Inner-approximating Reach-avoid Sets for Discrete-time Polynomial Systems

Bai Xue[1], Naijun Zhan[1] and Martin Fränzle[3]

***Abstract*— In this paper we propose a computational method based on semi-definite programming for synthesizing infinite-time reach-avoid sets in discrete-time polynomial systems. An infinite-time reach-avoid set is a set of initial states making the system eventually, i.e., within finite time enter the target set while remaining inside another specified (safe) set during each time step preceding the target hit. The reach-avoid set is first characterized equivalently as a strictly positive sub-level of a bounded value function, which in turn is shown to be a solution to a system of derived equations. The derived equations are further relaxed into a system of inequalities, which is encoded into semi-definite constraints based on the sum-of-squares decomposition for multivariate polynomials, such that the problem of synthesizing inner-approximations of the reach-avoid set can be addressed via solving a semi-definite programming problem. Two examples demonstrate the proposed approach.**

I. Introduction

Discrete-time dynamical systems, where state changes arise in discrete time instants, are important mathematical models used to describe the evolution of complex dynamical systems. For example, the population of a species that reproduces in annual turns is adequately modeled using discrete-time systems such as the Discrete Malthusian Growth model (named after the work of Thomas Malthus, 1766–1834). They also play an important role in understanding continuous-time dynamical systems by analysing their induced discrete-time snapshot sequences. In particular, with numerical simulation being the workhorse of many practical approaches to the analysis of complex dynamical systems, the point sequences calculated by a numerical ordinary differential equations solver form a discrete-time dynamical system that approximates the solution of an initial value problem for an ODE [5]. We here consider discrete-time polynomial systems whose dynamics are represented by polynomials. These systems are an important class of nonlinear systems due to the fact that many nonlinear systems such as Lotka-Volterra systems can be modelled as, transformed into, or approximated by polynomial systems.

Reach-avoid analysis is an established verification tool that provides formal guarantees of safety (via avoiding unsafe states) and liveness (via guaranteed reach of a target set) for dynamical systems, e.g., [14]. Automatic verification approaches usually involve computation of reach-avoid sets. A reach-avoid set or, in the terminology of viability theory, capture basin [3] is a set of initial states from which the system is guaranteed to eventually reach a desired target state set while avoiding a set of unsafe or otherwise undesirable states throughout the path to the target [21]. It is different from invariant sets widely studied in, e.g., [2], [6], [17], [23], [24], which are a set of states just enabling the system to avoid a set of unsafe states. Reach-avoid analysis plays a vital role in safety-critical system design such as air traffic management systems [15] and biomedical systems [13]. There is a body of work regarding the study of reach-avoid analysis on continuous-time deterministic systems [7], [14], [22] and discrete-time stochastic systems [1], [8], [20]. Although there are some approaches to reachability (rather than reach-avoid) analysis for discrete-time deterministic systems [10], [12], [18], [19], [24], studies on reach-avoid analysis are still few. Recently, a moment-based convex programming method (or, a semi-definite programming based method) was proposed to compute outer approximations (i.e., super-sets) of the reach-avoid set for discrete-time polynomial systems in [9].

This paper complements the aforementioned outer approximation by providing a computational inner approximation of the reach-avoid set of discrete-time polynomial systems. Such an inner approximation is a central tool in system synthesis, as it generates a set of states that reliably satisfy the desired reach-avoid property. Our method for synthesizing reach-avoid sets begins with a bounded value function whose certain strictly positive sub-level set is equal to the reach-avoid set. The value function is constructed by trajectories of a switched system generated by forcing the considered polynomial system to stay still when it touches either the target set or the complement of the specified safe set, and is reduced to a solution of a derived system of equations. Via relaxing the system of equations into a system of inequalities, which are further encoded into a set of semi-definite constraints via the sum-of-squares decomposition for multivariate polynomials, a semi-definite program is finally obtained for inner-approximating the reach-avoid set. Finally, two examples demonstrate the proposed approach.

The main contributions of this paper are summarized as follows. 1). The exact reach-avoid set over the infinite time horizon is characterized as a strictly positive sub-level set of a bounded value function, which is finally reduced to a unique solution to a system of equations. 2). An overall non-convex problem of estimating reach-avoid sets is reduced to a

[1]State Key Lab. of Computer Science, Institute of Software, CAS, Beijing, China, and the University of Chinese Academy of Sciences, Beijing, China {xuebai,znj}@ios.ac.cn

[2]Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany fraenzle@uni-oldenburg.de

This work has been supported through grants by NSFC under grant No. 61872341, 61836005, 61625206, the CAS Pioneer Hundred Talents Program under grant No. Y8YC235015, Deutsche Forschungsgemeinschaft through the grants DFG GRK 1765 "System Correctness under Adverse Conditions" and FR 2715/4-1 "Integrated Socio-technical Models for Conflict Resolution and Causal Reasoning".

problem of solving a single convex program. In addition, the inner approximations obtained are particularly simple, since they are represented by a sub-level set of a single polynomial of a predefined degree. Moreover, an inner approximation of the reach-avoid set can be readily obtained by solving a single semi-definite program using freely available software.

This paper is structured as follows. Section II introduces the concepts of discrete-time polynomial systems and the corresponding reach-avoid sets of interest. After elucidating our approach for inner-approximating reach-avoid sets in Section III, we demonstrate it on two examples in Section IV and finally conclude this paper in Section V.

II. Preliminaries

In this section we describe discrete-time polynomial systems and reach-avoid sets of interest in this paper. Before formulating the reach-avoid problem, let us introduce some basic notions used throughout this paper: $\mathbb{N}$ stands for the set of nonnegative integers and $\mathbb{R}$ for the set of real numbers. For a set Δ, Δ^c denotes its complement. $\mathbb{R}[\cdot]$ denotes the ring of polynomials in variables given by the argument. Vectors are denoted by boldface letters.

The discrete-time polynomial system (denoted **DPS** in the sequel) considered in this paper is an iterative polynomial map of the following form,

$$\begin{aligned} \boldsymbol{x}(l+1) &= \boldsymbol{f}(\boldsymbol{x}(l)), \forall l \in \mathbb{N}, \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0 \in \mathbb{R}^n, \end{aligned} \tag{1}$$

where $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_n(\boldsymbol{x}))^\top$ with $f_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$, $i = 1, \dots, n$.

We use $\boldsymbol{\phi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}^n$, induced by $\boldsymbol{x}_0$, to denote the trajectory of the system **DPS**, i.e., $\boldsymbol{\phi}_{\boldsymbol{x}_0}(l) := \boldsymbol{x}(l), \forall l \in \mathbb{N}$.

Now, we define the reach-avoid set such that any trajectory of the system **DPS** starting from this set will reach an open target set $\mathtt{TR}$ in finite time while staying within a compact set X till the target hitting time, where $\mathtt{TR}$ and X are defined by polynomial inequations as

$$\begin{aligned} &\mathtt{TR} = \{\boldsymbol{x} \in \mathbb{R}^n \mid g(\boldsymbol{x}) < 1\} \text{ and} \\ &X = \{\boldsymbol{x} \in \mathbb{R}^n \mid h_0(\boldsymbol{x}) \leq 0\} \end{aligned} \tag{2}$$

with $g(\boldsymbol{x}), h_0(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and $\mathtt{TR} \subseteq X$.

Definition 1 (Reach-Avoid Set): The reach-avoid set $\mathtt{RA}$ is the set of all initial states inducing trajectories of ***DPS*** which enter the target set $\mathtt{TR}$ eventually at some time $l \in \mathbb{N}$ while staying in the set X over the time horizon $[0, l] \cap \mathbb{N}$, i.e.,

$$\mathtt{RA} = \{\boldsymbol{x}_0 \in X \mid \exists l \in \mathbb{N}. \boldsymbol{\phi}_{\boldsymbol{x}_0}(l) \in \mathtt{TR} \wedge \bigwedge_{j=1}^{l} \boldsymbol{\phi}_{\boldsymbol{x}_0}(j) \in X\}.$$

An inner-approximation is a subset of the reach-avoid set $\mathtt{RA}$.

III. Inner-approximating Reach-avoid Sets

In this section, we present our approach for inner-approximating the reach-avoid set $\mathtt{RA}$ in Definition 1 via solving a semi-definite programming problem. To this end, we first define a value function generated by trajectories of a switched polynomial system such that its strict sub-level set w.r.t. level 1 is equal to the reach-avoid set $\mathtt{RA}$. The value function is shown to be the unique solution to a system of equations (if solutions exist at all). On the basis of the system of equations, we finally construct a semi-definite program for inner-approximating the reach-avoid set $\mathtt{RA}$.

The aforementioned value function is defined by trajectories of a switched discrete-time polynomial system, which is built upon the system **DPS** such that the state of the system **DPS** is forced to stay still once it enters either the complement X^c of the set X or the target set $\mathtt{TR}$.

Definition 2: A switched discrete-time polynomial system (or, ***SDPS***), which is formed by the system ***DPS***, is a quintuple $(\boldsymbol{x}_0, \widehat{L}, \widehat{X}, \widehat{\mathcal{X}}, \widehat{\mathcal{F}})$ with the following components:

- $\widehat{L} = \{1, 2, 3\}$ is a set of three locations;
- $\widehat{X}$ is the state constraint set;
- $\widehat{\mathcal{X}} = \{\widehat{X}_i, i = 1, 2, 3\}$;
- $\boldsymbol{x}_0 \in \widehat{X}$ is the initial state;
- $\widehat{\mathcal{F}} = \{\widehat{\boldsymbol{f}}_i(\boldsymbol{x}) : \widehat{X}_i \to \mathbb{R}^n, i = 1, 2, 3\}$ with

$$\widehat{\boldsymbol{f}}_1(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$$

constraining the evolution of the state by the iterative polynomial map $\tilde{\boldsymbol{x}} := \widehat{\boldsymbol{f}}_1(\boldsymbol{x})$ at location $i = 1$, and

$$\widehat{\boldsymbol{f}}_i(\boldsymbol{x}) = \boldsymbol{x}$$

constraining the evolution of the state by the iterative polynomial map $\tilde{\boldsymbol{x}} := \widehat{\boldsymbol{f}}_i(\boldsymbol{x})$ at location $i \in \{2, 3\}$,

where

1) $\widehat{X} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \leq 0\}$ with $h(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and

$$\widehat{X} \supseteq \Omega([0,1], \boldsymbol{f}, X)$$

is a compact set in $\mathbb{R}^n$, where

$$\Omega([0,1], \boldsymbol{f}, X) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = \boldsymbol{f}(\boldsymbol{y}), \boldsymbol{y} \in X\} \cup X$$

is the set of reach states of the system ***DPS*** starting from the set X with the time instants $[0, 1] \cap \mathbb{N}$;

2) $\widehat{X}_1 = X \setminus \mathtt{TR} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h_0(\boldsymbol{x}) \leq 0 \wedge 1 - g(\boldsymbol{x}) \leq 0\}$;
3) $\widehat{X}_2 = \mathtt{TR} = \{\boldsymbol{x} \in \mathbb{R}^n \mid g(\boldsymbol{x}) - 1 < 0\}$;
4) $\widehat{X}_3 = \widehat{X} \setminus X = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \leq 0 \wedge -h_0(\boldsymbol{x}) < 0\}$.

Remark 1: The set $\widehat{X}$ in Definition 2 exists since X is a compact set in $\mathbb{R}^n$ and $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$. It can be computed by solving a semi-definite programming problem, which will be presented later.

Definition 3: Given an initial state $\boldsymbol{x}_0 \in \widehat{X}$, if there exists a sequence $(\boldsymbol{x}(l))_{l \in \mathbb{N}}$ starting from $\boldsymbol{x}_0$ and satisfying the dynamics defined by the iterative piece-wise polynomial map

$$\boldsymbol{x}(l+1) = \widehat{\boldsymbol{f}}(\boldsymbol{x}(l)), , \forall l \in \mathbb{N},$$

where $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and

$$\widehat{\boldsymbol{f}}(\boldsymbol{x}) := 1_{\widehat{X}_1} \cdot \widehat{\boldsymbol{f}}_1(\boldsymbol{x}) + 1_{\widehat{X}_2} \cdot \widehat{\boldsymbol{f}}_2(\boldsymbol{x}) + 1_{\widehat{X}_3} \cdot \widehat{\boldsymbol{f}}_3(\boldsymbol{x}), \tag{3}$$

with

$$\widehat{\boldsymbol{f}}(\cdot) : S \to \mathbb{R}^n, S = \widehat{X}_i \text{ if } \boldsymbol{x} \in \widehat{X}_i,$$

and $1_{\widehat{X}_i}:\widehat{X}_i\rightarrow\{0,1\}$, $i=1,2,3$, representing the indicator function of the set $\widehat{X}_i$, i.e.,

$$1_{\widehat{X}_i}:=\begin{cases}1, & \text{if } \boldsymbol{x}\in\widehat{X}_i,\\ 0, & \text{if } \boldsymbol{x}\notin\widehat{X}_i,\end{cases}$$

then the trajectory $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(\cdot):\mathbb{N}\rightarrow\mathbb{R}^n$, induced by $\boldsymbol{x}_0$, of the system ***SDPS*** is defined as follows:

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l):=\boldsymbol{x}(l),\forall l\in\mathbb{N}.$$

The set $\widehat{X}$ is an invariant set for the system **SDPS**. That is, trajectories of the system **SDPS** starting from $\widehat{X}$ are trapped in it. This is formally formulated in Corollary 1.

Corollary 1: If $\boldsymbol{x}_0\in\widehat{X}$, $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)\in\widehat{X}$ for $l\in\mathbb{N}$.

Proof: Clearly, if $\boldsymbol{x}_0\in\widehat{X}_2\cup\widehat{X}_3$,

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)\in\widehat{X}_2\cup\widehat{X}_3,\forall l\in\mathbb{N}$$

holds.

If $\boldsymbol{x}_0\in\widehat{X}\setminus(\cup_{i=2}^{3}\widehat{X}_i)$, i.e., $\boldsymbol{x}_0\in\widehat{X}_1$, one of the following three cases hold:

1) there exists $l\in\mathbb{N}$ such that $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in\widehat{X}_2$ for $i\in[l,\infty)\cap\mathbb{N}$ and $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in\widehat{X}_1$ for $i\in[0,l)\cap\mathbb{N}$;
2) there exists $l\in\mathbb{N}$ such that $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in\widehat{X}_3$ for $i\in[l,\infty)\cap\mathbb{N}$ and $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in\widehat{X}_1$ for $i\in[0,l)\cap\mathbb{N}$;
3) $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)\in\widehat{X}_1$ for $l\in\mathbb{N}$ holds.

Therefore, the conclusion holds. ■

From the proof of Corollary 1, we observe that trajectories evolving in the viable domain $\widehat{X}$ can be classified into three disjoint groups:

1) trajectories touching the set $\widehat{X}_2=\mathtt{TR}$ in finite time and staying inside X prior to the target hitting time;
2) trajectories touching the set $\widehat{X}_3=\widehat{X}\setminus X$ in finite time;
3) trajectories staying in the set $\widehat{X}_1=X\setminus\mathtt{TR}$ for all time.

It is obvious that the reach-avoid set $\mathtt{RA}$ is equal to the set of initial states driving their trajectories to enter the target set $\mathtt{TR}$ in finite time and stay within the set X preceding the target hitting time. Let $\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}$ be the hitting time of the target set $\mathtt{TR}$ for the trajectory $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(\cdot):\mathbb{N}\rightarrow\mathbb{R}^n$, i.e.,

$$\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}=\min\{l\in\mathbb{N}\mid\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)\in\mathtt{TR}\}.$$

Lemma 1: $\mathtt{RA}=\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}$, where $\mathtt{RA}$ is the reach-avoid set in Definition 1.

Proof: We first prove that $\mathtt{RA}\subseteq\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}$.

Let $\boldsymbol{x}_0\in\mathtt{RA}$. If $\boldsymbol{x}_0\in\mathtt{TR}$, it is clear that $\boldsymbol{x}_0\in\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}$. Thus, we consider $\boldsymbol{x}_0\in\mathtt{RA}\setminus\mathtt{TR}$.

Definition 1 indicates that there exists $l\in\mathbb{N}$ such that

$$\boldsymbol{\phi}_{\boldsymbol{x}_0}(l)\in\mathtt{TR}\bigwedge\wedge_{i=0}^{l}\boldsymbol{\phi}_{\boldsymbol{x}_0}(i)\in X.$$

Let $\tau_{\mathtt{TR}}^{\boldsymbol{x}_0}=\min\{i\in\mathbb{N}\mid\boldsymbol{\phi}_{\boldsymbol{x}_0}(i)\in\mathtt{TR}\bigwedge\wedge_{j=1}^{i}\boldsymbol{\phi}_{\boldsymbol{x}_0}(j)\in X\}$. It is clear that $\tau_{\mathtt{TR}}^{\boldsymbol{x}_0}\geq 1$. Definition 2 implies that $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)=\boldsymbol{\phi}_{\boldsymbol{x}_0}(i)$ for $i\in[0,\tau_{\mathtt{TR}}^{\boldsymbol{x}_0}]\cap\mathbb{N}$. Therefore,

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(\tau_{\mathtt{TR}}^{\boldsymbol{x}_0})\in\mathtt{TR}\bigwedge\wedge_{i=0}^{\tau_{\mathtt{TR}}^{\boldsymbol{x}_0}-1}\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in X\setminus\mathtt{TR}.$$

Consequently, $\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}=\tau_{\mathtt{TR}}^{\boldsymbol{x}_0}\leq l<\infty$ and thus $\boldsymbol{x}_0\in\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}$. We have $\mathtt{RA}\subseteq\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}$.

Next, we show that $\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}\subseteq\mathtt{RA}$. Let $\boldsymbol{x}_0\in\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}$. If $\boldsymbol{x}_0\in\mathtt{TR}$, it is clear that $\boldsymbol{x}_0\in\mathtt{RA}$. Thus, we consider $\boldsymbol{x}_0\in\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}\setminus\mathtt{TR}$.

Then, there exists $l\in\mathbb{N}$ such that $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)\in\mathtt{TR}$, implying that

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)\in\mathtt{TR}\bigwedge\wedge_{i=0}^{l}\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in X$$

holds. Similar to the above procedure in proving $\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}<\infty$, we can show that that $\tau_{\mathtt{TR}}^{\boldsymbol{x}_0}=\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}$ and thus $\tau_{\mathtt{TR}}^{\boldsymbol{x}_0}\leq l$. Therefore, $\boldsymbol{x}_0\in\mathtt{RA}$ and $\{\boldsymbol{x}\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}}<\infty\}\subseteq\mathtt{RA}$.

In summary, $\mathtt{RA}=\{\boldsymbol{x}_0\in\widehat{X}\mid\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}<\infty\}$ holds. ■

Now, we present the value function $V(\boldsymbol{x}):\widehat{X}\rightarrow\mathbb{R}$, whose strict one sub-level set, i.e., $\{\boldsymbol{x}\in\widehat{X}\mid V(\boldsymbol{x})<1\}$, is equal to the reach-avoid set $\mathtt{RA}$.

$$V(\boldsymbol{x}):=\liminf_{l\rightarrow\infty}\frac{\sum_{i=0}^{l-1}g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(i))}{l}. \tag{4}$$

Since $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l)\in\widehat{X}$ for $l\in\mathbb{N}$ and $g(\boldsymbol{x})\in\mathbb{R}[\boldsymbol{x}]$, $V(\boldsymbol{x})$ is thus bounded over $\widehat{X}$.

Lemma 2: $\mathtt{RA}=\{\boldsymbol{x}\in\widehat{X}\mid V(\boldsymbol{x})<1)\}$, where $V(\cdot):\widehat{X}\rightarrow\mathbb{R}$ is the value function in (4).

Proof: Let $\boldsymbol{x}_0\in\mathtt{RA}$. Clearly, $\boldsymbol{x}_0\in\widehat{X}$. According to Lemma 1, we have that $\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}<\infty$. Therefore,

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in X,\forall i\in[0,\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}]\cap\mathbb{N}\text{ and}$$

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\in\mathtt{TR},\forall i\in[\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0},\infty)\cap\mathbb{N}.$$

Also, $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)=\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0})$ for $i\in[\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0},\infty)\cap\mathbb{N}$ and thus

$$g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i))=g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}))<1,\forall i\in[\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0},\infty)\cap\mathbb{N}.$$

Furthermore, we have

$$\lim_{l\rightarrow\infty}\frac{\sum_{i=0}^{l-1}g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i))}{l}=g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(\widehat{\tau}_{\mathtt{TR}}^{\boldsymbol{x}_0}))<1.$$

Consequently, $V(\boldsymbol{x}_0)<1$ and thus

$$\mathtt{RA}\subseteq\{\boldsymbol{x}\in\widehat{X}\mid V(\boldsymbol{x})<1\}.$$

Next, we show that $\{\boldsymbol{x}_0\in\widehat{X}\mid V(\boldsymbol{x})<1\}\subseteq\mathtt{RA}$.

Assume that $\boldsymbol{x}_0\in\{\boldsymbol{x}\in\widehat{X}\mid V(\boldsymbol{x})<1\}$ but $\boldsymbol{x}_0\notin\mathtt{RA}$.

Therefore, $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i)\notin\mathtt{TR}$ for $i\in\mathbb{N}$ and thus $g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(i))\geq 1$ for $i\in\mathbb{N}$. As a consequence, $V(\boldsymbol{x}_0)\geq 1$, contradicting the assumption that $V(\boldsymbol{x}_0)<1$. Thus, $\boldsymbol{x}_0\in\mathtt{RA}$ and further

$$\{\boldsymbol{x}\in\widehat{X}\mid V(\boldsymbol{x})<1\}\subseteq\mathtt{RA}.$$

In summary, $\{\boldsymbol{x}\in\widehat{X}\mid V(\boldsymbol{x})<1\}=\mathtt{RA}$. ■

Lemma 2 implies that the reach-avoid set $\mathtt{RA}$ follows once we obtained the bounded value function $V(\boldsymbol{x})$. In the following we show that the bounded value function $V(\boldsymbol{x})$ is a solution to a derived system of equations.

Theorem 1: If there exist bounded functions $v(\boldsymbol{x}):\widehat{X}\rightarrow\mathbb{R}$ and $w(\boldsymbol{x}):\widehat{X}\rightarrow\mathbb{R}$ such that for $\boldsymbol{x}\in\widehat{X}$,

$$v(\boldsymbol{x})=v(\widehat{\boldsymbol{f}}(\boldsymbol{x})), \tag{5}$$

$$v(\boldsymbol{x})=g(\boldsymbol{x})+w(\widehat{\boldsymbol{f}}(\boldsymbol{x}))-w(\boldsymbol{x}), \tag{6}$$

then

$$v(\boldsymbol{x})=V(\boldsymbol{x}),\forall\boldsymbol{x}\in\widehat{X}$$

and thus $\texttt{RA} = \{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\}$, where $V(\cdot): \widehat{X} \to \mathbb{R}$ is the value function in (4).

Proof: According to Corollary 1, we have that

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l) \in \widehat{X}, \forall l \in \mathbb{N}$$

if $\boldsymbol{x} \in \widehat{X}$.

From (5), we have that

$$v(\boldsymbol{x}) = v(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l)), \forall l \in \mathbb{N}. \tag{7}$$

From (6), we have that

$$v(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l)) = g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l)) + w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l+1)) - w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l)), \forall l \in \mathbb{N},$$

which further indicates that

$$\sum_{i=0}^{l-1} v(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(i)) = \sum_{i=0}^{l-1} g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(i)) + w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l)) - w(\boldsymbol{x}), \forall l \in \mathbb{N}.$$

Combining with (7), we have that

$$v(\boldsymbol{x}) = \frac{\sum_{i=0}^{l-1} g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(i))}{l} + \frac{w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l)) - w(\boldsymbol{x})}{l}, \forall l \in \mathbb{N}.$$

Since $w(\boldsymbol{x})$ is bounded over $\boldsymbol{x} \in \widehat{X}$,

$$v(\boldsymbol{x}) = \lim_{l \to \infty} \frac{\sum_{i=0}^{l-1} g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(i))}{l}$$

holds and thus $v(\boldsymbol{x}) = V(\boldsymbol{x})$.

An immediate consequence is $\texttt{RA} = \{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\}$ from Lemma 2. ∎

It may be challenging to directly solve the system of equations (5) and (6). However, an inner–approximation of the reach-avoid set $\texttt{RA}$ could be obtained by solving a system of inequalities, which is generated by relaxing the system of equations (5) and (6). This is formally stated in Corollary 2.

Corollary 2: If there exist a function $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and a function $w(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ such that for $\boldsymbol{x} \in \widehat{X}$,

$$v(\boldsymbol{x}) \geq v(\widehat{\boldsymbol{f}}(\boldsymbol{x})), \tag{8}$$

$$v(\boldsymbol{x}) \geq g(\boldsymbol{x}) + w(\widehat{\boldsymbol{f}}(\boldsymbol{x})) - w(\boldsymbol{x}), \tag{9}$$

then $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\} \subseteq \texttt{RA}$ is an inner-approximation of the set $\texttt{RA}$.

Proof: According to Corollary 1, $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}}(l) \in \widehat{X}$ for $l \in \mathbb{N}$ if $\boldsymbol{x} \in \widehat{X}$.

Let $\boldsymbol{x}_0 \in \{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\}$. Obviously, we have $\boldsymbol{x}_0 \in X$ due to (9) and the fact that $g(\boldsymbol{x}) \geq 1$ for $\boldsymbol{x} \in \widehat{X} \setminus X$. We will prove that there exists $l \in \mathbb{N}$ such that $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l) \in \texttt{TR}$, i.e., $g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) < 1$.

Assume $g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \geq 1$ for $l \in \mathbb{N}$, i.e.,

$$\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l) \notin \texttt{TR}, \forall l \in \mathbb{N}.$$

From (8), we have that

$$v(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \leq v(\boldsymbol{x}_0) < 1, \forall l \in \mathbb{N}.$$

(9) indicates

$$\begin{aligned} &g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) + w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l+1)) - w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \\ &\leq v(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \leq v(\boldsymbol{x}_0) < 1, \qquad \forall l \in \mathbb{N}. \end{aligned} \tag{10}$$

Thus,

$$w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l+1)) - w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) < 0, \forall l \in \mathbb{N}$$

and consequently

$$w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l+1)) < w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)), \forall l \in \mathbb{N}.$$

Also, since $w(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and thus the sequence $(w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)))_{l \in \mathbb{N}}$ is bounded, implying that the sequence $(w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)))_{l \in \mathbb{N}}$ converges and thus

$$\lim_{l \to \infty} w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l+1)) - w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) = 0. \tag{11}$$

However, (10) implies that

$$\begin{aligned} &w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l+1)) - w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \\ &\leq v(\boldsymbol{x}_0) - g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \leq v(\boldsymbol{x}_0) - 1, \forall l \in \mathbb{N}, \end{aligned} \tag{12}$$

which contradicts (11) since $v(\boldsymbol{x}_0) - 1 < 0$. Thus, there exists $l \in \mathbb{N}$ such that $\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l) \in \texttt{TR}$, i.e., $g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) < 1$. This further indicates $\boldsymbol{x}_0 \in \texttt{RA}$ according Lemma 1 and thus $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\} \subseteq \texttt{RA}$. ∎

(8) and (9) can be equivalently reformulated by the following formulas without indicator functions:

$$\begin{aligned} &\bigwedge_{i=1}^{3} \big[v(\boldsymbol{x}) - v(\widehat{\boldsymbol{f}}_i(\boldsymbol{x})) \geq 0, \forall \boldsymbol{x} \in \widehat{X}_i\big] \wedge \\ &\bigwedge_{i=1}^{3} \big[v(\boldsymbol{x}) - g(\boldsymbol{x}) - w(\widehat{\boldsymbol{f}}_i(\boldsymbol{x})) + w(\boldsymbol{x}) \geq 0, \forall \boldsymbol{x} \in \widehat{X}_i\big]. \end{aligned} \tag{13}$$

When $i = 2, 3$, we observe that $v(\boldsymbol{x}) = v(\widehat{\boldsymbol{f}}_i(\boldsymbol{x}))$ for all functions $v(\cdot): \widehat{X} \to \mathbb{R}$ and $\boldsymbol{x} \in \widehat{X}_i$. Also, since if there exists a function $v(\boldsymbol{x}): \widehat{X} \to \mathbb{R}$ such that $v(\boldsymbol{x}) \geq g(\boldsymbol{x})$ for $\boldsymbol{x} \in \widehat{X}_3$, $v(\boldsymbol{x}) \geq 1$ for $\boldsymbol{x} \in \widehat{X}_3$ holds. Based on these facts, we can construct another system of inequalities, which can also be used to synthesize inner-approximations of the reach-avoid set $\texttt{RA}$ but have more solutions than (13).

Corollary 3: If there exists a function $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and a function $w(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ such that

$$v(\boldsymbol{x}) \geq v(\boldsymbol{f}(\boldsymbol{x})), \forall \boldsymbol{x} \in X \setminus \texttt{TR}, \tag{14}$$

$$v(\boldsymbol{x}) \geq g(\boldsymbol{x}) + w(\boldsymbol{f}(\boldsymbol{x})) - w(\boldsymbol{x}), \forall \boldsymbol{x} \in X \setminus \texttt{TR}, \tag{15}$$

$$v(\boldsymbol{x}) \geq 1, \forall \boldsymbol{x} \in \widehat{X} \setminus X, \tag{16}$$

then $\{\boldsymbol{x} \in \widehat{X} \mid v(x) < 1\} \subseteq \texttt{RA}$ is an inner-approximation of the reach-avoid set $\texttt{RA}$.

Proof: Since $\widehat{X} = \cup_{i=1}^{3} \widehat{X}_i$, $\widehat{\boldsymbol{f}}_1(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$ for $\boldsymbol{x} \in X_1$, $\widehat{\boldsymbol{f}}_2(\boldsymbol{x}) = \boldsymbol{x}$ for $\boldsymbol{x} \in \widehat{X}_2$ and $\widehat{\boldsymbol{f}}_3(\boldsymbol{x}) = \boldsymbol{x}$ for $\boldsymbol{x} \in \widehat{X}_3$, where $\widehat{X}_1 = X \setminus \texttt{TR}$, $\widehat{X}_2 = \texttt{TR}$ and $\widehat{X}_3 = \widehat{X} \setminus X$, we have that if $v(\boldsymbol{x})$ satisfies (14), $v(\boldsymbol{x})$ satisfies (8), i.e.,

$$v(\boldsymbol{x}) \geq v(\widehat{\boldsymbol{f}}(\boldsymbol{x})), \forall \boldsymbol{x} \in \widehat{X}.$$

Let $\boldsymbol{x}_0 \in \{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\}$ but $\boldsymbol{x}_0 \notin \texttt{RA}$. (16) implies that $\boldsymbol{x}_0 \in X$. Also, (8) and (16) indicate that

$$v(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \in X \setminus \texttt{TR}, \forall l \in \mathbb{N},$$

which indicates that $g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \geq 1$ for $l \in \mathbb{N}$. Thus, from (15) we have that

$$v(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) \geq g(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)) + w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l+1)) - w(\widehat{\boldsymbol{\phi}}_{\boldsymbol{x}_0}(l)), \forall l \in \mathbb{N}.$$

Then following the proof of Corollary 2, we can obtain a contradiction and conclude that $\boldsymbol{x}_0 \in \mathtt{RA}$. Thus, $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\} \subseteq \mathtt{RA}$. ■

Comparing constraints (14)∼(16) and constraints (8)∼(9), solutions to constraints (8)∼(9) also satisfy constraints (14)∼(16). Consequently, we inner-approximate the reach-avoid set via solving constraints (14)∼(16) rather than (8)∼(9). For this sake, constraints (14)∼(16) are encoded into semi-definite constraints using the sum-of-squares decomposition for multivariate polynomials, finally leading to a semi-definite program (17), where $\sum[\boldsymbol{x}]$ is used to represent the set of sum-of-squares polynomials over variables $\boldsymbol{x}$, i.e.,

$$\sum[\boldsymbol{x}] = \{p \in \mathbb{R}[\boldsymbol{x}] \mid p = \sum_{i=1}^{k'} q_i^2, q_i \in \mathbb{R}[\boldsymbol{x}], i = 1, \dots, k'\}.$$

$$\begin{aligned} & \min \boldsymbol{c} \cdot \hat{\boldsymbol{w}} \\ & \text{s.t.} \\ & v(\boldsymbol{x}) - v(\boldsymbol{f}(\boldsymbol{x})) + s_0(\boldsymbol{x})h_0(\boldsymbol{x}) + s_1(\boldsymbol{x})(1 - g(\boldsymbol{x})) \in \sum[\boldsymbol{x}], \\ & v(\boldsymbol{x}) - g(\boldsymbol{x}) - w(\boldsymbol{f}(\boldsymbol{x})) + w(\boldsymbol{x}) + s_2(\boldsymbol{x})h_0(\boldsymbol{x}) \\ & \qquad + s_3(\boldsymbol{x})(1 - g(\boldsymbol{x})) \in \sum[\boldsymbol{x}], \\ & v(\boldsymbol{x}) - 1 + s_4(\boldsymbol{x})h(\boldsymbol{x}) - s_5(\boldsymbol{x})h_0(\boldsymbol{x}) \in \sum[\boldsymbol{x}], \end{aligned} \tag{17}$$

where $\boldsymbol{c} \cdot \widehat{\boldsymbol{w}} = \int_{\widehat{X}} v(\boldsymbol{x}) d\boldsymbol{x}$, $\hat{\boldsymbol{w}}$ is the constant vector computed by integrating the monomials in $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ over $\widehat{X}$, $\boldsymbol{c}$ is the vector composed of unknown coffecients in $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$, $w(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and $s_i(\boldsymbol{x}) \in \sum[\boldsymbol{x}]$, $i = 0 \dots, 5$.

Theorem 2: Let $(v(\boldsymbol{x}), w(\boldsymbol{x}))$ be a solution to the semi-definite program (17), then $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\}$ is an inner-approximation of the reach-avoid set $\mathtt{RA}$.

Proof: Since $v(\boldsymbol{x})$ satisfies the constraint in (17), we obtain that $v(\boldsymbol{x})$ satisfies (14)∼(16) according to $\mathcal{S}-$procedure presented in [4]. Consequently, $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) < 1\} \subseteq \mathtt{RA}$ holds from Corollary 3. ■

A function $h(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ such that $\widehat{X} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \leq 0\}$ in Definition 2 could be computed by solving the semi-definite program (18). The polynomial template for the function $h(\boldsymbol{x})$ in this paper is taken as $\sum_{i=1}^{n}(x_i - a_i)^2 - R$, where $(a_1, \dots, a_n)^\top$ is a specified interior point of the set X and R is an unknown parameter. The reason for taking this form is mainly for the fast and easy computation of the integral $\boldsymbol{c} \cdot \widehat{\boldsymbol{w}} = \int_{\widehat{X}} v(\boldsymbol{x}) d\boldsymbol{x}$ in (17).

Lemma 3: Let $h(\boldsymbol{x}))$ be a solution to (18), where $h(\boldsymbol{x}) = h_0(\boldsymbol{x}) - R$, then $\Omega([0,1], \boldsymbol{f}) \subseteq \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \leq 0\}$.

Proof: Since $h(\boldsymbol{x})$ satisfies the constraint in (18), we obtain that $h(\boldsymbol{x})$ satisfies according to $\mathcal{S}-$ procedure in [4]

$$h(\boldsymbol{x}) \leq 0 \text{ and } h(\boldsymbol{f}(\boldsymbol{x})) \leq 0, \forall \boldsymbol{x} \in X.$$

$$\begin{aligned} & \min R \\ & \text{s.t.} \\ & R - h_0(\boldsymbol{x}) + s_0' h_0(\boldsymbol{x}) \in \sum[\boldsymbol{x}], \\ & R - h_0(\boldsymbol{f}(\boldsymbol{x})) + s_1' h_0(\boldsymbol{x}) \in \sum[\boldsymbol{x}], \\ & i = 1, \dots, k, \end{aligned} \tag{18}$$

where $s_i' \in \sum[\boldsymbol{x}]$, $i = 0, 1$.

	SDP (17)			
Ex.	d_v	d_w	d_s	T
1	15	15	20	26.28
2	16	16	16	61.42

TABLE I

PARAMETERS OF OUR IMPLEMENTATIONS ON (17) FOR EXAMPLES 1 AND 2. d_v, d_w: DEGREE OF POLYNOMIALS v, w IN (17), RESPECTIVELY; AND d_s : DEGREE OF POLYNOMIALS s_i IN (17), RESPECTIVELY, $i = 0, \dots, 5$; T: COMPUTATION TIMES (SECONDS).

Therefore, $\Omega([0,1], \boldsymbol{f}, X) \subseteq \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \leq 0\}$. ■

IV. EXAMPLES

In this section we demonstrate our semi-definite programming based approach on two examples. All computations were performed on an i7-7500U 2.70GHz CPU with 32GB RAM running Windows 10. The parameters controlling the performance of our approach are presented in Table I and Table II. YALMIP [11] and Mosek [16] were used to implement the semi-definite program (17) and (18).

Example 1: We consider a computer-based model of the following academic ordinary differential equation:

$$\begin{cases} \dot{x} = -0.5x - 0.5y + 0.5xy \\ \dot{y} = -0.5y + 1 \end{cases}.$$

When performing computer simulations, Euler's method is often used to analyze an ordinary differential equation, which employs the idea of a linear extrapolation along the local derivative. When the simulation step is 0.01, the resulting discrete-time system is of the following form:

$$\begin{cases} x(l+1) = x(l) + 0.01(-0.5x(l) - 0.5y(l) + 0.5x(l)y(l)) \\ y(l+1) = y(l) + 0.01(-0.5y(l) + 1) \end{cases}. \tag{19}$$

	SDP (18)		
Ex.	d_h	$d_{s'}$	T
1	2	2	0.50
2	2	4	0.95

TABLE II

PARAMETERS OF OUR IMPLEMENTATIONS ON (18) FOR EXAMPLES 1 AND 2. d_h: DEGREE OF POLYNOMIALS h IN (18), RESPECTIVELY; AND $d_{s'}$: DEGREE OF POLYNOMIALS s_i' IN (18), RESPECTIVELY, $i = 0, 1$; T: COMPUTATION TIMES (SECONDS).

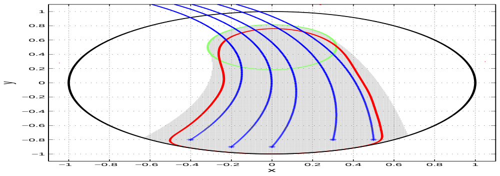

Fig. 1. Estimations of RA for Example 1. (Black curve denotes the boundary of X. Green curve denotes the boundary of the target set TR. Gray region denotes RA obtained by simulation methods. Red curve denotes the boundary of the computed inner-approximation of the reach-avoid set RA. Blue curves denote five trajectories starting from (-0.4,-0.8), (-0.2,-0.9), (-0.0,-0.9), (0.3,-0.8) and (0.5,-0.8), respectively.)

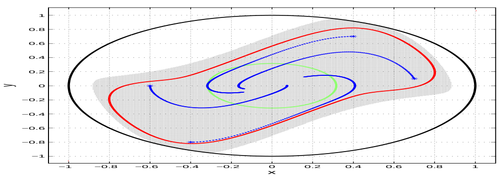

Fig. 2. An illustration of inner-approximating RA for Example 2. (Black curve denotes the boundary of X. Green curve denotes the boundary of the target set TR. Gray region denotes RA obtained by simulation methods. Red curve denotes the boundary of the computed inner-approximation of the reach-avoid set RA. Blue curves denote five trajectories starting from (-0.5,0.0), (-0.4,-0.8), (0.7,0.1) and (0.4,0.7), respectively.)

Assume that $X = \{(x,y) \mid x^2+y^2-1 \leq 0\}$ and $\mathtt{TR} = \{(x,y) \mid 10x^2+10(y-0.5)^2 < 1\}$.

In this example we computed $\widehat{X} = \{(x,y) \mid x^2+y^2-1.1 \leq 0\}$ by solving the semi-definite program (18). The computed inner-approximation of the reach-avoid set RA by solving the semi-definite program (17) is showcased in Fig. 1, which also presents five trajectories of system (19) leaving the set X finally after hitting the target set TR.

Example 2: In this example we consider a computer-based model of the reversed-time Van der Pol oscillator

$$\begin{cases} \dot{x} = -2y \\ \dot{y} = 0.8x + 10(x^2-0.21)y \end{cases}.$$

Discretizing the model with the explicit Euler scheme with a sampling time 0.01, the associated discrete-time system is

$$\begin{cases} x(l+1) = x(l) + 0.01(-2y(l)) \\ y(l+1) = y(l) + 0.01\big(0.8x(l) + 10(x^2(l)-0.21)y(l)\big) \end{cases} \quad (20)$$

Assume that $X = \{(x,y) \mid x^2+y^2-1 \leq 0\}$ and $\mathtt{TR} = \{(x,y) \mid 10x^2+10y^2 < 1\}$.

In this example we computed $\widehat{X} = \{(x,y) \mid x^2+y^2-1.1 \leq 0\}$ by solving the semi-definite program (18). The computed inner-approximation of the reach-avoid set RA by solving the semi-definite program (17) is showcased in Fig. 2, which also presents four trajectories of system (20).

A. Comparisons

Due to the fact that the dual problem of computing inner-approximations is the computation of outer-approximations, in this subsection we make comparisons between the semi-definite programming based method (17) and the method based on the computation of outer-approximations. The reach-avoid set RA is equal to the intersection of the reach-avoid set $X \setminus \widetilde{\mathtt{RA}}$ and the reach set $\widetilde{R}$, where $\widetilde{\mathtt{RA}} =$

$$\{\boldsymbol{x}_0 \in X \mid \exists l \in \mathbb{N}. \boldsymbol{\phi}_{\boldsymbol{x}_0}(l) \in \widehat{X} \setminus X \wedge \bigwedge_{j=1}^{l} \boldsymbol{\phi}_{\boldsymbol{x}_0}(j) \in \widehat{X} \setminus \mathtt{TR}\}.$$

is a set of initial states making the system enter the set $\widehat{X} \setminus X$ in finite time while remaining inside the set $\widehat{X} \setminus \mathtt{TR}$ during each time step preceding the target hitting and the reach set

$$\widetilde{R} = \{\boldsymbol{x}_0 \in X \mid \exists l \in \mathbb{N}. \boldsymbol{\phi}_{\boldsymbol{x}_0}(l) \in \mathtt{TR}\}.$$

is a set of initial states making the system enter the target set TR in finite time. It is clear that the intersection between an inner-approximation of the set $X \setminus \widetilde{\mathtt{RA}}$ and an inner-approximation of the set $\widetilde{R}$ is an inner-approximation of the reach-avoid set RA. Since $\widetilde{R} = \{\boldsymbol{x}_0 \in X \mid \exists l \in \mathbb{N}. g(\boldsymbol{f}^l(\boldsymbol{x}_0)) < 1\}$, where

$$\boldsymbol{f}^l(\boldsymbol{x}_0) = \underbrace{\boldsymbol{f}\Big(\boldsymbol{f}\Big(\cdots \boldsymbol{f}(\boldsymbol{f}(\boldsymbol{x}_0))\cdots\Big)\Big)}_{l},$$

the set $\widetilde{R}_N = \cup_{l=0}^{N} R_l$ with $R_l = \{\boldsymbol{x}_0 \in X \mid g(\boldsymbol{f}^l(\boldsymbol{x}_0)) < 1\}$ is an inner-approximation of the reach set $\widetilde{R}$, where $N \in \mathbb{N}$ is an arbitrary but fixed non-negative integer, we just need to compute the reach-avoid set $\widetilde{\mathtt{RA}}$, which is a challenging issue generally. Fortunately, we can compute an outer-approximation $\widetilde{\mathtt{ORA}}$ of the reach-avoid set $\widetilde{\mathtt{RA}}$ thanks to the semi-definite programming based method (or the moment-based optimization method) in [9], consequently leading to the fact that $\widetilde{R}_N \cap X \setminus \widetilde{\mathtt{ORA}}$ is an inner-approximation of the reach-avoid set RA.

The semi-definite program (7) in [9] is employed for computing outer-approximations $\widetilde{\mathtt{ORA}}$, which are respectively represented by polynomials of degree 15 and 16 for Examples 1 and 2. The resulting inner-approximations $X \setminus \widetilde{\mathtt{ORA}}$ for Examples 1 and 2 are respectively illustrated in Fig. 3 and Fig. 4. Also, the set $\widetilde{R}_{49}$ is illustrated in Fig. 3 for Example 1 and the set $\widetilde{R}_{17}$ is illustrated in Fig. 4 for Example 2.

From Fig. 3 and 4, we observe that both the inner-approximations $\widetilde{R}_{49} \cap X \setminus \widetilde{\mathtt{ORA}}$ for Example 1 and $\widetilde{R}_{17} \cap X \setminus \widetilde{\mathtt{ORA}}$ for Example 2 computed by the method based on the computation of outer-approximations are more conservative than the ones from the semi-definite programming based method (17). Moreover, the representation $\widetilde{R}_N \cap X \setminus \widetilde{\mathtt{ORA}}$ of inner-approximations computed by the method based on the computation of outer-approximations is computationally much more complex than the one obtained by the semi-definite program (17) for some systems. For instance, the function $g(\boldsymbol{f}^l(\boldsymbol{x}))$ in $\widetilde{R}_l$ for Example 1 is a polynomial of degree higher than 100 when $l \geq 49$ and the function $g(\boldsymbol{f}^l(\boldsymbol{x}))$ in $\widetilde{R}_l$ for Example 2 is a polynomial of degree higher than 100 when $l \geq 17$. The polynomials of degree higher than 100 are challenging to manipulate generally in practice. This is why we just show $\widetilde{R}_{49}$ and $\widetilde{R}_{17}$ in Fig. 3 and 4, respectively. In contrast, the inner-approximations

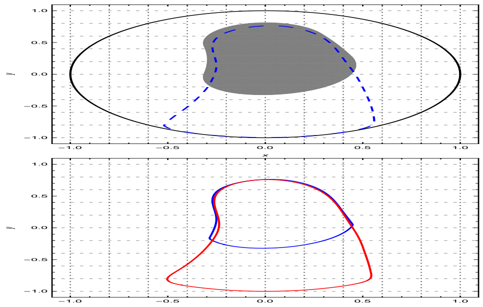

Fig. 3. Estimations of $\mathtt{RA}$ for Example 1 using the semi-definite programming based method (17) and the method based on the computation of outer-approximations. (Above: Black curve denotes the boundary of X. Blue dashed curve denotes the boundary of the inner-approximation $X\setminus\widetilde{\mathtt{ORA}}$ computed by the method in [9]. Gray region denotes the set $\widetilde{R}_{49}$. Below: Red curve denotes the boundary of the inner-approximation computed by the semi-definite programming based method. Blue curve denotes the boundary of the inner-approximation $\widetilde{R}_{49}\cap X\setminus\widetilde{\mathtt{ORA}}$ computed by the method based on the computation of outer-approximations.)

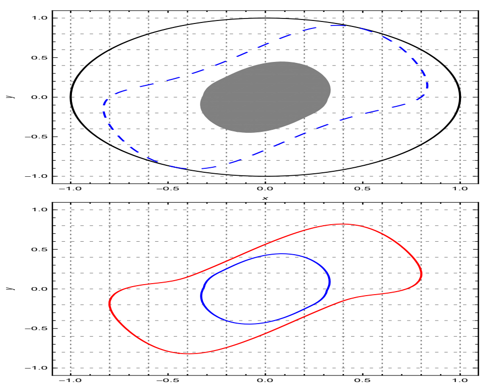

Fig. 4. Estimations of $\mathtt{RA}$ for Example 2 using the semi-definite programming based method (17) and the method based on the computation of outer-approximations. (Above: Black curve denotes the boundary of X. Blue dashed curve denotes the boundary of the inner-approximation $X\setminus\widetilde{\mathtt{ORA}}$ computed by the method in [9]. Gray region denotes the set $\widetilde{R}_{17}$. Below: Red curve denotes the boundary of the inner-approximation computed by the semi-definite programming based method. Blue curve denotes the boundary of the inner-approximation $\widetilde{R}_{17}\cap X\setminus\widetilde{\mathtt{ORA}}$ computed by the method based on the computation of outer-approximations.)

obtained by the semi-definite program (17) are respectively represented by a sub-level set of a single polynomial of degree 15 and 16 for Examples 1 and 2.

V. Conclusion

In this paper we considered the unbounded-time reach-avoid problem for discrete-time polynomial dynamical systems. The reach-avoid set of interest is a set of states taking the system to an eventual hit of the target set while remaining inside a specified (safe) set till the target hit. The reach-avoid set is approximated from the inner by solving a semi-definite programming problem, which was constructed by a system of equations whose solution (if existing) characterizes the exact reach-avoid set. Two examples demonstrated the performance of the proposed approach.

In the future work we would extend the method proposed in this paper to reach-avoid analysis for continuous-time systems. Moreover, we would like to investigate the convergence of the proposed method.

References

[1] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

[2] A. Adjé, P.-L. Garoche, and V. Magron. A sums-of-squares extension of policy iterations. *Nonlinear Analysis: Hybrid Systems*, 25:60–78, 2017.

[3] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre. *Viability theory: new directions*. Springer Science & Business Media, 2011.

[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[5] J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[6] D. Cattaruzza, A. Abate, P. Schrammel, and D. Kroening. Unbounded-time safety verification of guarded lti models with inputs by abstract acceleration. *Journal of Automated Reasoning*, 2020.

[7] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry. Reach-avoid problems with time-varying dynamics, targets and constraints. In *HSCC'15*, pages 11–20, 2015.

[8] J. D. Gleason, A. P. Vinod, and M. M. Oishi. Underapproximation of reach-avoid sets for discrete-time stochastic systems via lagrangian methods. In *CDC'17*, pages 4283–4290. IEEE, 2017.

[9] W. Han and R. Tedrake. Controller synthesis for discrete-time polynomial systems via occupation measures. In *IROS'18*, pages 6911–6918. IEEE, 2018.

[10] M. Korda, D. Henrion, and C. N. Jones. Convex computation of the maximum controlled invariant set for polynomial control systems. *SIAM Journal on Control and Optimization*, 52(5):2944–2969, 2014.

[11] J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *CACSD'04*, pages 284–289. IEEE, 2004.

[12] V. Magron, P.-L. Garoche, D. Henrion, and X. Thirioux. Semidefinite approximations of reachable sets for discrete-time polynomial systems. *SIAM Journal on Control and Optimization*, 57(4):2799–2820, 2019.

[13] J. N. Maidens, S. Kaynama, I. M. Mitchell, M. M. Oishi, and G. A. Dumont. Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica*, 49(7):2017–2029, 2013.

[14] K. Margellos and J. Lygeros. Hamilton–jacobi formulation for reach–avoid differential games. *IEEE Transactions on Automatic Control*, 56(8):1849–1861, 2011.

[15] K. Margellos and J. Lygeros. Toward 4-d trajectory management in air traffic control: a study based on monte carlo simulation and reachability analysis. *IEEE Transactions on Control Systems Technology*, 21(5):1820–1833, 2012.

[16] A. Mosek. The mosek optimization toolbox for matlab manual, 2015.

[17] A. Oustry, M. Tacchi, and D. Henrion. Inner approximations of the maximal positively invariant set for polynomial dynamical systems. *IEEE Control Systems Letters*, 3(3):733–738, 2019.

[18] S. V. Rakovic, E. C. Kerrigan, D. Q. Mayne, and J. Lygeros. Reachability analysis of discrete-time systems with disturbances. *IEEE Transactions on Automatic Control*, 51(4):546–561, 2006.

[19] M. A. B. Sassi, R. Testylier, T. Dang, and A. Girard. Reachability analysis of polynomial systems using linear programming relaxations. In *ATVA*, pages 137–151. Springer, 2012.

[20] S. Summers and J. Lygeros. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica*, 46(12):1951–1961, 2010.

[21] 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.

[22] B. Xue, M. Fränzle, and N. Zhan. Inner-approximating reachable sets for polynomial systems with time-varying uncertainties. *IEEE Transactions on Automatic Control*, 65(4):1468–1483, 2020.

[23] B. Xue, Q. Wang, N. Zhan, and M. Fränzle. Robust invariant sets generation for state-constrained perturbed polynomial systems. In *HSCC'19*, pages 128–137, 2019.

[24] B. Xue and N. Zhan. Robust invariant sets computation for switched discrete-time polynomial systems. *arXiv preprint arXiv:1811.11454*, 2018.