2, the computation of $Reach^{max\ T}_{\mathbf{f},(\mathbf{X}_0,t_0)}$ and $Reach^{min\ t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$ can be converted to finding $Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$ for all $\mathbf{u_0} \in \mathcal{U}$ and $t \in [t_0, t_0+T]$. Inspired by [11], by using the analytic solution of the system, we introduce the *evolution function* (EF) as follows, to characterize $Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mathbf{X}_0,t_0)}$.

Definition 4. *For an analytic function* $g : \mathbb{R}^n \to \mathbb{R}$ *and a system* (1) *with given parameter* $\mathbf{u_0}$*, the* evolution function *of the system with* $g(\cdot)$ *is defined as*

$$Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) = g(\phi_{\mathbf{u_0}}(t_0;(\mathbf{x},t))),$$

where $(\mathbf{x},t) \in \mathbb{D}\times[t_0, t_0+T]$.

Since $\mathbf{f}(\mathbf{x},t,\mathbf{u_0})$ in (1) is analytic and thus local Lipschitz w.r.t. $\mathbf{x}$, then based on Definition 4, we have that for all $(\mathbf{x},t) \in \mathbb{D}\times[t_0, t_0+T]$,

$$Evo_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t) = g(\mathbf{x}_0), \tag{2}$$

indicating that $Evo_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t)$ does not change with t for all $\mathbf{x}_0 \in \mathbb{D}$. An important proposition, connecting evolution function and reachable set, will be given directly after introducing *sub-zero level set*.

Definition 5. *The* sub-zero level set $\mu(v(\cdot))$ *of function* $v : \mathbb{R}^n \to \mathbb{R}$ *is defined as:*

$$\mu(v(\cdot)) = \{\mathbf{x} \in \mathbb{R}^n \mid v(\mathbf{x}) \le 0\}.$$

Remark 2. *Throughout this paper, if the initial set* $\mathbf{X}_0$ *can be represented by a sub-zero level set of a function* $g(\cdot)$*, i.e.* $\mathbf{X}_0 = \mu(g)$*, we use* $Reach^{\mathbf{u_0},t}_{\mathbf{f},(g,t_0)}$ *and* $\boldsymbol{Reach}^{\mathbf{u_0},T}_{\mathbf{f},(g,t_0)}$ *as aliases of* $Reach^{\mathbf{u_0},t}_{\mathbf{f},(\mu(g),t_0)}$ *and* $\boldsymbol{Reach}^{\mathbf{u_0},T}_{\mathbf{f},(\mu(g),t_0)}$*, respectively.*

Proposition 1. *For an analytic function* $g : \mathbb{R}^n \to \mathbb{R}$ *and an uncertain system* (1) *with given parameter* $\mathbf{u_0} \in \mathcal{U}$*, we have* $Reach^{\mathbf{u_0},t}_{\mathbf{f},(g,t_0)} = \mu(Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t))$.

Proof. According to (2), $Evo_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t) \le 0 \Leftrightarrow g(\mathbf{x}_0) \le 0$, and thus Proposition 1 can be easily obtained. □

Proposition 1 shows that the computation of reachable set of system (1) with given parameter $\mathbf{u_0}$ can be converted to computations of evolution function of the system defined by its analytic solution and the corresponding sub-zero level set.

III. Over- and Under- Approximations of Reachable Sets

From Proposition 1, for computing the reachable set, we can alternatively compute the evolution function defined by the analytic solution of the system. However, the analytic solution is hard to get in general. Instead of computing evolution function by directly using the analytic solution, following the work in [11], we in this section present an explicit series representation of the evolution function using its Lie derivatives, and then introduce a methodology based on estimating the remainders of partial sums of the series representation to compute over- and under-approximations of evolution functions, arriving at over- and under-approximations of reachable sets. For this purpose, we first define over- and under- approximations of set and function as follows.

Definition 6. *For sets* S_1 *and* S_2*,* S_1 *is called as* over- (under-, resp.) approximation *of* S_2 *iff* $S_1 \supseteq (\subseteq) S_2$*. Moreover, for two* n*-dimensional scalar functions* $f_1(\mathbf{x})$ *and* $f_2(\mathbf{x})$ *and a set* S*,* $f_1(\mathbf{x})$ *is called as* over- (under-, resp.) approximation *of* $f_2(\mathbf{x})$ *over* S *if* $f_1(\mathbf{x}) \le (\ge) f_2(\mathbf{x})$*,* $\forall \mathbf{x} \in S$.

A. t-expansion of Evolution Function

Recall that for a vector field $\mathbf{c}(\mathbf{x})$ defined from $\mathbb{R}^n$ to $\mathbb{R}^n$, the *Lie derivative* of function $h : \mathbb{R}^n \to \mathbb{R}$ w.r.t. $\mathbf{c}(\mathbf{x})$ is given by $\mathcal{L}_{\mathbf{c}}(h) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \cdot \mathbf{c}(\mathbf{x})$. Then, the ith-order Lie derivative of $h(\cdot)$ w.r.t. $\mathbf{c}(\mathbf{x})$, written as $\mathcal{L}^{(i)}_{\mathbf{c}}(h)$, is defined as $\mathcal{L}^{(0)}_{\mathbf{c}}(h) = h$ and $\mathcal{L}^{(i)}_{\mathbf{c}}(h) = \mathcal{L}_{\mathbf{c}}(\mathcal{L}^{(i-1)}_{\mathbf{c}}(h))$ for $i \ge 1$. Moreover, by Definition 4, for $Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$, where $(\mathbf{x},t) \in \mathbb{D}\times[t_0, t_0+T]$, its first-order derivative $Evo'_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$ w.r.t. t is $\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}|_{\phi_{\mathbf{u_0}}(t_0;(\mathbf{x},t))} \cdot (-\mathbf{f}(\mathbf{x},t,\mathbf{u_0}))$ since $\frac{\partial \phi_{\mathbf{u_0}}(t_0;(\mathbf{x},t))}{\partial t} = -\mathbf{f}(\mathbf{x},t,\mathbf{u_0})$, which indicates that

$$Evo'_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \cdot (-\mathbf{f}(\mathbf{x},t,\mathbf{u_0})) = \mathcal{L}_{-\mathbf{f}}(g).$$

Similarly we can compute the $(n+1)$th-order derivative of $Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$ $(n \geq 1)$ as

$$\begin{aligned} Evo^{(n+1)}_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) &= \frac{dEvo^{(n)}_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)}{dt} \\ &= \frac{\partial Evo^{(n)}_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)}{\partial \mathbf{x}} \cdot \mathbf{f}(\mathbf{x},t,\mathbf{u_0}) + \frac{\partial Evo^{(n)}_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)}{\partial t}, \end{aligned}$$

arriving at $Evo^{(n+1)}_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x}_0,t_0) = \mathcal{L}^{(n+1)}_{-\mathbf{f}}(g)$ since $Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x}_0,t_0) = g(\mathbf{x_0})$. Thus the Taylor expansion w.r.t. t at $t=t_0$ (namely the *t-expansion*) of $Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$ can be expressed as

$$\begin{aligned} Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) &= \sum_{i=0}^{+\infty} Evo^{(i)}_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x}_0,t_0)\frac{(t-t_0)^i}{i!} \\ &= \sum_{i=0}^{+\infty} \mathcal{L}^i_{-\mathbf{f}}(g)\frac{(t-t_0)^i}{i!}. \end{aligned} \tag{3}$$

Equivalently, we can write (3) as

$$Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) = \sum_{i=0}^{+\infty} \frac{\mathcal{M}^i_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)}{i!}(-t+t_0)^i, \tag{4}$$

where for all $(\mathbf{x},t) \in \mathbb{D}\times[t_0,t_0+T]$, $\mathcal{M}^i_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$ is defined inductively as $\mathcal{M}^0_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) = g(\mathbf{x})$ and $\mathcal{M}^{i+1}_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) = \frac{\partial \mathcal{M}^i_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)}{\partial \mathbf{x}} \cdot \mathbf{f}(\mathbf{x},t,\mathbf{u_0}) + \frac{\partial \mathcal{M}^i_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)}{\partial t}$. Moreover, we denote $Evo^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$ as its Nth partial sum of *t-expansion*, i.e.,

$$Evo^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) \equiv \sum_{i=0}^{N} \frac{(-t+t_0)^i}{i!}\mathcal{M}^i_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t). \tag{5}$$

B. RE based Methodology for Approximations

In this subsection, we provide a remainder estimation based methodology to compute over- and under-approximations of reachable sets. For this, we introduce the following theorem.

Theorem 1. *For a system* (1) *with given parameter* $\mathbf{u_0} \in \mathcal{U}$*, an analytic function* $g : \mathbb{R}^n \to \mathbb{R}$*, and an arbitrary but fixed* $N \in \mathbb{N}$*, we have that for all* $(\mathbf{x},t) \in \mathbb{D}\times[t_0,T+t_0]$*,*

$$Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) = Evo^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) + Rem^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t), \tag{6}$$

where $Rem^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$ *is defined to be* $\int_0^{t-t_0} \frac{(t-t_0-r)^N}{N!}(-1)^{N+1}\mathcal{M}^{N+1}_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t-r;(\mathbf{x},t)),t-r)dr$.

Proof. For given $\mathbf{u_0} \in \mathbb{U}$ and arbitrary $N \in \mathbb{N}$, by considering the derivative of $Evo^N_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t)$, where $\mathbf{x}_0 \in \mathbb{D}$ and $t \in [t_0,t_0+T]$, we have that: $\frac{dEvo^N_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t)}{dt} = \frac{(-t+t_0)^N}{N!}\mathcal{M}^{N+1}_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t)$. Thus, for any $(\mathbf{x}_0,t) \in \mathbb{D}\times[t_0,t_0+T]$,

$$\begin{aligned} &Evo^N_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t) \\ =&Evo^N_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t_0;(\mathbf{x}_0,t_0)),t_0) \\ &+\int_{t_0}^{t} \frac{(-s+t_0)^N}{N!}\mathcal{M}^{N+1}_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(s;(\mathbf{x}_0,t_0)),s)ds \\ =&Evo_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) - Rem^N_{\mathbf{f},g,\mathbf{u_0}}(\phi_{\mathbf{u_0}}(t;(\mathbf{x}_0,t_0)),t). \end{aligned}$$

Therefore, we completed the proof of the theorem. □

Moreover, for all $(\mathbf{x},t,\mathbf{u}) \in \mathbb{D}\times[t_0,T+t_0]\times\mathbb{U}$, where $\mathbf{u}$ is the uncertain parameter, according to Equations (4) and (5), we can construct a function $Tr^N(\mathbf{x},t,\mathbf{u})$ as

$$Tr^N(\mathbf{x},t,\mathbf{u}) = \sum_{i=0}^{N} \frac{(-t+t_0)^i}{(i)!}M^i(\mathbf{x},t,\mathbf{u}), \tag{7}$$

where for all $(\mathbf{x},t,\mathbf{u}) \in \mathbb{D}\times[t_0,T+t_0]\times\mathbb{U}$, $M^i(\mathbf{x},t,\mathbf{u})$ is defined inductively as $M^0(\mathbf{x},t,\mathbf{u}) = g(\mathbf{x})$ and $M^i(\mathbf{x},t,\mathbf{u}) = \frac{dM^{i-1}(\mathbf{x},t,\mathbf{u})}{dt}$. Since $\mathbf{u}$ satisfies $\frac{du}{dt} = 0$, for a system (1) with any given parameter $\mathbf{u_0} \in \mathcal{U}$ and $N \in \mathbb{N}$, we have that $M^N(\mathbf{x},t,\mathbf{u_0}) = \mathcal{M}^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$, i.e.

$$Tr^N(\mathbf{x},t,\mathbf{u_0}) = Evo^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t). \tag{8}$$

Clearly, according to equations (7) and (8), $Evo^N_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)$ can be regarded as a function of the uncertain parameter $\mathbf{u}$, and thus, according to equation (4), $Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)$ can also be regarded as a series function of the uncertain parameter $\mathbf{u}$.

Due to Definition 6, to arrive at over- and under- approximations of evolution function, we need to fix S first. Here, since we consider the reachable set with all $\mathbf{u} \in \mathbb{U}$ and their over- and under-approximations, S should be a bound of the maximal reachable set with time interval $[t_0,T+t_0]$. Thus, due to Theorem 1 and Equation (8), once we can find the upper and lower bounds of $M^{N+1}(\mathbf{x},t,\mathbf{u})$, then we can estimate $Rem^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t)$. However, for bounding $M^{N+1}(\phi_{\mathbf{u_0}}(t-r;(x,t)),t-r,\mathbf{u_0})$, for all $\mathbf{u_0} \in \mathbb{U}$, $\phi_{\mathbf{u_0}}(t-r;(x,t))$ is required. Since $x \in S$ and $r \in [0,t-t_0]$, then $\mathbf{y} = \phi_{\mathbf{u_0}}(t-r;(\mathbf{x},t)) \in Reach^{\mathbf{u_0},t-r}_{\mathbf{f},(S,T+t_0)}$ and thus for all $\mathbf{u_0} \in \mathbb{U}$, $\mathbf{y} \in \mathbf{Reach}^{max\ -T}_{\mathbf{f},(S,T+t_0)}$. So the backward reachable set is additionally required. Correspondingly, we have the following theorem for describing the over- and under-approximations of evolution function.

Theorem 2. *For given uncertain system* (1) *and analytic function* $g : \mathbb{R}^n \to \mathbb{R}$*, assume that* S *and* S' *are two compact sets of states such that* $\mathbb{D} \supseteq S \supseteq \boldsymbol{Reach}^{max\ T}_{\mathbf{f},(g,t_0)}$ *and* $S' \supseteq \boldsymbol{Reach}^{max\ -T}_{\mathbf{f},(S,T+t_0)}$*. For given degree* $N \in \mathbb{N}$*, if we can find two real numbers* L_{N+1} *and* U_{N+1}*, satisfying* $\forall(\mathbf{x},t,\mathbf{u}) \in S'\times[t_0,t_0+T]\times\mathbb{U}$*,* $L_{N+1} \leq (-1)^{N+1}M^{N+1}(\mathbf{x},t,\mathbf{u}) \leq U_{N+1}$ *, then for all* $\mathbf{u} \in \mathbb{U}$*, we have that*

1. $Over(\mathbf{x},t,\mathbf{u}) = Tr^N(\mathbf{x},t,\mathbf{u}) + L_{N+1}\frac{(t-t_0)^{N+1}}{(N+1)!}$ / $Under(\mathbf{x},t,\mathbf{u}) = Tr^N(\mathbf{x},t,\mathbf{u}) + U_{N+1}\frac{(t-t_0)^{N+1}}{(N+1)!}$ *is over- / under- approximation of* $Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)$ *over* S*;*
2. *the precision for the approximations are bounded by* $(U_{N+1}-L_{N+1})\frac{(t-t_0)^{N+1}}{(N+1)!}$.

Proof. To find over- and under-approximations of evolution function over S, we need to estimate upper and lower bounds of $Rem^N_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)$. According to the hypothesis of L_{N+1} and U_{N+1}, for all $(\mathbf{y},r,\mathbf{u}) \in S'\times[0,t-t_0]\times\mathbb{U}$, $L_{N+1} \leq (-1)^{N+1}M^{N+1}(\mathbf{y},t-r,\mathbf{u}) \leq U_{N+1}$, which implies that

for any $\mathbf{u_0} \in \mathbb{U}$, $L_{N+1}\frac{(t-t_0)^{N+1}}{(N+1)!} \leq Rem^N_{\mathbf{f},g,\mathbf{u_0}}(\mathbf{x},t) \leq U_{N+1}\frac{(t-t_0)^{N+1}}{(N+1)!}$. Together with Equations (6) and (8), we have

$$\begin{aligned} Tr^N(\mathbf{x},t,\mathbf{u}) + L_{N+1}\frac{(t-t_0)^{N+1}}{(N+1)!} &\leq Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t) \\ &\leq Tr^N(\mathbf{x},t,\mathbf{u}) + U_{N+1}\frac{(t-t_0)^{N+1}}{(N+1)!} \end{aligned} \tag{9}$$

for all $(\mathbf{x},t,\mathbf{u}) \in S \times [t_0, t_0+T] \in \mathbb{U}$, which is the result *1*.

Sequentially, from (9) we can immediately have that $Under(\mathbf{x},t,\mathbf{u}) - Over(\mathbf{x},t,\mathbf{u}) = (U_{N+1} - L_{N+1})\frac{(t-t_0)^{N+1}}{(N+1)!}$ for all $(\mathbf{x},t) \in S \times [t_0, t_0+T]$, and thus the result *2* holds. □

Now, let us consider the Hausdorff metric between a reachable set and its over- or under-approximations theoretically generated by Theorem 2.

Definition 7. *The* Hausdorff metric *between sets S_1 and S_2 is defined as $H(S_1,S_2) = \max(\xi(S_1,S_2),\xi(S_2,S_1))$, where $\xi(S_1,S_2) = \max_{\mathbf{x}\in S_1}\min_{\mathbf{y}\in S_2}\|\mathbf{x}-\mathbf{y}\|$.*

From the proof of Theorem 2, it is obvious that the Hausdorff metric between $\mu(Evo_{\mathbf{f},g,\mathbf{u}}(\cdot,t))$ with either over- or under-approximation generated by Theorem 2 is bounded by $H(\mu(Over(\cdot,t,\mathbf{u})),\mu(Under(\cdot,t,\mathbf{u})))$. Then, similar to Proposition 3 in [42], we have the following theorem.

Theorem 3. *For a system* (1) *with given $\mathbf{u_0}$, an analytic but nonconstant function $g : \mathbb{R}^n \to \mathbb{R}$, and sets S and S' satisfying $\mathbb{D} \supseteq S \supseteq$ **Reach**$^{max\ T}_{\mathbf{f},(g,t_0)}$ and $S' \supseteq$ **Reach**$^{max\ -T}_{\mathbf{f},(S,T+t_0)}$, suppose that $M^{N+1}(\mathbf{x},t,\mathbf{u_0}) = o(\frac{N!}{T^N})$ uniformly in S'. Moreover, for arbitrary but fixed $r > 0$, let $\bar{x}_t = \max_{\mathbf{x}\in\{\mathbf{x}|Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)=0\}} \| \frac{\partial Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)}{\partial \mathbf{x}} \|$, $\varepsilon_t = \max_{\mathbf{x}\in\{\mathbf{x}|Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)=0\}} \min_{l\in[0,r]} \frac{1}{T} Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x} + \frac{l}{\bar{x}_t}\frac{\partial Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)}{\partial \mathbf{x}}, t)$ and $\varepsilon^* = \min_{t\in[0,T]}\varepsilon_t$. Then, for any $\varepsilon \in [0, \varepsilon^* T]$, there exists a degree N with the corresponding $Over(\mathbf{x},t,\mathbf{u_0})$ and $Under(\mathbf{x},t,\mathbf{u_0})$ defined by Theorem 2 such that for all $(\mathbf{x},t) \in S \times [t_0, t_0+T]$, $|Under(\mathbf{x},t,\mathbf{u_0}) - Over(\mathbf{x},t,\mathbf{u_0})| \leq \varepsilon$, and*

$$H(\mu(Over(\cdot,t,\mathbf{u_0})),\mu(Under(\cdot,t,\mathbf{u_0}))) \leq r.$$

C. Algorithm for Approximations of Reachable Set

For given uncertain system (1), analytic $g(\cdot)$, time interval $[t_0, t_0+T]$ and designated precision ϵ_1, we in this subsection attempt to use Theorem 2 to design an algorithm for computing over- and under-approximations of reachable sets for all $\mathbf{u} \in \mathbb{U}$. According to Theorem 2, we need to determine the S and S' first. For this, starting from a box B containing the initial set $\mu(g)$, we call **CORA**-2021 to get an over-approximation (named as `Bound`) of the maximal reachable set as S, and continue to call **CORA** to get an over-approximation (named as `Interval`) of the backward maximal reachable set of S as S'. **CORA** [43] can compute the over-approximation of reachable set within given time-interval and return the resulting boxes, which is really suitable for our purpose. Then, according to this S', we can iteratively increase the degree N of $Tr^N(\mathbf{x},t,\mathbf{u})$ and estimate the lower bound L_{N+1} and upper bound U_{N+1} of $(-1)^{N+1}M^{N+1}(\mathbf{x},t,\mathbf{u})$ in $S' \times [t_0, t_0+T] \times \mathcal{U}$ with interval arithmetic in **CORA** until $U_{N+1} - L_{N+1} \leq \frac{\epsilon_1\cdot(N+1)!}{T^{N+1}}$, and then generate over- and under-approximations of $Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x},t)$ with all $\mathbf{u} \in \mathbb{U}$ according to Theorem 2 with the designated precision ϵ_1. Note that if for all $\mathbf{u} \in \mathbb{U}$, $M^{N+1}(\mathbf{x},t,\mathbf{u}) = o(\frac{N!}{T^N})$ uniformly in S', then for any $\epsilon_1 > 0$, there exists an N, such that L_{N+1} and U_{N+1} estimated by **CORA** can satisfy that $U_{N+1} - L_{N+1} \leq \frac{\epsilon_1\cdot(N+1)!}{T^{N+1}}$. Consequently, we have Algorithm 1 to simultaneously compute over- and under-approximations of evolution function with given precision ϵ_1.

Algorithm 1

Input: $\mathbf{f}(\mathbf{x},t,\mathbf{u})$, $g(\mathbf{x})$, B, t_0, T, ϵ_1, $[a_i,b_i]$;
Output: $Over(\mathbf{x},t,\mathbf{u})$, $Under(\mathbf{x},t,\mathbf{u})$, and `Bound`.

1: Call **CORA**(`f`, B, T) to find `Bound` as S;
2: Call **CORA**(`f`, `Bound`, $-T$) to find `Interval` as S';
3: $M(\mathbf{x},t,\mathbf{u}) \Leftarrow g(\mathbf{x})$; $Tr(\mathbf{x},t,\mathbf{u}) \Leftarrow g(\mathbf{x})$; $N \Leftarrow 0$;
4: **while not** $U - L \leq \frac{\epsilon_1\cdot(N+1)!}{T^{N+1}}$ **do**
5: $\quad M(\mathbf{x},t,\mathbf{u}) \Leftarrow \frac{\partial M(\mathbf{x},t,\mathbf{u})}{\partial \mathbf{x}} \cdot \mathbf{f}(\mathbf{x},t,\mathbf{u}) + \frac{\partial M(\mathbf{x},t,\mathbf{u})}{\partial t}$;
6: $\quad Tr(\mathbf{x},t,\mathbf{u}) \Leftarrow Tr(\mathbf{x},t,\mathbf{u}) + \frac{(-t+t_0)^{N+1}}{(N+1)!}M(\mathbf{x},t,\mathbf{u})$; $N \Leftarrow N+1$;
7: $\quad$ Compute L and U for $M(\mathbf{x},t,\mathbf{u})$ in `Interval` $\times [t_0, t_0+T] \times \prod_{i=1}^m [a_i,b_i]$;
8: Let $Tail_o(t) = L\frac{(t-t_0)^{N+1}}{(N+1)!}$ and $Tail_u(t) = U\frac{(t-t_0)^{N+1}}{(N+1)!}$;
9: $Over(\mathbf{x},t,\mathbf{u}) \Leftarrow Tr(\mathbf{x},t,\mathbf{u}) + Tail_o(t)$; $Under(\mathbf{x},t,\mathbf{u}) \Leftarrow Tr(\mathbf{x},t,\mathbf{u}) + Tail_u(t)$;
10: **return** $Over(\mathbf{x},t,\mathbf{u})$, $Under(\mathbf{x},t,\mathbf{u})$, `Bound`.

Moreover, according to Theorem 3, for arbitrary but fixed $r > 0$, we can find ϵ_1 and the corresponding $Over(\mathbf{x},t,\mathbf{u})$ and $Under(\mathbf{x},t,\mathbf{u})$ with $|Over(\mathbf{x},t,\mathbf{u}) - Under(\mathbf{x},t,\mathbf{u})| \leq \epsilon_1$ such that $H(\mu(Over(\cdot,t,\mathbf{u})),\mu(Under(\cdot,t,\mathbf{u}))) \leq r$. So, we can make the Hausdorff metric between over- and under-approximations of the reachable set arbitrarily small since the precision ϵ_1 is user-provided.

IV. Avoidance-Reachability Verification

In this section, we will introduce a methodology based on over- and under-approximations of the reachable set to verify our reach-avoid problem. In detail, we in Subsection IV-A propose a numerical quantifier elimination based verification approach; further, we in Subsection IV-B use the time-splitting technique to improve the verification approach.

A. Numerical Quantifier Elimination based Verification

Based on Theorem 2, Proposition 1, and Algorithm 1, $\{\mathbf{x} \in S \mid Over(\mathbf{x},t,\mathbf{u}) \leq 0\}$ and $\{\mathbf{x} \in S \mid Under(\mathbf{x},t,\mathbf{u}) \leq 0\}$ are over- and under- approximations of $Reach^{\mathbf{u},t}_{\mathbf{f},(g,t_0)}$ with the designated precision ϵ_1 respectively, where S is the output `Bound` of Algorithm 1. Thus, we can obtain sufficient conditions for avoidability and reachability in $[t_0, t_0+T]$ as follows.

Theorem 4. *For given time-varying obstacle set $RA_O = \{\mathbf{x} \mid O(\mathbf{x},t) \leq 0\}$ and time-varying target set $RA_T = \{\mathbf{x} \mid T(\mathbf{x},t) \leq 0\}$ and the output of Algorithm 1,*

1. *If the constraint* $\forall t \in [t_0, t_0+T], \forall \mathbf{u} \in \prod_{i=1}^{m}[a_i, b_i], \forall \mathbf{x} \in S, [Over(\mathbf{x}, t, \mathbf{u}) \leq 0 \Rightarrow O(\mathbf{x}, t) > 0]$ *holds, then the obstacle set* RA_O *is avoidable.*
2. *If the constraint* $\exists t \in [t_0, t_0+T], \forall \mathbf{u} \in \prod_{i=1}^{m}[a_i, b_i], \forall \mathbf{x} \in S, [T(\mathbf{x}, t) \leq 0 \Rightarrow Under(\mathbf{x}, t, \mathbf{u}) \leq 0]$ *holds, then the target set* RA_T *is reachable.*

Then, for given system (1), initial set $\mathbf{X}_0 = \mu(g)$, time interval $[0, T]$, obstacle set$O(\mathbf{x}, t)$, and target set$T(\mathbf{x}, t)$, according to Theorem 4, we can design an algorithm for reach-avoid verification as follows. We first call Algorithm 1 with the precision ϵ_1 for $Over(\mathbf{x}, t, \mathbf{u})$, $Under(\mathbf{x}, t, \mathbf{u})$, and `Bound`, and let `Bound` be the box S used for Theorem 4. Then we use **RSolver** [12] with specified error bound ϵ_2 to deal with the constraints in Theorem 4. Note that **RSolver** can solve quantified constraints, that is, formulae in the first order predicate language over the reals, and returns 'true', 'false', and 'unknown' with corresponding intervals according to a user-provided error bound. Further, we process the outputs of **RSolver** as follows: if the interval for 'false' is not empty, we terminate the loop and return 'the system may be unavoidable/unreachable'; if the interval for 'false' is empty but the interval for 'unknown' is not empty, implying that the precision is not satisfied, we reduce both ϵ_1 and ϵ_2 and repeat the previous steps; if the output of **RSolver** is only 'true', we terminate the algorithm and return 'the obstacle/target set is avoidable/reachable'.

Consequently we design Algorithm 2 to simultaneously verify reachability and avoidability of the system. Therein, to avoid confusion, we use ϵ_1 and ϵ_2 for avoidable verification, but $\epsilon_1^{'}$ and $\epsilon_2^{'}$ for reachable verification.

Remark 3. *It is noteworthy that if* $M^{N+1}(\mathbf{x}, t, \mathbf{u}) = o(\frac{N!}{T^N})$ *uniformly in* $S^{'}$ *and the system is robustly avoidable, there must exist a number* $r > 0$ *satisfying that* $\min_{\mathbf{x} \in Reach^{max\ t}_{\mathbf{f},(g,t_0)}} \min_{\mathbf{y} \in \mu(O(\cdot,t))} \|\mathbf{x}, \mathbf{y}\|_2 \geq r$, $\forall t \in [0, T]$. *Moreover, for this* r*, according to Theorem 3 and Algorithm 2, we can certainly find* ϵ_1*, such that for any given* $\mathbf{u_0} \in \mathbb{U}$*, the corresponding* $Over(\mathbf{x}, t, \mathbf{u_0})$ *and* $Under(\mathbf{x}, t, \mathbf{u_0})$ *satisfy that*

$$H(\mu(Over(\cdot, t, \mathbf{u_0})), \mu(Under(\cdot, t, \mathbf{u_0}))) \leq r/2,$$

implying that $H(\mu(Over(\cdot, t, \mathbf{u_0})), Reach^{\mathbf{u_0}, T}_{\mathbf{f},(g,t_0)}) \leq r/2$*; meanwhile, we can find* ϵ_2*, such that* $\epsilon_2 \leq r/2$*. Thus, if* $\mathcal{M}^{N+1}_{\mathbf{f},g,\mathbf{u}}(\mathbf{x}, t) = o(\frac{N!}{T^N})$ *uniformly in* $S^{'}$ *and the system is robustly avoidable, the return of* ***RSolver*** *with (*$Over(\mathbf{x}, t, \mathbf{u})$*,* $O(\mathbf{x}, t)$*,* ϵ_2*) for the avoidable condition in Theorem 4 will be definitely 'true'. Similarly, if the system is robustly reachable, the return of* ***RSolver*** *with (*$Under(\mathbf{x}, t, \mathbf{u})$*,* $T(\mathbf{x}, t)$*,* ϵ_2*) for the avoidable condition will be definitely 'true'. As a result, if for all* $\mathbf{u} \in \mathbb{U}$*,* $\mathcal{M}^{N+1}_{\mathbf{f},g,\mathbf{u}}(\mathbf{x}, t) = o(\frac{N!}{T^N})$ *uniformly in* $S^{'}$ *and the system is robustly avoidable and reachable, Algorithm 2 can definitely terminate with 'true'.*

Remark 4. *For given time-varying obstacle set* $RA_O = \{O(\mathbf{x}, t) \leq 0\}$ *and target set* $RA_T = \{T(\mathbf{x}, t) \leq 0\}$*, if the constraint* $\exists t \in [t_0, t_0+T], \exists \mathbf{u} \in \mathcal{U}, \exists \mathbf{x} \in S, [Under(\mathbf{x}, t, \mathbf{u}) \leq 0 \Rightarrow O(\mathbf{x}, t) \leq 0]$ *holds, then the obstacle set* RA_O *is unavoidable; if the constraint* $\forall t \in [t_0, t_0+T], \exists \mathbf{u} \in \mathcal{U}, \exists \mathbf{x} \in S, [Over(\mathbf{x}, t, \mathbf{u}) \geq 0 \Rightarrow T(\mathbf{x}, t) < 0]$ *holds, then the target set* RA_T *is unreachable. Clearly, we can use a method similar to Algorithm 2 for the falsification of reach-avoid problem.*

Algorithm 2

Input: $\mathbf{f}(\mathbf{x}, t, \mathbf{u})$, $g(\mathbf{x})$, B, T, ϵ_1, ϵ_2, $\mathcal{U}$, $O(\mathbf{x}, t)$, $T(\mathbf{x}, t)$;
Output: 'Avoidable', 'Reachable'.

```
1: Initialize Avo ⇐ 'unk'; Rch ⇐ 'unk'; ϵ1' ⇐ ϵ1; ϵ2' ⇐ ϵ2;
2: while Avo='unk' do
3:    Call Algo. 1 with ϵ1 for Over(x, t, u) and Bound;
4:    Call RSolver with (T, Over(x, t, u), O(x), U, Bound,
      ϵ2) for the avoidable condition in Theorem 4;
5:    if the interval for 'false' ≠ ∅ then
6:       Avo ⇐ 0
7:    else if the interval for 'unknown' ≠ ∅ then
8:       ϵ1 ⇐ 0.1ϵ1; ϵ2 ⇐ 0.5ϵ2;
9:    else
10:      Avo ⇐ 1;
11: while Rch='unk' do
12:   Call Algo. 1 with ϵ1' for Under(x, t, u) and Bound;
13:   Call RSolver with (T, Under(x, t, u), T(x), U,
      Bound, ϵ2') for the reachable condition in Theorem 4;
14:   if the interval for 'false' ≠ ∅ then
15:      Rch ⇐ 0
16:   else if the interval for 'unknown' ≠ ∅ then
17:      ϵ1 ⇐ 0.1ϵ1; ϵ2 ⇐ 0.5ϵ2;
18:   else
19:      Rch ⇐ 1;
20: if Avo = 1 ⋀ Rch = 1 then
21:   return 'Avoidable' and 'Reachable'.
```

B. Improvements of Algorithm 2

Note that for large time intervals, Algorithm 1 usually requires a higher degree N and thus a larger number of terms to ensure accuracy, which makes it hard for **RSolver** to find solutions. Hence, for performance improvement, we modify Algorithm 2 to Algorithm 3 with the time-splitting technique, separating the given time interval $[0, T]$ into K parts with equal length $\Delta T = T/K$. One observation is that for small time bound ΔT, $Evo^{N}_{\mathbf{f},g,\mathbf{u}}(\mathbf{x}, t)$ can approximate $Evo_{\mathbf{f},g,\mathbf{u}}(\mathbf{x}, t)$ well in low degree N, so that the computational complexity of **RSolver** can be considerably reduced; meanwhile, the reachable set can stay close to the initial set so that the corresponding S may be small, making it easier for **RSolver** to find solutions. Correspondingly, different from Lines 1–2 in Algorithm 1, Lines 2–3 in Algorithm 3 successively call **CORA** to get an enclosure for the reachable set in each time segment based on the previous enclosure, denoted as `Bound`, and Lines 5–6 in Algorithm 3 inversely call **CORA** for each time segment to calculate the over-approximation of the backward reachable set of previous over-approximation, denoted as `Interval`. Here `Bound` and `Interval` satisfy that: $\mathbf{Reach}^{max\ \Delta T}_{\mathbf{f},\mathtt{Bound}(i-1)} \subseteq \mathtt{Bound}(i)$ for all $2 \leq i \leq \mathtt{K}$,

Algorithm 3

Input: $\mathbf{f}(\mathbf{x},t,\mathbf{u})$, $g(\mathbf{x})$, B, ΔT, K, r, $\Delta\epsilon_1$, ϵ_2, $\mathcal{U}$, $O(\mathbf{x},t)$, $T(\mathbf{x},t)$.

Output: 'Avoidable', 'Reachable'.

1: Initialize $Init(\mathbf{x},t,\mathbf{u}) \Leftarrow g(\mathbf{x})$; $Tail_o^{'} \Leftarrow 0$; $Tail_u^{'} \Leftarrow 0$; Avo $\Leftarrow$ 'unk'; Rch $\Leftarrow$ 'unk';
2: **for** i=1:K **do**
3: Call **CORA**(f, B, ΔT) to update B, $\texttt{Bound}(i) \Leftarrow B$;
4: $\texttt{Interval}(K+1) \Leftarrow \texttt{Bound}(K)$;
5: **for** i=K:1 **do**
6: Call **CORA**(f, B, $-\Delta T$) to update B, $\texttt{Interval}(j) \Leftarrow B$;
7: **for** $i = 1 : K$ **do**
8: Call Lines 3–8 in Algo. 1 with ($Init(\mathbf{x},t,\mathbf{u})$, ΔT, $(i-1)\Delta T$, $\texttt{Interval}(i)$, $r\Delta\epsilon_1$) to obtain $Tr(\mathbf{x},t,\mathbf{u})$, $Tail_o$ and $Tail_u$;
9: Simplify $Tr(\mathbf{x},t,\mathbf{u})$ with precision $\epsilon_1^i = \frac{(1-r)\Delta\epsilon_1}{2}$;
10: $\texttt{Over}^i \Leftarrow Tr(\mathbf{x},t,\mathbf{u}) + Tail_o^{'} - \epsilon_1^i + Tail_o$; $\texttt{Under}^i \Leftarrow Tr(\mathbf{x},t,\mathbf{u}) + Tail_u^{'} + \epsilon_1^i + Tail_u$;
11: $Tail_o^{'} \Leftarrow Tail_o^{'} - \frac{(1-r)\Delta\epsilon}{2} + Tail_o(i\Delta T)$; $Tail_u^{'} \Leftarrow Tail_u + \frac{(1-r)\Delta\epsilon}{2} + Tail_u(i\Delta T)$;
12: $Init(\mathbf{x}) \Leftarrow Tr(\mathbf{x}, i\Delta T, \mathbf{u})$;
13: Build piecewise over- and under- approximations function $\texttt{Over} := \texttt{Over}^j(\mathbf{x},t,\mathbf{u})$, $\texttt{Under} := \texttt{Under}^j(\mathbf{x},t,\mathbf{u})$, $\forall t \in [\frac{j-1}{\mathrm{K}}\mathrm{T}, \frac{j}{\mathrm{K}}\mathrm{T}], 1 \le j \le \mathrm{K}$;
14: **for** i=1:K **do**
15: **while** Avo = 'unk; **do**
16: Call **RSolver** with ($[(i-1)\Delta T, i\Delta T]$, $Over(i)$, $O(\mathbf{x})$, $\mathcal{U}$, $\texttt{Bound}(i)$, ϵ_2) for the avoidable condition;
17: **if** the interval for 'false' $\neq \emptyset$ **then**
18: Avo $\Leftarrow$ 0;
19: **else if** the interval for 'unknown' $\neq \emptyset$ **then**
20: $\epsilon_1 \Leftarrow 0.1\epsilon_1$; $\epsilon_2 \Leftarrow 0.5\epsilon_2$; Back to Line 7;
21: **while** Rch = 'unk'; **do**
22: Call **RSolver** with ($[(i-1)\Delta T, i\Delta T]$, $Under(i)$, $T(\mathbf{x})$, $\mathcal{U}$, $\texttt{Bound}(i)$, ϵ_2) for the reachable condition;
23: **if** the interval for 'false' $= \emptyset$ **then**
24: **if** the interval for 'unknown' $= \emptyset$ **then**
25: Rch $\Leftarrow$ 1;
26: **else**
27: $\epsilon_1 \Leftarrow 0.1\epsilon_1$; $\epsilon_2 \Leftarrow 0.5\epsilon_2$; Back to Line 7;
28: **if** Avo $\neq 0 \bigwedge$ Rch $= 1$ **then**
29: **return** 'Avoidable' and 'Reachable'

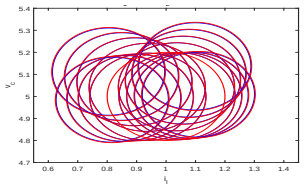
$\texttt{Bound(K)} = \texttt{Interval(K+1)}$, and $\mathbf{Reach}^{max\ -\Delta T}_{\mathbf{f},\texttt{Interval}(i+1)} \subseteq \texttt{Interval}(i)$ for all $1 \le i \le \mathrm{K}$. Note that for the ith segment, the initial set is $\mathbf{X}_{i-1} = \mu(Tr(\cdot,\Delta T,\mathbf{u}))$, $\mathbf{Reach}^{max\ \Delta T}_{\mathbf{f},\mathbf{X}_{i-1}} \subseteq \texttt{Bound}(i)$ and $\mathbf{Reach}^{max\ \Delta T}_{-\mathbf{f},\texttt{Bound}(i)} \subseteq \texttt{Interval}(i)$; moreover, $\texttt{Bound}(i)$ and $\texttt{Interval}(i)$ will be the enclosure S and $S^{'}$ used for Theorem 2 respectively. Next, starting from $g(\mathbf{x})$, Line 8 in Algorithm 3 repeatedly calls Lines 3–8 in Algorithm 1 for $Tr(\mathbf{x},t,\mathbf{u})$, $Tail_o$ and $Tail_u$ in each time segment. As a result, for each segment, Line 8 in Algorithm 3 returns approximations with precision $r\Delta\epsilon_1$. It is worth mentioning that during time-splitting, if we directly use the result obtained in the previous iteration as the initial of the successive iteration, the scale of $Tr(\mathbf{x},t,\mathbf{u})$ will grow rapidly when the iteration continues. As a remedy, we design a simplification procedure, as shown in Line 9 in Algorithm 3, to reduce the scale of the outputs from Algorithm 1 in each segment, reducing the computational complexity of the successive iteration and sacrificing some precisions in the final over- and under-approximations. Specifically, we firstly use interval arithmetics to find upper bounds for the absolute values of all terms in $Tr(\mathbf{x},t,\mathbf{u})$ in $S' \times [\frac{j-1}{\mathrm{K}}\mathrm{T}, \frac{j}{\mathrm{K}}\mathrm{T}] \times \mathcal{U}$, and then sort these upper bounds in ascending order; afterwards, we gather the terms until the sum of the corresponding upper bounds of the absolute values of the collected terms exceeds $\frac{(1-r)\Delta\epsilon_1}{2}$ and then remove the previous collected terms, sacrificing the precision $\frac{(1-r)\Delta\epsilon_1}{2}$. Finally, by adding $-\frac{(1-r)\Delta\epsilon_1}{2}/\frac{(1-r)\Delta\epsilon_1}{2}$ to the over/under approximation (see Lines 12-13), we can return approximations with precision ϵ_1.

Consequently, compared to Algorithm 2, Algorithm 3 can work on a longer time bound with the help of the time-splitting technique and guarantee the effectiveness of **RSolver** due to the simplification procedure.

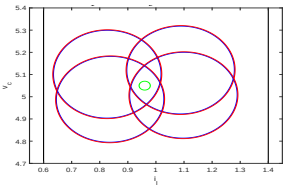
Remark 5. *Naturally, we can extend our method to systems with uncertain piecewise disturbances.*

V. Examples with Comparisons and Discussions

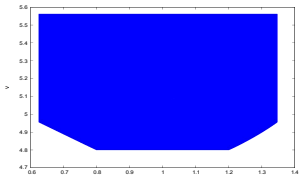
In this section, we demonstrate our method with comparisons on five examples whose computations are all performed on a Laptop 1.8GHz Intel Core i7 (4 cores) and 8Gb of RAM. For all these examples, we set $\epsilon_1 = 10^{-2}$ and $\epsilon_2 = 0.03$ as our initial precision for Algorithm 3. Some data of the results are listed in Table I, and the figures of obtained over/under-approximation are also shown. Note that due to the uncountable assignments of the uncertain parameter $\mathbf{u}$, it is very difficult to get the image of the reachable set. Therefore, by sampling certain $\mathbf{u}$, we list part of the images to intuitively show that our reach-avoid verification is correct. Therein, the black/green lines represent the boundaries of the obstacle/target set and the red/blue lines represent the boundaries of over-/under- approximation of $Reach^{\mathbf{u},t}_{\mathbf{f},g}$. Moreover, for all these five examples, we compare our method with **CORA** and **Flow*** on the precision for over-approximations of maximal reachable sets; especially, for Example 2, which has also been used in [8], we additionally compare our method with the method from [8] for both over-approximations of maximal reachable sets and under-approximations of minimal reachable sets. The results, given by **CORA**, **Flow*** and the method from [8], are also listed. Therein, we set options.timeStep = 0.05, options.zonotopeOrder = 10 and options.taylorTerms = 5 in **CORA**, adaptive orders $\{min4, max16\}$, remainder estimation 1e-4 and fixed steps 0.01 in **Flow*** and sampling-time = 0.02, order = 3 in method from [8]. Note that, we do not compare our method with **CORA** and **Flow*** for under-approximations of minimal reachable sets since it is difficult to adapt **CORA**-2021 and **Flow***-2.1.0 for our systems; and we



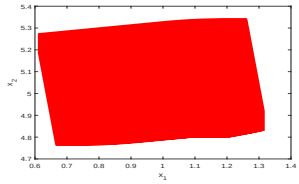
(a) Approx. of $Reach_{\mathbf{f},g}^{\mathbf{u},t}$ for specific $\mathbf{u}$ and t by our method



(b) Results at t=2 for specific $\mathbf{u}$ by our method

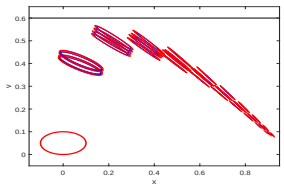


(c) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **Flow***

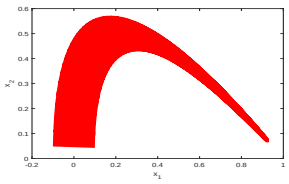


(d) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **CORA**

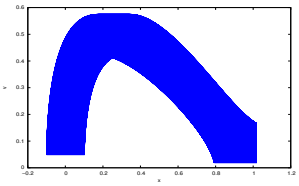
Fig. 1: Results of Example 1.



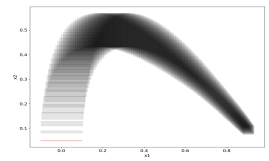
(a) Approx. of $Reach_{\mathbf{f},g}^{\mathbf{u},t}$ for specific $\mathbf{u}$ and t by our method



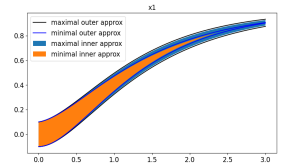
(b) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **CORA**



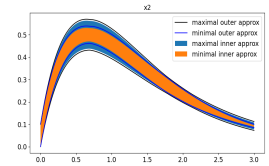
(c) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **Flow***



(d) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by [8]



(e) Evolutions of $x(t)$ w.r.t t by [8]



(f) Evolutions of $v(t)$ w.r.t t by [8]

Fig. 2: Results of Example 2.

do not compare our method with the method from [8] for other examples since it is also hard to adapt the method from [8] for them. Moreover, for fair comparisons, we also experimented with higher parameter settings for **CORA**-2021, **Flow***-2.1.0, and the method from [8] (see Tables II, III and IV).

Example 1. *Consider a DC-DC boost converter system [44]:*

$$\dot{\mathbf{x}} = A(\mathbf{u})\mathbf{x} + B(\mathbf{u}),$$

where $\mathbf{x}(t) = [i_l(t), v_c(t)]$, $i_l(t)$ *is the inductor current,* $v_c(t)$ *is the capacitor voltage,* $\mathbf{u} = [r_0, v_s]$ *is the disturbance, and* $A(\mathbf{u})$ *and* $B(\mathbf{u})$ *are described by*

$$A(\mathbf{u}) = \frac{1}{200r_0+1}\begin{bmatrix} \frac{-(220r_0+1)}{60} & -40r_0/3 \\ 100r_0/7 & -20/7 \end{bmatrix}, B(\mathbf{u}) = \begin{bmatrix} v_s/3 \\ 0 \end{bmatrix}.$$

We set the initial set $\{(i_l, v_c) \mid (i_l-1)^2 + (v_c-5)^2 \leq 0.04\}$, *and assume that* $\mathbf{u} \in [1,5] \times [0.8, 1.2]$. *Then, by Algorithm 3, with the obstacles set* $RA_O = \{-(i_l - 0.9)^2 \leq -0.16\}$ *and the target set* $RA_T = \{(i_l - 0.96)^2 + (v_c - 5.05)^2 + (t-2)^2 \leq 0.02^2\}$, *we can verify the system is avoidable and reachable within time interval* $[0, 2]$. *The obtained over/under-approximations of reachable sets corresponding to* $\mathbf{u} = (3 \pm 2, 1 \pm 0.2)$ *at* $t = 0, 0.2, \ldots, 2$ *are shown in Figure 1a, and we also show the target set, the obstacle set, and the obtained over/under-approximation results at* $t = 2$ *in Figure 1b.*

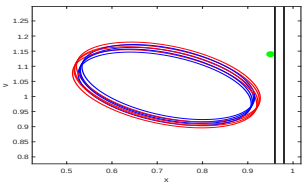
The result given by ***CORA*** *with initial set* $[0.8, 1.2] \times [4.8, 5.2]$ *is shown in Figure 1c and* ***Flow**** *has no result for adaptive orders* $\{min4, max16\}$ *and fixed steps 0.01. However, after resetting adaptive orders* $\{min32, max64\}$ *and fixed steps 0.001 in* ***Flow****, the result given by* ***Flow**** *is shown in Figure 1d. From Figures 1a, 1c and 1d, we can see that our result is comparable with* ***CORA*** *and* ***Flow****.*

Example 2. *Consider a PD-controller [8], [45], controlling position* x *and velocity* v *of a car by adjusting its acceleration depending on the current distance to a reference position* p_r*:*

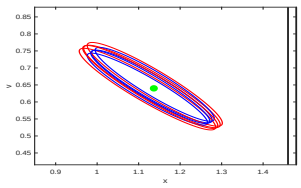
$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ -K_p(x - p_r) - K_d v \end{bmatrix}.$$

We set the initial set $\mathbf{X}_0 = \{(x, v) \mid x^2 + (2v - 0.1)^2 - 0.01 \leq 0\}$ *and* $p_r = 1$, *and assume that* $\mathbf{u} = (K_p, K_d) \in [1.95, 2.05] \times [2.95, 3.05]$. *Then, by Algorithm 3, with the obstacle set* $RA_O = \{(v - 0.6)^2 - 0.01^2 \leq 0\}$, *we can verify that the system is avoidable within time interval* $[0, 3]$. *Note that the minimal reachable set is empty set, so we do not consider our reachability verification for this example. The obstacle set, and the obtained over/under-approximation results of reachable sets corresponding to* $\mathbf{u} = (2 \pm 0.05, 3 \pm 0.05)$ *at* $t = 0, 0.3, \ldots, 3$ *are shown in Figure 2a.*

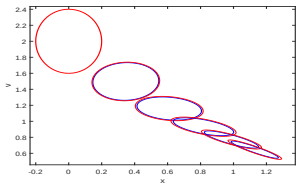
We also list the results for over-approximations of maximal reachable sets with the initial set $\{(x, v) : x \in [-0.1, 0.1], v = 0.05\}$ *obtained by* ***CORA****-2021,* ***Flow****-2.1.0, and the method from [8] in Figures 2b, 2c, and 2d, respectively. From Figures 2a, 2b, 2c, and 2d, we can easily see that our method is slightly better than* ***CORA****,* ***CORA*** *is distinguishingly better than the method from [8], and the method from [8] is distinguishingly better than* ***Flow****. Note that we also use* $[-0.1, 0.1] \times [0, 0.1]$ *as initial for the method from [8] here and show the results of under-approximations of minimal reachable sets in Figures 2e and 2f, which show that the results of our method and the method from [8] are comparable.*



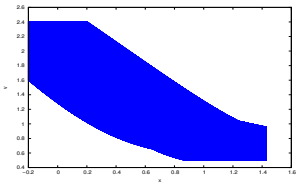
(a) Results at $t = 0.5$ for specific $\mathbf{u}$ by our method



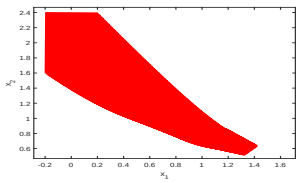
(b) Results at $t = 1$ for specific $\mathbf{u}$ by our method



(c) Approximations of $Reach_{\mathbf{f},g}^{\mathbf{u},t}$ for specific $\mathbf{u}$ and t by our method



(d) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **Flow***



(e) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **CORA**

Fig. 3: Results of Example 3.

Example 3. *We modified Example 2 to consider time-varying reference positions, obstacles set, and target set, where*

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{p_r} \end{bmatrix} = \begin{bmatrix} v \\ -K_p(x - p_r) - K_d v \\ cos(\frac{\pi t}{4}) \end{bmatrix}.$$

We set the initial set $\mathbf{X}_0 = \{(x, v, p_r) \mid x^2 + (v-2)^2/4 + 4(p_r - 1)^2 - 0.04 \leq 0\}$ *and assume that* $\mathbf{u} = (K_p, K_d) \in [1.98, 2.02] \times [2.48, 2.52]$. *By Algorithm 3, with the obstacles set* $RA_O = \{(x - t - 0.47)^2 - 0.01^2 \leq 0\}$ *and the target set* $RA_T = \{(p_r - x - 0.5)^2 + (v - 0.64)^2 - 0.005^2 \leq 0\}$, *we can verify the system is avoidable and reachable. The target set, the obstacle set and the obtained over/under-approximations of reachable sets corresponding to* $\mathbf{u} = (2.00 \pm 0.02, 2.50 \pm 0.02)$ *at* $t = 0, 0.2, \ldots, 1$ *in Figure 3c.*

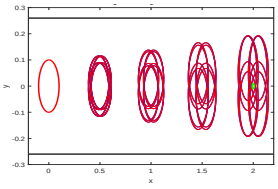
The results given by ***Flow**** *and* ***CORA*** *with initial set* $[-0.2, 0.2] \times [1.6, 2.4] \times [0.8, 1.2]$ *are shown in Figures 3d and 3e. Clearly, in Figure 3c,* x *obtained by our method is less than 1.4, implying that our method can produce higher precision than* ***CORA*** *and even higher precision than* ***Flow****. Since the obstacle set is time-varying, results of* ***Flow**** *and* ***CORA*** *cannot be directly used for avoidance verification.*

Example 4. *We consider a Dubins car [33] described by the following dynamic system with parameters:*

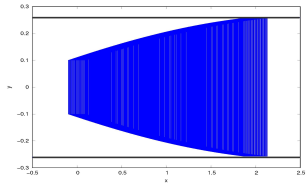
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} vcos\theta + u_x \\ vsin\theta + u_y \\ \omega + u_\theta \end{bmatrix}.$$

Let the initial set $\mathbf{X}_0 = \{(x, y, \theta) \mid x^2 + y^2 + \theta^2 - 0.01 \leq 0\}$, *the speed as* $v = 2$, *the control as* $\omega = -y/2$, *and assume* $u = (u_x, u_y, u_\theta) \in [-0.02, 0.02] \times [-0.02, 0.02] \times [-0.01, 0.01]$. *Then, by using Algorithm 3, with the obstacles set* $RA_O = \{(0.26 - y)(0.26 + y) \leq 0\}$ *and the target set* $RA_T = \{(x-2)^2 + y^2 + \theta^2 - 0.01^2 \leq 0\}$, *we can verify that the system is avoidable and reachable within time interval* $[0, 1]$. *Figure 4a shows the target set, the obstacle set and the obtained over/under-approximation results of reachable sets corresponding to* $\mathbf{u} = (\pm 0.02, \pm 0.02, \pm 0.01)$ *at* $t = 0, 0.25, 0.5, 0.75, 1$ *on the plane with* $\theta = 0$.

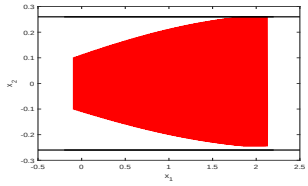
The results given by ***Flow**** *and* ***CORA*** *with initial* $[-0.1, 0.1] \times [-0.1, 0.1] \times [-0.1, 0.1]$ *are shown in Figures 4b and 4c. Clearly,* $y(t) \geq -0.2$ *for all* $t \in [0, 1]$ *in Figure 4a, while the over-approximations given by* ***CORA*** *and* ***Flow**** *both intersect with the obstacle set.*



(a) Approximations of $Reach_{\mathbf{f},g}^{\mathbf{u},t}$ for specific $\mathbf{u}$ and t by our method



(b) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **Flow***



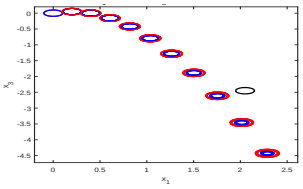
(c) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **CORA**

Fig. 4: Results of Example 4.

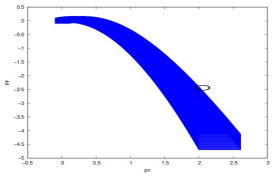
Example 5. *We modified an 8-dimensional non-polynomial system [45] with uncertain disturbances* D_x, D_y *and* D_ϕ:

$$\begin{bmatrix} \dot{p_x} \\ \dot{v_x} \\ \dot{p_y} \\ \dot{v_y} \\ \dot{\phi} \\ \dot{\omega} \\ \dot{T_1} \\ \dot{T_2} \end{bmatrix} = \begin{bmatrix} v_x \\ -\frac{C_D^v v_x}{m} \\ v_y \\ \frac{-mg - C_D^v v_y}{m} \\ \omega \\ -\frac{C_D^\phi \omega}{I_{yy}} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ D_x \\ 0 \\ D_y \\ 0 \\ D_\phi \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{-\sin\phi}{m} & \frac{-\sin\phi}{m} \\ 0 & 0 \\ \frac{\cos\phi}{m} & \frac{\cos\phi}{m} \\ 0 & 0 \\ \frac{l}{I_{yy}} & \frac{l}{I_{yy}} \\ 0 & \alpha \\ -\alpha & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix},$$

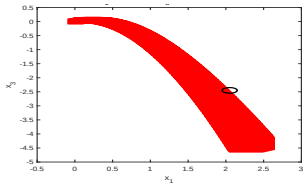
where p_x, p_y, *and* ϕ *represent the horizontal, vertical, rotational positions of the quadrotor,* v_x, v_y, *and* ω *represent the corresponding velocities,* T_1 *and* T_2 *are input thrusts*



(a) Approximations of $Reach_{\mathbf{f},g}^{\mathbf{u},t}$ for specific $\mathbf{u}$ and t by our method



(b) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **Flow***



(c) Over-approximation of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **CORA**

Fig. 5: Results of Example 5.

TABLE I: Data of Results for Examples.

	RT(s)	K	N	Terms	ϵ_1	ϵ_2	A.	R.
Ex.1	140	10	3	84	10^{-3}	0.015	T.	T.
Ex.2	1981	10	13	3077	10^{-2}	0.03	T.	F.
Ex.3	256	4	9	781	10^{-2}	0.03	T.	T.
Ex.4	827	2	5	1472	10^{-4}	0.0075	T.	T.
Ex.5	4404	5	7	9175	10^{-2}	0.03	T.	T.

exerted on either end of the quadrotor, and $\alpha = 1$. *For the coefficients in the system, we choose* $C_D^v = 0.1$ *for translational drag,* $m = 5$ *for the vehicles mass,* $g = 9.8$ *for gravity,* $C_D^\phi = 0.1$ *for rotational drag,* $I_{yy} = 10$ *for the moment of inertia,* $l = 0.5$ *and* $\alpha = 1$. *We set the initial set as* $\mathbf{X}_0 = \{\mathbf{x} = (p_x, v_x, \ldots, T_1, T_2) \mid \|\mathbf{x} - (0, 2, 0, 1, \frac{\pi}{6}, 0.1, 10, -10)^T\|_2 \leq 0.1\}$ *and let* $\mathbf{u} = (D_x, D_y, D_\phi) \in [-0.02, 0.02] \times [-0.02, 0.02] \times [-0.02, 0.02]$. *Then, with the obstacles set* $RA_O = \{(p_x - 2.05)^2 + (p_y + 2.45)^2 - 0.01 \leq 0\}$ *and the target set* $RA_T = \{\|\mathbf{x} - (2.29, 2.78, -4.425, -10.35, 0.31, -0.82, -3, -13)^T\|_2 \leq 0.005\}$, *Algorithm 3 can verify that the system is avoidable and reachable within interval* $[0, 1]$. *Figure 5a shows the target set, the obstacle set and the obtained over/under-approximations of reachable sets corresponding to* $\mathbf{u} = (\pm 0.02, \pm 0.02, \pm 0.02)$ *at* $t = 0, 0.1, \ldots, 1$ *onto the corresponding* $p_x - p_y$ *planes defined by letting* $(v_x, v_y, \phi, \omega, T_1, T_2)$ *equal to the simulated trajectory with* $(D_x, D_y, D_\phi) = (0, 0, 0)$ *starting from* $\mathbf{x}_0 = (0, 2, 0, 1, \frac{\pi}{6}, 0.1, 10, -10)^T$.

The results given by ***Flow**** *and* ***CORA*** *with the smallest box containing* $\mathbf{X}_0$ *and the obstacle set are shown in Figures 5b and 5c. Neither of* ***Flow**** *and* ***CORA*** *can verify that the system is avoidable, which shows that our method has higher precision than* ***CORA*** *and* ***Flow****.

In Table I, we list certain data obtained by our method, i.e. the running times (RT), the number of time segments (K), the maximal order of expansion (N), the maximum number of terms (Term) in each segment of $Over(\mathbf{x}, t, \mathbf{u})$ or $Under(\mathbf{x}, t, \mathbf{u})$, final precisions ϵ_1 and ϵ_2, and the verification results for avoidability (A.) and reachability (R.). Clearly, for polynomial systems with low degree and trigonometric systems with low dimension, our method shows a good performance. This is because the obtained $Over(\mathbf{x}, t, \mathbf{u})$ and $Under(\mathbf{x}, t, \mathbf{u})$ have a low degree and few terms such that **RSolver** can deal with them efficiently. However, since Algorithm 3 needs to increase time segments to handle longer time intervals (see Example 2) and increase the degree of expansion to achieve the higher precision requirement (see Example 4) or deal with a higher dimensional system (see Example 5), these will definitely lead to an increase in the number of terms for the approximations as well as an increase in the running time. Moreover, it is noteworthy that we do not need extra calculation when dealing with time-varying target set and obstacle set (see Example 3).

TABLE II: Parameters Setting for **CORA**

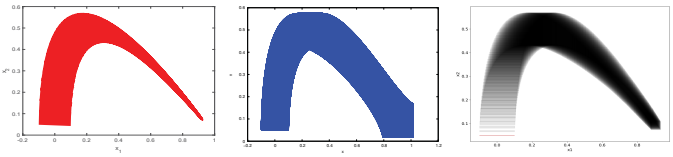
CORA	options.timeStep	.zonotopeOrder	.taylorTerms
	0.05/0.01/0.005/0.001	10/30/50	5/50/100

TABLE III: Parameters Setting for **Flow***

Flow*	fixed steps	adaptive orders
	0.001/0.0005	16/32/64

TABLE IV: Parameters Setting for method in [8]

method in [8]	sampling-time	order
	0.02/0.01/0.005	3/30/50



(a) Over-approx. of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **CORA** (b) Over-approx. of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by **Flow*** (c) Over-approx. of $\mathbf{Reach}_{\mathbf{f},g}^{max\ T}$ by [8]

Fig. 6: Results of Example 2 with Higher Parameters Setting.

Moreover, for fair comparisons, we also experimented with all the combinations of parameter settings listed in Tables II and III for **CORA** and **Flow*** for all examples, and Table IV for the method in [8] for Example 2. However, the results did not change significantly. To intuitively understand this, we show the results of Example 2 with options.timeStep=0.001, options.zonotopeOrder=50 and options.taylorTerms=100 for **CORA**, fixed steps=0.0005 and adaptive orders=64 for **Flow***, and sampling-time=0.005 and order=50 for the method in [8] in Figure 6. Clearly, from Figures 2, 5 and 6, we can see that to some extent, the method in [8] has higher precision than **Flow***, **CORA** has higher precision than the method in [8], and our method has higher precision than **CORA**.

VI. Conclusion

In this paper, for a class of time-varying systems with uncertain disturbances, we first introduced the evolution function for exactly describing the reachable set and then obtained the over-

and under- approximations of the reachable set by a remainder estimation based methodology. Afterwards, we proposed a numerical quantifier elimination based methodology for verifying our reach-avoid problem. Especially, if the system is robustly avoidable and reachable, our algorithm can definitely terminate with "true". Finally, we confirmed the efficacy and precision of our method by five benchmarks with comparisons.

In the future, we will investigate the reach-avoid verification of switched and even hybrid systems with time-varying disturbances. Moreover, it is also interesting to study the control synthesis problem for avoidance-guaranteed reachability.

REFERENCES

[1] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. In *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1094–1099, 2015.

[2] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment. In *Proc. of AAIA Guid., Navigat. Control Conf. Exhibit*, pp. 1–20, 2007.

[3] A. B. Kurzhanski and T. F. Filippova. On the Theory of Trajectory Tubes — A Mathematical Formalism for Uncertain Dynamics, Viability and Control. In *Advances in Nonlinear Dynamics and Control: A Report from Russia*, pp. 122–188, 1993

[4] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin. Decomposition of Reachable Sets and Tubes for a Class of Nonlinear Systems. *IEEE Trans. Automat. Contr. (TAC)*, 63(11): 3675–3688, 2018.

[5] I. M. Mitchell. Comparing Forward and Backward Reachability as Tools for Safety Analysis. In *HSCC'07*, pp. 428–443, 2007.

[6] C. J. Tomlin, J. Lygeros, and S. S. Sastry. A Game Theoretic Approach to Controller Design for Hybrid Systems. *Proceedings of the IEEE,*, 88(7): 949–970, 2000.

[7] S. Kaynama, M. Oishi, I. M. Mitchell, and G. A. Dumont. The Continual Reachability Set and Its computation Using Maximal Reachability Techniques. In *IEEE Conference on Decision and Control (CDC)*, pp. 6110–6115, 2011.

[8] E. Goubault and S. Putot. Inner and Outer Reachability for the Verification of Control Systems. In *HSCC'19*, pp. 11–22, 2019.

[9] K. Margellos and J. Lygeros. Hamilton-Jacobi Formulation for Reach-Avoid Differential Games. *IEEE Trans. Automat. Contr. (TAC)*, 56(8): 1849–1861, 2011.

[10] O. Bokanowski, N. Forcadel, and H. Zidani. Reachability and Minimal Times for State Constrained Nonlinear Problems without Any Controllability Assumption. *SIAM Journal on Control and Optimization*, 48(7): 4292–4316, 2010.

[11] M. Li and Z. She. Over- and Under-Approximations of Reachable Sets with Series Representations of Evolution Functions. *IEEE Trans. Automat. Contr. (TAC)*, 66(3): 1414–1421, 2021.

[12] S. Ratschan. Efficient Solving of Quantified Inequality Constraints over the Real Numbers. *ACM Transactions on Computational Logic*, pp. 723–748, 2006.

[13] [Online]. Available: $https://tumcps.github.io/CORA/$.

[14] [Online]. Available: $https://flowstar.org/dowloads/$.

[15] M. Althof. Reachability Analysis of Nonlinear Systems Using Conservative Polynomialization and Non-convex Sets In *HSCC'13*, pp. 173–182, 2013.

[16] T. Dang, O. Maler, and R. Testylier. Accurate Hybridization of Nonlinear Systems. In *HSCC'10*, pp. 11–20, 2010.

[17] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Taylor Model Flowpipe Construction for Non-linear Hybrid Systems. In *RTSS'33*, pp. 183–192, 2012.

[18] N. S. Nedialkov, K. Jackson, and G. Corliss. Validated Solutions of Initial Value Problems for Ordinary Differential Equations. *Applied Mathematics and Computation Volume*, 105(1): 21–68, 1999.

[19] X. Chen, S. Sankaranarayanan, and E. Ábrahám. Under-approximate Flowpipes for Non-linear Continuous Systems. In *Formal Methods in Computer-Aided Design*, pp. 59–66, 2014.

[20] B. Xue, Z. She, and A. Easwaran. Under-Approximating Backward Reachable Sets by Polytopes. In *CAV*, pp. 457–476, 2016.

[21] E. Goubault and S. Putot. Forward Inner-approximated Reachability of Non-linear Continuous Systems. In *HSCC'17*, pp. 1–10, 2017.

[22] A. Girard, C. Le Guernic, and O. Maler. Efficient Computation of Reachable Sets of Linear Time-invariant Systems with Inputs. In *HSCC'06*, pp. 257–271, 2006.

[23] T. Dang, O. Maler, and R. Testylier. Accurate Hybridization of Nonlinear Systems. In *HSCC'10*, pp. 11–20, 2010.

[24] J. Lygeros. On Reachability and Minimum Cost Optimal Control. *Automatica*, 40(6): 917–927, 2004.

[25] M. Althoff, O. Stursberg, and M. Buss, Reachability Analysis of Nonlinear Systems with Uncertain Parameters Using Conservative Linearization. In *47th IEEE Conference on Decision and Control (CDC)*, pp. 4042–4048, 2008.

[26] MAB. Sassi, R. Testylier, T. Dang, and A. Girard. Reachability Analysis of Polynomial Systems Using Linear Programming Relaxations. In *ATVA'12*, pp. 137–151, 2012.

[27] A. Girard, C. L. Guernic, and O. Maler. Efficient Computation of Reachable Sets of Linear Time-Invariant Systems with Inputs. In *HSCC'06*, pp. 257–271, 2006.

[28] A. B. Kurzhanski and P. Varaiya. Ellipsoidal Techniques for Reachability Analysis. In *HSCC'00*, pp. 202–214, 2000.

[29] T. F. Filippova. Description of Dynamics of Ellipsoidal Estimates of Reachable Sets of Nonlinear Control Systems with Bilinear Uncertainty. In *Numerical Methods and Applications*, pp. 97–105, 2018.

[30] E. Goubault and S. Putot. Robust Under-Approximations and Application to Reachability of Non-Linear Control Systems With Disturbances. *IEEE Control Systems Letters (CSL)*, 4(4): 928-933, 2020.

[31] E. Goubault, S. Putot, and L. Sahlman. Inner and Outer Approximating Flowpipes for Delay Differential Equations. In *CAV'18*, pp. 523–541, 2018.

[32] B. Xue, P. N. Mosaad, M. Fränzle, M. Chen, Y. Li, and N. Zhan. Safe Over-and Under-Approximation of Reachable Sets for Delay Differential Equations. In *FORMATS*, pp. 281–299, 2017.

[33] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances. In *56th IEEE Conference on Decision and Control (CDC)*, pp. 2242-2253, 2017.

[34] B. Xue, M. Fränzle, and N. Zhan. Inner-approximating Reachable Sets for Polynomial Systems with Time-varying Uncertainties. *IEEE Trans. Automat. Contr. (TAC)*, 65(4): 1468–1483, 2020.

[35] Z. Zhou, J. Ding, H. Huang, R. Takei, and C. Tomlin. Efficient Path Planning Algorithms in Reach-avoid Problems. *Automatica*, 89(3): 28–36, 2018.

[36] M. Chen, S. Herbert, and C. J. Tomlin. Fast Reachable Set Approximations via State Decoupling Disturbances. In *55th IEEE Conference on Decision and Control (CDC)*, pp. 191–196, 2016.

[37] M. Althoff, G. Frehse, and A. Girard. Set Propagation Techniques for Reachability Analysis. Annual Review of Control, Robotics, and Autonomous Systems, 4, 369-395, 2021.

[38] J. N. Maidens and S. Kaynama and I. M. Mitchell and M. M.K. Oishi, and G. A. Dumont. Lagrangian Methods for Approximating the Viability Kernel in High-dimensional Systems. *Automatica*, 49(7): 2017–2029, 2013.

[39] S. Kaynama, J. Maidens, M. Oishi, I. M. Mitchell, and G. A. Dumont. Computing the Viability Kernel Using Maximal Reachable Sets. In *HSCC'12*, pp. 55–64, 2012.

[40] B. Xue, A. Easwaran, N. Cho, and M. Fränzle. Reach-Avoid Verification for Nonlinear Systems Based on Boundary Analysis. *IEEE Trans. Automat. Contr. (TAC)*, 62(7): 3518–3523, 2017.

[41] J. Fisac, M. Chen, C. J. Tomlin, and S. Sastry. Reach-Avoid Problems with Time-Varying Dynamics, Targets and Constraints. In *HSCC'15*, pp. 11–20, 2015.

[42] T. C. Wang, S. Lall, and M. West. Polynomial Level-set Method for Polynomial System Reachable Set estimation. *IEEE Trans. Automat. Contr. (TAC)*, 58(10): 2508–2521, 2013.

[43] M. Althoff. An Introduction to CORA 2015. In *ARCH*, pp. 120–151, 2015.

[44] M. Rungger and M. Zamani. Accurate Reachability Analysis of Uncertain Nonlinear Systems. In *HSCC'18*, pp. 61–70, 2018.

[45] R. Hu and Z. She. OURS: Over- and Under-Approximating Reachable Sets for Analytic Time-invariant Differential Equations. *Journal of Systems Architecture*, vol. 128, 102580, pp. 1–12, 2022.