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

Changyuan Zhao[1,2], Shuyuan Zhang[3], Lei Wang[3] and Bai Xue[1,2], IEEE Member

***Abstract*—Reach-avoid analysis, which involves the computation of reach-avoid sets, is an established tool that provides hard guarantees of safety (via avoiding unsafe states) and target reachability (via reaching target sets), and therefore is widely used in safe-critical systems design such as air traffic management systems and biomedical systems. This paper investigates the problem of inner-approximating robust reach-avoid sets for discrete-time polynomial dynamical systems subject to disturbances over open (i.e., not bounded a priori) time horizons. The robust reach-avoid set of interest is a set of all initial states such that the system starting from each of them should reach a target set within a bounded time horizon while staying inside a safe set before the first target hitting time, despite the actual disturbance. Based on a discounted value function and the dynamic programming principle, we show that the robust reach-avoid set can be characterized exactly via the unique bounded solution to a Bellman-type equation. Due to the insurmountable challenge in solving this equation analytically, we reformulate the Bellman-type equation such that its straightforward relaxation can result in a set of novel constraints for inner-approximating the robust reach-avoid set. When the datum involved are polynomials, i.e., the safe set, target set and disturbance set are semi-algebraic, the problem of solving this set of constraints can be encoded into a semi-definite program, which can be solved efficiently in polynomial time via interior point methods. Finally, several examples demonstrate the performance of the proposed method.**

***Index Terms*—Perturbed Discrete-time Systems; Robust Reach-avoid Sets; Inner-approximations.**

I. Introduction

The complexity of today's technological applications induces a quest for pursuing automation, a growing number of autonomous systems are coming into our daily life [17]. Many of these systems such as medical devices, aircraft flight control, and nuclear systems are safety-critical [14]. Their failure will cause catastrophic consequences like loss of life and physical destruction. Being safety-critical, their dynamic behaviors over time have to robustly sustain safety despite *disturbances*, which are ubiquitous and could degrade system performance or result in system failure [23].

The process of designing and verifying with mathematical rigor that a dynamical system behaves correctly is a well-established branch of formal methods in computer science [8]. Because of being capable of providing formal guarantees on safety (via avoiding unsafe states) and progress (via the guaranteed reach of a target set) for dynamical systems, reach-avoid analysis in formal methods has attracted increasing attention in recent years. It becomes an important tool in safety-critical systems design such as air traffic management systems [20]) and biomedical systems [19]. It generally involves computations of reach-avoid sets, a set of initial states from which the system is guaranteed to reach a desired target state set while avoiding a set of undesirable states throughout the path to the target.

In existing literature there are two types of dynamical systems, i.e., continuous-time ones (modeled by differential equations) and discrete-time ones (modeled by difference equations). The robust reach-avoid analysis against disturbances for discrete-time polynomial dynamical systems (modeled by polynomial difference equations) [5], which describe the evolution of a vast class of systems such as biological systems, robots and digital controllers arising from the real world [4], [9], [15], [16], [22], is the focus of this paper.

This paper proposes convex optimization methods for inner-approximating robust reach-avoid sets for discrete-time polynomial systems subject to disturbances over open time horizons. Although there are some approaches to reach-avoid analysis for discrete-time systems free of disturbances [29], there is no work on investigating reach-avoid analysis for discrete-time dynamical systems subject to disturbances over open time horizons, to the best of our knowledge. In the present work, the robust reach-avoid set of interest is a set of all initial states such that for each of them, there exists a finite-time upper bound such that every possible trajectory starting from it will hit the target set within this finite-time upper bound while avoiding a set of unsafe states before the first target hitting time. Our method begins with a discounted value function, whose strict zero upper-level set is equal to the robust reach-avoid set. The discounted value function is built upon a switched system, which is induced by freezing the dynamics of the original system outside the safe set. It is then reduced to a unique bounded solution to a Bellman-type equation, which is challenging to solve analytically. Via reformulating the Bellman-type equation in another form, we construct a set of constraints for inner-approximating the robust reach-void set. When the datum involved are polynomials, i.e., the safe set, target set and disturbance set are semi-algebraic, the problem of solving this set of constraints can be encoded into a semi-definite program. Finally, several examples are used to demonstrate the performance of the proposed method and make comparisons with existing related methods.

Co-first Authors: Changyuan Zhao and Shuyuan Zhang.

This work has been supported through grants by NSFC under grant No. 61836005, 61872341, 61873017, 62192730-62192734, CAS Pioneer Hundred Talents Program, and CAS Project for Young Scientists in Basic Research(No.YSBR-040).

1. State Key Lab. of Computer Science, Institute of Software, CAS, Beijing, China Email: {zhaocy,xuebai}@ios.ac.cn

2. University of Chinese Academy of Sciences, Beijing, China

3. Beihang University, Beijing, China Email: {zhshuyuan,lwang}@buaa.edu.cn

The contributions of this paper are summarized as follows.

1) The problem of inner-approximating robust reach-avoid sets for discrete-time polynomial systems subject to disturbances is investigated. Our approach originates from a discounted value function that can precisely describe the robust reach-avoid set of interest. Based on the discounted value function, a Bellman-type equation is then derived, which admits a unique bounded solution. Although we did not use this equation in our computations directly, this equation plays a fundamental role in our method, since it is the origin of the constructed new set of quantified constraints for inner-approximating robust reach-avoid sets. It is worth noting that although we consider discrete-time polynomial systems in the present work, the derived equation for characterizing the robust reach-avoid set applies to general discrete-time nonlinear systems.
2) We reformulate the Bellman-type equation aforementioned into another equivalent form and construct a set of new quantified constraints for inner-approximating the robust reach-avoid set.
3) When the datum involved are polynomials, i.e., the safe set, target set and disturbance set are semi-algebraic, the problem of solving the set of quantified constraints is addressed within the semi-definite programming framework, thus transforming the non-convex problem of computing robust reach-avoid sets into a convex one which can be solved efficiently in polynomial time via interior point methods [3] and for which there are many powerful off-the-shelf tools [21], [24], [26], [30].

Related Work

Reach-avoid analysis for discrete-time dynamical systems is an important tool in designing safety-critical systems and has attracted increasing attention.

There are some methods for computing outer and inner approximations of reach-avoid sets over open time horizons for discrete-time polynomial systems free of disturbances. For instance, a moment-based convex programming method was proposed to compute outer approximations of reach-avoid sets in [12]. A set of constraints, which is derived from a system of equations, was proposed in our previous work [29] to computing inner-approximations. Although these two works investigate the reach-avoid analysis for discrete-time systems over open time horizons, they consider polynomial systems free of disturbances. In this work, we extend the work [29] and study the reach-avoid analysis for discrete-time systems subject to disturbances. Different from the work [29], a new value function is defined. This new value function facilitates the derivation of a Bellman-type equation featuring a unique bounded solution, which can describe the exact robust reach-avoid set of interest in this paper, and a new set of constraints for inner-approximating the robust reach-avoid set. It is worth noting that a Bellman-type equation was also proposed in the work [28]. However, this equation and the equation in the present work have different objectives and thus have different meanings. The Bellman-type equation in the present work is constructed for characterizing robust reach-avoid sets, while the equation in [28] is for robust invariant sets. The differences between robust invariant sets and robust reach-avoid sets are twofold. One is that every possible trajectory starting inside a robust invariant set is just required to stay inside a safe set for all time without the requirement of reaching a target set. The other is that trajectories starting within the robust reach-avoid set are allowed to leave the safe set after hitting the target set. Moreover, the derivation of these two equations are different. The equation in [28] was constructed based on the original system, while the equation in the present work is obtained using an auxiliary system, whose dynamics are frozen outside the safe set and coincide with the ones of the original system inside the safe set. Besides, the direct relaxation of the Bellman-type equation in [28] can obtain a set of constraints for inner-approximating robust invariant sets, as commented in Section IV in [28]. However, a set of constraints for inner-approximating robust reach-avoid sets cannot be obtained by directly relaxing the Bellman-type equation in the present work. It is obtained via reformulating the equation into another equivalent form.

On the other hand, there are many methods developed for stochastic discrete-time systems. For these systems, the reach-avoid analysis is conducted in the probabilistic context, which assures the probabilistic success of the reach-avoid objective with at least a desired probability p, i.e., admits initial states from which the probability of (eventually or within a given duration) reaching the target while avoiding the unsafe set exceeds p. Establishing methods for computationally solving this problem rely on dynamic programming [1], [25], approximate dynamic programming [13], semi-definite programs [6] and Lagrangian techniques [10]. These works are generally confined to reach-avoid problems of stochastic discrete-time systems over bounded time horizons. Recently, a set of constraints, which is derived from a system of equations, was proposed in [27] for inner-approximating reach-avoid sets of stochastic discrete-time systems over open time horizons. Although a set of initial states ensuring satisfaction of reach-avoid specifications with probability one can be computed by the method in [27], this set considers all disturbances modulo sets of measure zero. In contrast, the present work considers the (inner-)approximating problem of reach-avoid sets in the robust context rather than the probabilistic one, thus considering all actual disturbances and further assuring the satisfiability of reach-avoid properties robustly.

The structure of this paper is presented below. In Section II we introduce discrete-time polynomial dynamical systems subject to disturbances and robust reach-avoid sets generation problems of interest. Then, Section III presents the Bellman-type equation for estimating the exact robust reach-avoid set and our semi-definite programming method for computing its inner-approximations. After demonstrating and discussing the proposed methods on several examples in Section IV, we conclude this paper in Section V.

The following basic notions are used throughout this paper: $\mathbb{N}$ stands for the set of non-negative integers, $\mathbb{N}_{\leq k}$ and $\mathbb{N}_{\geq k}$ with $k \in \mathbb{N}$ denote the set of integers less than or equal to k and larger than or equal to k respectively, and $\mathbb{R}$ for the set of

real numbers; $\mathbb{R}[\cdot]$ denotes the ring of polynomials in variables given by the argument; vectors are denoted by boldface letters; for a set Δ, $\partial\Delta$ denotes its boundary; $\sum[\boldsymbol{x}]$ denotes the set of sum of squares polynomials, i.e.,

$$\sum[\boldsymbol{x}] = \left\{ p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}] \,\middle|\, \begin{array}{l} p(\boldsymbol{x}) = \sum_{i=1}^{k} q_i^2(\boldsymbol{x}), \\ q_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}], i = 1, \ldots, k \end{array} \right\}.$$

II. Preliminaries

In this section, discrete-time polynomial dynamical systems subject to disturbances and robust reach-avoid sets generation problems of interest are introduced.

A discrete-time polynomial dynamical system subject to disturbances (abbr., DPDSD) is a system being modeled by difference equations of the following form:

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

where $\boldsymbol{x}(\cdot) : \mathbb{N} \to \mathbb{R}^n$ are state vectors, $\boldsymbol{d}(\cdot) : \mathbb{N} \to D$ are disturbances with the disturbance set

$$D = \{\boldsymbol{d} \in \mathbb{R}^m \mid q(\boldsymbol{d}) \leq 0\}$$

being bounded in $\mathbb{R}^m$ and $q(\boldsymbol{d}) \in \mathbb{R}[\boldsymbol{d}]$, and $\boldsymbol{f}(\cdot,\cdot) : \mathbb{R}^n \times D \to \mathbb{R}^n$ with $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d}) \in \mathbb{R}[\boldsymbol{x}, \boldsymbol{d}]$.

In the following we define disturbance trajectories and their resulting trajectories for DPDSD.

Definition 1. *A disturbance trajectory π for* DPDSD *is a sequence $(\boldsymbol{d}(l))_{l\in\mathbb{N}}$, where $\boldsymbol{d}(\cdot) : \mathbb{N} \to D$. Furthermore, we define Π as the set of all disturbance trajectories.*

Definition 2. *Given an initial state $\boldsymbol{x}_0 \in \mathbb{R}^n$ and a disturbance trajectory $\pi = (\boldsymbol{d}(l))_{l\in\mathbb{N}}$, the trajectory of* DPDSD*, induced by $\boldsymbol{x}_0$ and π, is a sequence $(\boldsymbol{\phi}^{\pi}_{\boldsymbol{x}_0}(l))_{l\in\mathbb{N}}$ satisfying*

$$\boldsymbol{\phi}^{\pi}_{\boldsymbol{x}_0}(l+1) = \boldsymbol{f}(\boldsymbol{\phi}^{\pi}_{\boldsymbol{x}_0}(l), \boldsymbol{d}(l))$$

for $l \in \mathbb{N}$.

Now, we define the robust reach-avoid set with respect to the target set

$$\texttt{TR} = \{\boldsymbol{x} \in \mathbb{R}^n \mid g(\boldsymbol{x}) < 0\}$$

and the bounded safe set

$$X = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) < 0\},$$

where $\texttt{TR} \subseteq X$, $g(\cdot) : \mathbb{R}^n \to \mathbb{R}$ with $g(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and $h(\cdot) : \mathbb{R}^n \to \mathbb{R}$ with $h(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$. Here, although we restrict the sets $\texttt{TR}$ and X to be bounded open sets, our method in this paper also applies to compact sets $\texttt{TR}$ and X. Besides, our method also applies to bounded sets $\texttt{TR}$ and X that are in the form of upper-level sets of intersections or unions of polynomial inequalities. A simple technique is to apply the $\max \setminus \min$ operator. For instance, when $\texttt{TR} = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_1(\boldsymbol{x}) < 0, \ldots, g_l(\boldsymbol{x}) < 0\}$, we can express it equivalently as $\{\boldsymbol{x} \in \mathbb{R}^n \mid \max_{i\in\{1,\ldots,l\}} g_i(\boldsymbol{x}) < 0\}$. Similarly, when $\texttt{TR} = \cup_{i=1}^{l} \texttt{TR}_i$ with $\texttt{TR}_i = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) < 0\}$, we can express it equivalently as $\{\boldsymbol{x} \in \mathbb{R}^n \mid \min_{i\in\{1,\ldots,l\}} g_i(\boldsymbol{x}) < 0\}$.

Definition 3. *The robust reach-avoid set* RA *is the set of all initial states in the safe set X satisfying that for every initial state in it, there exists a finite upper bound $N \in \mathbb{N}$ such that every possible trajectory of* DPDSD *starting from the initial state will enter the target set* TR *within the finite time horizon $[0, N] \cap \mathbb{N}$ while staying inside the safe set X before the first target hitting time, i.e.,*

$$\texttt{RA} = \left\{ \boldsymbol{x}_0 \in X \,\middle|\, \begin{array}{l} \exists N \in \mathbb{N}. \forall \pi \in \Pi. \exists k \in \mathbb{N}_{\leq N}. \\ [\boldsymbol{\phi}^{\pi}_{\boldsymbol{x}_0}(k) \in \texttt{TR} \bigwedge \wedge_{l=0}^{k} \boldsymbol{\phi}^{\pi}_{\boldsymbol{x}_0}(l) \in X] \end{array} \right\}.$$

An inner-approximation is a subset of the set RA*.*

Remark 1. *If* DPDSD *is free of disturbances, i.e.,*

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

Then,

$$\texttt{RA} = \left\{ \boldsymbol{x}_0 \in X \,\middle|\, \exists k \in \mathbb{N}. \boldsymbol{\phi}_{\boldsymbol{x}_0}(k) \in \texttt{TR} \bigwedge \wedge_{l=0}^{k} \boldsymbol{\phi}_{\boldsymbol{x}_0}(l) \in X \right\}.$$

Its inner-approximations were computed in [29].

Remark 2. *The robust reach-avoid set* RA *is different from the following one*

$$\texttt{RA}' = \left\{ \boldsymbol{x}_0 \in X \,\middle|\, \begin{array}{l} \forall \pi \in \Pi. \exists k \in \mathbb{N}. \\ [\boldsymbol{\phi}^{\pi}_{\boldsymbol{x}_0}(k) \in \texttt{TR} \bigwedge \wedge_{l=0}^{k} \boldsymbol{\phi}^{\pi}_{\boldsymbol{x}_0}(l) \in X] \end{array} \right\}.$$

Obviously, RA *is a subset of* RA′*, i.e.,* $\texttt{RA} \subseteq \texttt{RA}'$*. The main difference between* RA *and* RA′ *lies in that an initial state in* RA *is the one, from which all trajectories starting will enter the target set within a bounded time horizon, while such a uniform bound may not exist for some initial states in* RA′*. The condition such that* $\texttt{RA} = \texttt{RA}'$ *will be investigated in the future work.*

In this paper, we propose methods for inner-approximating the robust reach-avoid set RA under Assumption 1.

Assumption 1. *The robust reach-avoid set* RA *has nonempty interiors.*

III. Robust Reach-avoid Sets Characterization

In this section, we elucidate our approach for computing guaranteed inner-approximations of the set RA. We first derive a Bellman-type equation such that the strict zero super-level set of its unique bounded solution equals the robust reach-avoid set RA. Based on this equation, we then construct a set of quantified inequality constraints such that the certain level sets of its solution are inner-approximations of the set RA. Finally, the set of constraints is encoded into semi-definite constraints via the sum-of-squares decomposition technique for multivariate polynomials that can be solved efficiently in polynomial time via interior-point methods.

A. Bellman-type Equations

In this subsection, we derive a Bellman-type equation such that the strict zero super-level set of its unique bounded solution is equal to the robust reach-avoid set RA.

The Bellman-type equation is derived based on a switched discrete-time system subject to disturbances. This switched system is induced via forcing DPDSD to stay still once it goes outside the safe set X. It differs from the one in [29] without taking into account the presence of disturbances, which is constructed via also requiring DPDSD to stay still once it enters the target set TR.

Definition 4. *A switched discrete-time system subject to disturbances (or,* SDS*), which is constructed based on* DPDSD*, is a system being modeled by iterative nonlinear maps of the following form:*

$$\begin{aligned} &\boldsymbol{x}(l+1) = \widehat{\boldsymbol{f}}(\boldsymbol{x}(l), \boldsymbol{d}(l)), \forall l \in \mathbb{N}, \\ &\boldsymbol{x}(0) = \boldsymbol{x}_0, \end{aligned} \tag{2}$$

where

$$\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}) = 1_{\widehat{X}_1}(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d}) + 1_{\widehat{X}_2}(\boldsymbol{x}) \cdot \boldsymbol{x}$$

with $1_{\widehat{X}_i}(\cdot) : \widehat{X}_i \to \{0, 1\}, i = 1, 2$*, representing the indicator function of the set* $\widehat{X}_i$*, i.e.,*

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

and

1) $\widehat{X}_1 = X = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) < 0\}$*;*
2) $\widehat{X}_2 = \widehat{X} \setminus X = \{\boldsymbol{x} \in \mathbb{R}^n \mid h_0(\boldsymbol{x}) \leq 0 \wedge -h(\boldsymbol{x}) \leq 0\}$*;*
3) $\widehat{X} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h_0(\boldsymbol{x}) \leq 0\}$ *with* $h_0(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ *and* $\widehat{X} \supseteq \Omega([0, 1], \boldsymbol{f}, X)$ *is a compact set in* $\mathbb{R}^n$*, where*

$$\Omega(\mathbb{N}_{\leq 1}, \boldsymbol{f}, X) = \{\boldsymbol{x} \mid \boldsymbol{x} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{d}), \boldsymbol{y} \in X, \boldsymbol{d} \in D\} \cup X$$

is the union of the set X *and all reach states of* DPDSD *starting from* X *in one step.*

Similarly, we define trajectories of SDS, which are formally presented in Definition 5.

Definition 5. *Given an initial state* $\boldsymbol{x}_0 \in \widehat{X}$ *and a disturbance trajectory* $\pi = (\boldsymbol{d}(l))_{l \in \mathbb{N}}$*, the trajectory of* SDS*, induced by* $\boldsymbol{x}_0$ *and* π*, is a sequence* $(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l))_{l \in \mathbb{N}}$ *satisfying*

$$\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l+1) = \widehat{\boldsymbol{f}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l), \boldsymbol{d}(l)).$$

Remark 3. *The existence of the set* $\widehat{X}$ *in Definition 4 can be assured by the boundedness of sets* X *and* D *as well as the fact that* $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d}) \in \mathbb{R}[\boldsymbol{x}, \boldsymbol{d}]$*. As in [29], the set* $\widehat{X}$ *in* Definition *5 can be computed by solving a semi-definite programming problem like* (3)*:*

$$\begin{aligned} &\min R \\ &\text{s.t.} \\ &R - h(\boldsymbol{x}) + s_0' h(\boldsymbol{x}) \in \sum[\boldsymbol{x}], \\ &R - h(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d})) + s_1'(\boldsymbol{x}, \boldsymbol{d}) h(\boldsymbol{x}) + s_2'(\boldsymbol{x}, \boldsymbol{d}) q(\boldsymbol{d}) \in \sum[\boldsymbol{x}, \boldsymbol{d}], \end{aligned} \tag{3}$$

where $s_0' \in \sum[\boldsymbol{x}]$ and $s_i' \in \sum[\boldsymbol{x}, \boldsymbol{d}]$ $i = 1, 2$.

If $R \in \mathbb{R}$ *satisfies the constraints in the semi-definite program* (3)*, the set* $\widehat{X} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h_0(\boldsymbol{x}) \leq 0\}$ *with* $h_0(\boldsymbol{x}) = h(\boldsymbol{x}) - R$ *satisfies the requirement.*

The robust reach-avoid set RA for DPDSD can be equivalently characterized via the set of initial states deriving SDS to reach the target set TR. This relationship is detailed in Proposition 1.

Proposition 1. *The robust reach-avoid set* RA *is equal to*

$$\{\boldsymbol{x}_0 \in \widehat{X} \mid \exists N \in \mathbb{N}. \forall \pi \in \Pi. \exists k \in \mathbb{N}_{\leq N}. \widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k) \in \mathtt{TR}\}.$$

Now we introduce the discounted value function $V(\cdot) : \widehat{X} \to \mathbb{R}$ defined in the following form:

$$V(\boldsymbol{x}) := \inf_{\pi \in \Pi} \sup_{l \in \mathbb{N}} \alpha^l 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}}(l)), \tag{4}$$

where $\alpha \in (0, 1)$ is the discounted factor. It is easy to conclude that $V(\boldsymbol{x}) \leq 1$ for $\boldsymbol{x} \in \widehat{X}$. Its relationship with the robust reach-avoid set RA is presented in Lemma 1.

Lemma 1. *The robust reach-avoid set* RA *is equal to the strict zero super level set of the discounted value function* $V(\boldsymbol{x})$ *in* (4)*, i.e.,*

$$\mathtt{RA} = \{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\}.$$

The discounted value function $V(\boldsymbol{x})$ in (4) satisfies the following dynamic programming principle.

Lemma 2. *For* $\boldsymbol{x}_0 \in \widehat{X}$ *and* $k \in \mathbb{N}_{\geq 1}$*,*

$$V(\boldsymbol{x}_0) = \inf_{\pi \in \Pi} \max\big\{ \sup_{l \in \mathbb{N}_{\leq k-1}} \alpha^l 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l)), \alpha^k V(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k)) \big\}.$$

According to Lemma 2, we deduce that the function $V(\boldsymbol{x})$ in (4) is the unique bounded solution to a Bellman-type equation.

Theorem 1. *The function* $V(\cdot) : \widehat{X} \to \mathbb{R}$ *in* (4) *is the unique bounded solution to the following Bellman-type equation:*

$$\min\big\{V(\boldsymbol{x}) - 1_{\mathtt{TR}}(\boldsymbol{x}), V(\boldsymbol{x}) - \alpha \inf_{\boldsymbol{d} \in D} V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}))\big\} = 0. \tag{5}$$

Generally, it is challenging to solve the Bellman-type equation (5) analytically. Due to the uniqueness of bounded solutions to (5) with $\alpha \in (0, 1)$, an estimate of the robust reach-avoid set RA can be obtained via the value iteration algorithm.

Theorem 2. *Suppose the sequence of functions* $(V_i(\boldsymbol{x}))_{i \in \mathbb{N}}$ *with* $V_i(\cdot) : \widehat{X} \to \mathbb{R}$ *is generated by the value iteration algorithm starting from some bounded function* $V_0(\cdot) : \widehat{X} \to \mathbb{R}$ *according to*

$$V_{i+1}(\boldsymbol{x}) = \max\{1_{\mathtt{TR}}(\boldsymbol{x}), \alpha \inf_{\boldsymbol{d} \in D} V_i(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}))\} \tag{6}$$

for $\boldsymbol{x} \in \widehat{X}$ *and* $i \in \mathbb{N}$*, where* $\alpha \in (0, 1)$*, then the function* $V_i(\boldsymbol{x})$ *uniformly approximates* $V(\boldsymbol{x})$ *over* $\widehat{X}$ *as* i *tends to infinity, where* $V(\boldsymbol{x})$ *is the unique bounded solution to equation* (5)*.*

Generally, the practical implementation of the value iteration algorithm requires covering the state space $\widehat{X}$ and disturbance space D with a discrete mesh of sufficient resolution, which exhibits exponential blow-up in complexity with

increasing size of the system state and disturbance variables. Please refer to [28] for more details. However, the robust reach-avoid set computed from such an implementation is just an approximation of the robust reach-avoid set $\mathtt{RA}$, which is neither an outer-approximation nor inner-approximation and thus may not fulfill reach-avoid specifications in certain formal designs which require the system starting from the computed robust reach-void set to reach the target set safely. To circumvent this issue, we in the sequel reformulate equation (5) into another equivalent form and further obtain a set of quantified inequality constraints for inner-approximating the robust reach-avoid set, which can be efficiently addressed by semi-definite programming methods.

Remark 4. *If $\alpha = 1$ in* (4)*, that is,*

$$V(\boldsymbol{x}) = \inf_{\pi\in\Pi} \sup_{l\in\mathbb{N}} 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}}(l)),$$

we have that $\mathtt{RA}' = \{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) = 1\}$, where $\mathtt{RA}'$ is the set presented in Remark 2. We can show that $V(\boldsymbol{x})$ is a solution to equation (5) *with $\alpha = 1$. However, the uniqueness of bounded solutions to this equation cannot be guaranteed, since $V(\boldsymbol{x}) \equiv \beta$ for $\boldsymbol{x} \in \widehat{X}$ is also a solution to equation* (1) *with $\alpha = 1$, where β is an arbitrary but fixed value being larger than or equal to 1. Also, we cannot obtain a set of inequality constraints for inner-approximating the set $\mathtt{RA}'$. This point will be further clarified in Remark 5 in Subsection III-B.*

B. Inner-approximating Robust Reach-avoid Sets

In this subsection, based on equation (5), we construct a set of novel constraints for inner-approximating the robust reach-avoid set $\mathtt{RA}$, i.e., to compute an inner-approximation of the robust reach-avoid set $\mathtt{RA}$. The problem of solving this set of constraints can be addressed within the semi-definite programming framework.

The direct relaxation of equation (5) by removing the minimum/infinimum operator leads to the following constraints

$$\begin{aligned} &V(\boldsymbol{x}) - 1 \geq 0, \forall \boldsymbol{x} \in \mathtt{TR},\\ &V(\boldsymbol{x}) \geq 0, \forall \boldsymbol{x} \in \widehat{X} \setminus \mathtt{TR},\\ &V(\boldsymbol{x}) \geq \alpha V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d})), \forall \boldsymbol{x} \in \widehat{X}, \forall \boldsymbol{d} \in D, \end{aligned} \tag{7}$$

which cannot be used for inner-approximating the robust reach-avoid set $\mathtt{RA}$. Therefore, we reformulate equation (5) into another equivalent form such that its straightforward relaxation can lead to a system of inequalities for inner-approximating the robust reach-avoid set $\mathtt{RA}$.

It is easy to obtain that $V(\cdot) : \widehat{X} \to \mathbb{R}$ in (4) satisfies

$$V(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \mathtt{TR},\\ 0, & \text{if } \boldsymbol{x} \in \widehat{X} \setminus X. \end{cases}$$

According to the Bellman-type equation (5), we have that for $\boldsymbol{x} \in X \setminus \mathtt{TR}$,

$$\min\{V(\boldsymbol{x}), V(\boldsymbol{x}) - \alpha \inf_{\boldsymbol{d}\in D} V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}))\} = 0$$

holds. Since $V(\cdot) : \widehat{X} \to \mathbb{R}$ in (4) is larger than or equal to zero over $\widehat{X}$, we have that

$$V(\boldsymbol{x}) = \alpha \inf_{\boldsymbol{d}\in D} V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d})), \forall \boldsymbol{x} \in X \setminus \mathtt{TR}.$$

Consequently, $V(\cdot) : \widehat{X} \to \mathbb{R}$ in (4) satisfies

$$V(\boldsymbol{x}) = \begin{cases} \alpha \inf_{\boldsymbol{d}\in D} V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d})), \forall \boldsymbol{x} \in X \setminus \mathtt{TR},\\ 1, \quad \text{if } \boldsymbol{x} \in \mathtt{TR},\\ 0, \quad \text{if } \boldsymbol{x} \in \widehat{X} \setminus X. \end{cases}$$

This conclusion is formally presented in Lemma 3.

Lemma 3. *The function $V(\boldsymbol{x}) : \widehat{X} \to \mathbb{R}$ in* (4) *with $\alpha \in (0, 1)$ is the unique bounded solution to the following system of equations:*

$$v(\boldsymbol{x}) = \alpha \inf_{\boldsymbol{d}\in D} v(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d})), \forall \boldsymbol{x} \in X \setminus \mathtt{TR}, \tag{8}$$

$$v(\boldsymbol{x}) = 1, \forall \boldsymbol{x} \in \mathtt{TR}, \tag{9}$$

$$v(\boldsymbol{x}) = 0, \forall \boldsymbol{x} \in \widehat{X} \setminus X. \tag{10}$$

Based on the system of equations in Lemma 3, a system of inequalities for inner-approximating the robust reach-avoid set $\mathtt{RA}$ has been derived, as shown in Corollary 1.

Corollary 1. *Given $\alpha \in (0, 1)$, if a bounded function $v(\cdot) : \widehat{X} \to \mathbb{R}$ satisfies the following constraints*

$$v(\boldsymbol{x}) \leq \alpha v(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d})), \forall \boldsymbol{x} \in X \setminus \mathtt{TR}, \forall \boldsymbol{d} \in D, \tag{11}$$

$$v(\boldsymbol{x}) \leq 1, \forall \boldsymbol{x} \in \mathtt{TR}, \tag{12}$$

$$v(\boldsymbol{x}) \leq 0, \forall \boldsymbol{x} \in \widehat{X} \setminus X, \tag{13}$$

then the strict zero upper level set of the function $v(\cdot) : \widehat{X} \to \mathbb{R}$ is an inner-approximation of the robust reach-avoid set $\mathtt{RA}$, i.e.,

$$\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) > 0\} \subseteq \mathtt{RA}.$$

Remark 5. *When $\alpha = 1$ in* (11)*, $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) > 0\}$ cannot be guaranteed to be an inner-approximation of the set $\mathtt{RA}'$, where $v(\boldsymbol{x})$ is a bounded solution to constraints* (11)-(13)*. For instance, $v(\boldsymbol{x})$ satisfying $v(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in X$ and $v(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \widehat{X} \setminus X$ is a bounded solution to constraints* (11)-(13)*, but $X = \{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) > 0\} \subseteq \mathtt{RA}'$ does not hold generally.*

From Corollary 1, we obtain that an inner-approximation of the robust reach-avoid set $\mathtt{RA}$ can be computed via solving the system of inequalities (11)-(13) which can be equivalently transformed into the following inequalities (14)-(16) via removing the indicator functions:

$$v(\boldsymbol{x}) \leq \alpha v(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d})), \forall \boldsymbol{x} \in X \setminus \mathtt{TR}, \forall \boldsymbol{d} \in D, \tag{14}$$

$$v(\boldsymbol{x}) \leq 1, \forall \boldsymbol{x} \in \mathtt{TR}, \tag{15}$$

$$v(\boldsymbol{x}) \leq 0, \forall \boldsymbol{x} \in \widehat{X} \setminus X. \tag{16}$$

Remark 6. *When $\mathtt{TR} = \cup_{i=1}^{l} \mathtt{TR}_i$ with $\mathtt{TR}_i = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) < 0\}$, the system of inequalities* (14)-(16) *can be rewritten equivalently as*

$$v(\boldsymbol{x}) \leq \alpha v(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d})), \forall \boldsymbol{x} \in X \cap \mathtt{TR}', \forall \boldsymbol{d} \in D, \tag{17}$$

$$v(\boldsymbol{x}) \leq 1, \forall \boldsymbol{x} \in \mathtt{TR}_i, i = 1, \ldots, l, \tag{18}$$

$$v(\boldsymbol{x}) \leq 0, \forall \boldsymbol{x} \in \widehat{X} \setminus X, \tag{19}$$

where $\mathtt{TR}' = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_1(\boldsymbol{x}) \geq 0, \ldots, g_l(\boldsymbol{x}) \geq 0\}$.

If only polynomials $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ are searched for solving the system of inequalities (14)-(16), they could be obtained via encoding the system of inequalities (14)-(16) using the

$$
\begin{aligned}
&\max \boldsymbol{c}^\top \cdot \hat{\boldsymbol{w}} \\
&\text{s.t.} \\
&\alpha v(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{d})) - v(\boldsymbol{x}) + s_0(\boldsymbol{x},\boldsymbol{d})h(\boldsymbol{x}) \\
&\qquad - s_1(\boldsymbol{x},\boldsymbol{d})g(\boldsymbol{x}) + s_2(\boldsymbol{x},\boldsymbol{d})q(\boldsymbol{d}) \in \sum[\boldsymbol{x},\boldsymbol{d}], \\
&1 - v(\boldsymbol{x}) + s_3(\boldsymbol{x})g(\boldsymbol{x}) \in \sum[\boldsymbol{x}], \\
&-v(\boldsymbol{x}) + s_4(\boldsymbol{x})h_0(\boldsymbol{x}) - s_5(\boldsymbol{x})h(\boldsymbol{x}) \in \sum[\boldsymbol{x}],
\end{aligned} \tag{20}
$$

where $\boldsymbol{c}^\top \cdot \hat{\boldsymbol{w}} = \int_{\widehat{X}} v(\boldsymbol{x})d\boldsymbol{x}$, $\hat{\boldsymbol{w}}$ is the constant vector computed by integrating the monomials in $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ over the set $\widehat{X}$, $\boldsymbol{c}$ is the vector composed of unknown coefficients in $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and $s_i(\boldsymbol{x},\boldsymbol{d}) \in \mathbb{R}[\boldsymbol{x},\boldsymbol{d}]$, $i = 0,\ldots,2$, $s_j(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$, $j = 3,4,5$.

sum-of-squares decomposition for multivariate polynomials and further solving the resulting semi-definite program (20).

Theorem 3. *If a function $v(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ satisfies the semi-definite program* (20)*, the set $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) > 0\}$ is an inner approximation of the robust reach-avoid set* RA*.*

Herein, we give a discussion on the computational complexity of our semi-definite programming method. The semi-definite program (20) belongs to the category of convex optimization, which can be efficiently solved by interior point methods in polynomial time. However, the size of semi-definite program (20) (i.e., the number of decision variables) increases exponentially with the total number of state and disturbance variables (i.e., $n+m$) and the degree of polynomials $(v(\boldsymbol{x}), s_i(\boldsymbol{x},\boldsymbol{d}), i = 0,\ldots,2, s_j(\boldsymbol{x}), j = 3,\ldots,5)$ [2]. For fixed degrees the size of semi-definite program (20) is polynomial with respect to the total number of state and disturbance variables; for a given system in which the total number of state and disturbance variables is fixed, the size of semi-definite program (20) is also polynomial with respect to the degree. In order to balance accuracy and computational cost, polynomials of appropriate degree should be chosen.

IV. Examples

This section presents four examples and evaluates the performance of inner-approximating the robust reach-avoid set via solving the set of inequalities (14)-(16) based on the semi-definite program (20). In order to gauge the quality of computed inner-approximations, we also present the robust reach-avoid sets obtained via the Monte-Carlo simulation method and the value iteration algorithm for solving equation (5) for the comparisons. All computations were performed on an i7-P51s 2.6GHz CPU with 32GB RAM running Window 10. The sum-of-squares programming problems are formulated using the sum-of-squares module YALMIP [18] and solved by the academic version of semi-definite programming solver MOSEK [21]. Since the work [29] considers the inner-approximating problem of reach-avoid sets for discrete-time polynomial systems free of disturbances, the first two examples, which involve systems without disturbances, were mainly used to compare the performances between the semi-definite programming method in [29] and the one (20). They were also used to discuss the effect of the discounted factor α on the computed inner-approximations. The related parameters for our semi-definite programming method and the value iteration algorithm are presented in Table I and II, respectively.

Ex.	α	d_v	d_s	T
1	0.999	12	20	12.92
1	0.99	12	20	12.02
1	0.9	12	20	14.52
2	0.999	12	20	11.93
2	0.99	12	20	10.92
2	0.9	12	20	12.22
3	0.999	10	18	154.67
4	0.999	8	8	556.85
4	0.9	8	8	514.50

TABLE I

PARAMETERS OF THE IMPLEMENTATIONS ON SOLVING (20) FOR EXAMPLES 1-4. α : THE DISCOUNTED FACTOR IN (20); d_v: DEGREE OF THE POLYNOMIAL v IN (20); d_s : DEGREE OF POLYNOMIALS s_i IN (20), $i = 0,\ldots,5$; T: COMPUTATION TIMES (SECONDS).

Ex.	α	ϵ	N	M
1	0.999	10^{-6}	10^4	-
2	0.999	10^{-6}	10^4	-
3	0.999	10^{-6}	10^4	10
4	0.999	10^{-6}	10^8	10

TABLE II

Parameters and performance of the value iteration on Examples 1-4. α: the discounted factor in (5)*; ϵ: the stopping criterion in the value iteration; N, M: numbers of uniform grids in the state and disturbance variable spaces respectively. For more explanations, please refer to [28].*

Example 1. *Consider the following discrete-time polynomial system from [29]:*

$$
\begin{cases}
x(l+1) = x(l) + 10^{-2}(-0.5x(l) - 0.5y(l) + 0.5x(l)y(l)) \\
y(l+1) = y(l) + 10^{-2}(-0.5y(l) + 1)
\end{cases}
$$

with $X = \{(x,y)^\top \mid x^2 + y^2 - 1 < 0\}$ and TR *$= \{(x,y)^\top \mid 10x^2 + 10(y-0.6)^2 - 1 < 0\}$.*

Due to the absence of disturbances, the robust reach-avoid set RA *is also equal to the reach-avoid set in [29] as stated in Remark 1. Both the reach-avoid sets computed via the simulation method and the value iteration algorithm for solving equation* (5) *are presented in Fig. 1. We observe from the visualized results in Fig. 1 that the one from the value iteration algorithm matches the one from the simulation method well. We compute inner-approximations from the semi-definite program* (20) *with $\alpha = 0.999$. The set*

$$\widehat{X} = \{(x,y)^\top \mid x^2 + y^2 - 1.1 \leq 0\}$$

is obtained for computations by solving the semi-definite program (3)*. The computed inner-approximations are presented in Fig. 1. Note that we also inner-approximate the reach-avoid set* RA *via the semi-definite programming method in [29] based on the parameters in Table I, which is also showcased in Fig. 1. The visualized results in Fig. 1 demonstrate that the computed inner-approximation from semi-definite programming method* (20) *is less conservative than the one from [29] but they almost coincide with each other.*

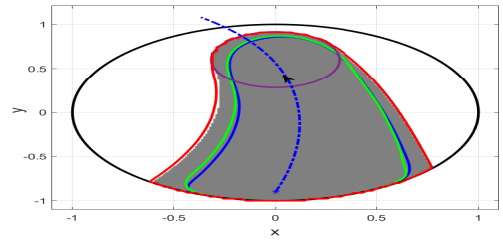

Fig. 1. An illustration of inner-approximating RA for Example 1. (Black and purple curves denote ∂X and ∂TR, respectively. Red curve denotes ∂RA obtained via the value iteration. Green and blue curves denote the boundaries of computed inner-approximations of RA from [29] and (20), respectively. Gray region denotes RA estimated via simulation methods. Blue dashded curve denotes one trajectory starting from $(0.0, -0.9)^\top$.)

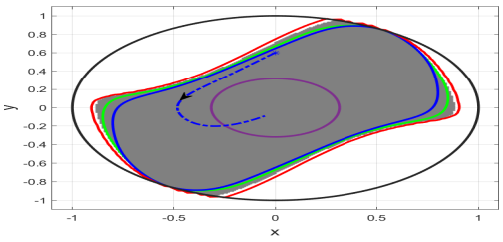

Fig. 2. An illustration of inner-approximating RA for Example 2. (Black and purple curves denote ∂X and ∂TR, respectively. Red curve denotes ∂RA obtained via the value iteration. Blue and green curves denote the boundaries of computed inner-approximations of RA from [29] and (20), respectively. Gray region denotes RA estimated via simulation methods. Blue dashed curve denotes one trajectory starting from $(0.0, 0.6)^\top$.)

Example 2. *Consider a computer-based model of the reversed-time Van der Pol oscillator from [29]:*

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

with $X = \{(x,y)^\top \mid x^2 + y^2 - 1 < 0\}$ *and* TR $= \{(x,y)^\top \mid 10x^2 + 10y^2 - 1 < 0\}$.

Analogous to Example 1, due to the absence of disturbances, the robust reach-avoid set RA *is equal to the reach-avoid set in [29] as stated in Remark 1. Both the reach-avoid sets computed via the simulation method and the value iteration algorithm for solving equation* (5) *are presented in Fig. 2, which shows that the one from the value iteration algorithm matches the one from the simulation method well. Also, we compute inner-approximations of the reach-avoid set* RA *via solving the semi-definite program* (20) *with* $\alpha = 0.999$. *The set*

$$\widehat{X} = \{(x,y)^\top \mid x^2 + y^2 - 1.1 \le 0\}$$

is obtained for computations by solving the semi-definite program (3). *The computed inner-approximation is also presented in Fig. 2, which also shows the computed inner-approximation from the semi-definite programming method in [29]. The visualized results demonstrate that the computed inner-approximation from semi-definite programming method* (20) *is less conservative than the one obtained from [29].*

Based on Examples 1 and 2, we discuss the effect of parameter α on the computed reach-avoid sets in our semi-definite programming method. Regarding the presence of α in constraint (14) and $\lim_{l\to\infty} \alpha^l = 0$, if there are some states slowly reaching the target set TR, the function $v(\boldsymbol{x})$ satisfying $v(\boldsymbol{x}) \le \alpha^l v(\boldsymbol{\phi}^\pi_{\boldsymbol{x}_0}(l)) \le \alpha^l$ (since $v(\boldsymbol{x}) \le 1$ for $\boldsymbol{x} \in$ TR) over these states will be approximately equal to zero in numerical computations and thus may not fall within the computed inner-approximation $\{\boldsymbol{x} \in X \mid v(\boldsymbol{x}) > 0\}$ in practical computations, leading to more conservative results, where $l = \inf\{i \in \mathbb{N} \mid \boldsymbol{\phi}^\pi_{\boldsymbol{x}_0}(i) \in \text{TR} \bigwedge \wedge_{j=0}^{i} \boldsymbol{\phi}^\pi_{\boldsymbol{x}_0}(j) \in X\}$. For instance, if $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99$, we obtain $\alpha^{50} = 1.00e-50, 1.13e-35, 7.18e-27, 1.27e-20, 8.89e-16, 8.09e-12, 1.80e-08, 1.43e-05, 5.17e-03, 0.61$. These datum are presented in Fig. 3. It is observed that α^{50} is almost zero when $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$. However, an estimate of RA for Examples 1 and 2 within the 50th time steps is too conservative, as illustrated in Fig. 4 and 6.

In order to circumvent this issue, our general suggestion for the parameter α is to take values as close to one as possible. This is why we choose $\alpha = 0.999$ in the semi-definite program (20) for Examples 1-2. In addition, we present the inner-approximations computed by solving the semi-definite program (20) with $\alpha = 0.99$ and $\alpha = 0.9$ for Examples 1 and 2 in Fig. 5 and 7, respectively. The visualized results in these two figures also show that both $\alpha = 0.99$ and $\alpha = 0.9$ lead to more conservative inner-approximations than $\alpha = 0.999$ for both Examples 1 and 2.

Example 3. *Consider the discrete-generation predator-prey model from [11],*

$$\begin{cases} x(j+1) = 0.5x(j) - x(j)y(j) \\ y(j+1) = -0.5y(j) + (d(j)+1)x(j)y(j) \end{cases}$$

with $X = \{(x,y)^\top \mid x^2 + y^2 - 1 \le 0\}$, $D = \{d \mid d^2 - 0.01 < 0\}$ *and* TR $= \{(x,y)^\top \mid 100(x^4 + y^4) - 1 < 0\}$.

The computed reach-avoid sets from the Monte-Carlo simulation method and the value iteration algorithm are presented in Fig. 8. It is observed that these two estimations match very well. Then, we compute inner-approximations of the robust reach-avoid set RA *via solving the semi-definite program* (20) *with* $\alpha = 0.999$. *The set*

$$\widehat{X} = \{(x,y)^\top \mid x^2 + y^2 - 1.6 \le 0\}$$

is obtained for computations by solving the semi-definite program (3). *The computed inner-approximations are presented in Fig. 8. The visualized results in Fig. 8 demonstrate that it almost coincides with the robust reach-avoid sets computed from the value iteration algorithm and the Monte-Carlo simulation method.*

Example 4. *Consider the discrete-time Clohessy-Wiltshire-Hill(CWH) equation, which describes the relative motion of chasing spacecraft for a target that is a circular orbit about a central body,*

$$\begin{cases} x(l+1) = x(l) + 0.01y(l) \\ y(l+1) = y(l) + 0.01(2\omega^2 x(l) + 2\omega w(l) + \frac{u_1+d}{m_c}) \\ z(l+1) = z(l) + 0.01w(l) \\ w(l+1) = w(l) + 0.01(-2\omega y(l) + \frac{u_2+d}{m_c}) \end{cases}$$

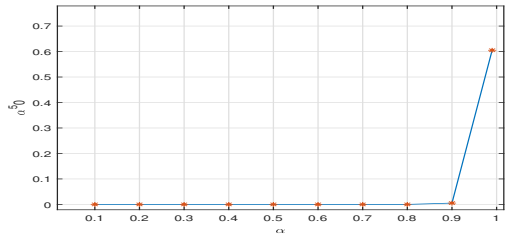

Fig. 3. An illustration of α^{50}.

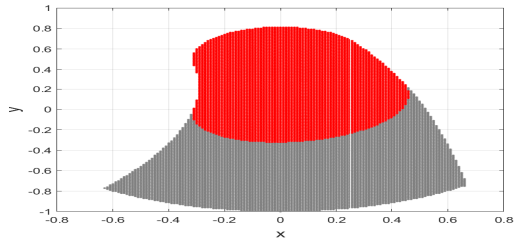

Fig. 4. An illustration of inner-approximating RA for Example 1 within the 50th time steps. (Gray region denotes an estimate of RA obtained via the simulation techniques within the 10000th steps. Red region denotes an estimate of RA obtained via the simulation techniques within the 50th steps.)

where $m_c = 100$, $\omega = 1$, $u_1 = 100$, $u_2 = 1$, $X = \{(x, y, z, w)^\top \mid x^2+y^2+w^2+z^2-0.25 < 0\}$, $D = \{d \mid d^2 - 1 \leq 0\}$ *and* $\mathtt{TR} = \{(x, y, z, w)^\top \mid x^2+y^2+w^2+z^2-0.1 < 0\}$.

For this example, we can not gain an estimate of the robust reach-avoid set RA *via solving equation* (1) *within one hour based on the parameters in Table II. The robust reach-avoid sets on the planes* $w = z = 0$ *and* $x = w = 0$ *computed by the Monte-Carlo simulation method are respectively presented Fig. 9 and 10. In the semi-definite program* (20)*, the set*

$$\widehat{X} = \{(x, y, z, w)^\top \mid x^2 + y^2 + z^2 + w^2 - 0.4 \leq 0\}$$

is obtained by solving the semi-definite program (3)*. Inner-approximations are computed with* $\alpha = 0.9$ *and* $\alpha = 0.999$*, which on the planes* $w = z = 0$ *and* $x = w = 0$ *are respectively presented Fig. 9 and 10. The visualized results show that although the computed inner-approximation with* $\alpha = 0.999$ *is less conservative than the one with* $\alpha = 0.9$*, these two inner-approximations do not have an inclusion relationship, i.e., the former does not include the latter.*

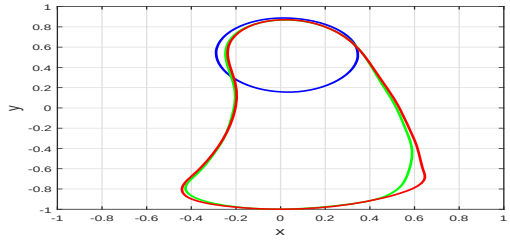

Fig. 5. An illustration of inner-approximating RA for Example 1. Red, green and blue curves respectively denote the boundaries of computed inner-approximations of RA with $\alpha = 0.999, 0.99$ and 0.9.

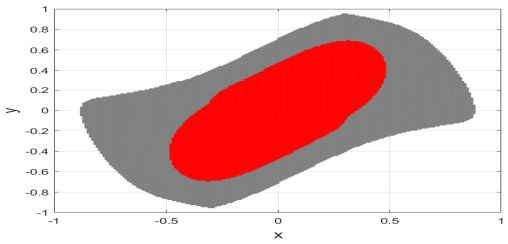

Fig. 6. An illustration of inner-approximating RA for Example 2 within the 50th time steps. (Gray region denotes an estimate of RA obtained via the simulation techniques within the 10000th steps. Red region denotes an estimate of RA obtained via the simulation techniques within the 50th steps.)

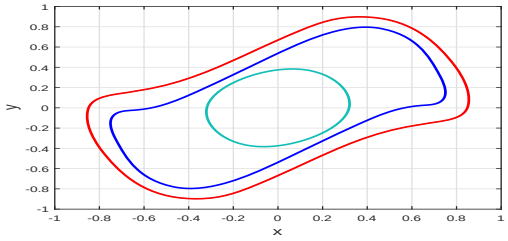

Fig. 7. An illustration of inner-approximating RA for Example 2. Red, blue and cyan curves respectively denote the boundaries of computed inner-approximations of RA with $\alpha = 0.999, 0.99$ and 0.9, respectively.

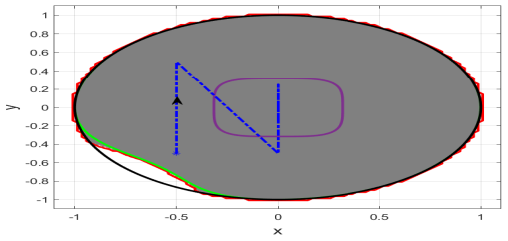

Fig. 8. An illustration of inner-approximating RA for Example 3. (Black and purple curves denote ∂X and ∂TR, respectively. Red curve denotes ∂RA obtained via the value iteration. Green curve denotes the boundary of computed inner-approximation of RA from (20). Gray region denotes RA estimated via the Monte-Carlo simulation method. Blue dashed curve denotes one trajectory starting from $(-0.5, 0.5)^\top$ driven by the disturbance trajectory $\pi = \{\mathbf{0}\}$.)

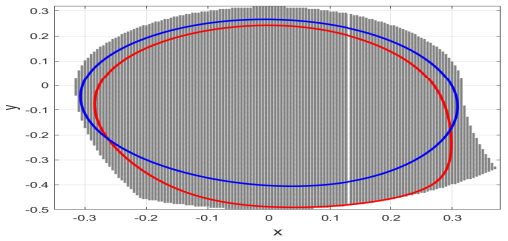

Fig. 9. An illustration of estimating RA on the plane $w = z = 0$ for Example 4. (Red and blue curves denote the boundaries of computed inner-approximations of RA from (20) with $\alpha = 0.99$ and $\alpha = 0.9$, respectively. Gray region denotes RA estimated via simulation methods.)

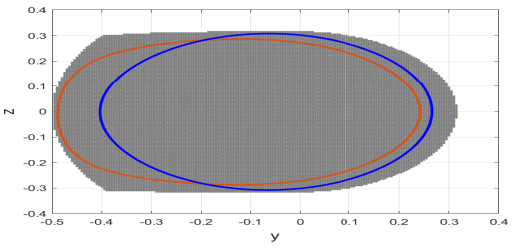

Fig. 10. An illustration of estimating RA on the plane $x = w = 0$ for Example 4. (Red and blue curves denote the boundaries of computed inner-approximations of RA from (20) with $\alpha = 0.99$ and $\alpha = 0.9$, respectively. Gray region denotes RA estimated via simulation methods.)

V. Conclusion

This paper investigated the problem of inner-approximating robust reach-avoid sets for discrete-time polynomial systems subject to disturbances over open time horizons, which is to determine initial states driving the system to meet the reach-avoid specification. Via defining a discounted value function, we reduced the robust reach-avoid set to the strict zero super-level set of the unique bounded solution to a Bellman-type equation. Based on this equation, we further obtained a set of novel quantified inequality constraints to inner-approximate the robust reach-avoid set, which was successfully transformed into a semi-definite programming problem that can be efficiently solved in polynomial time via interior point methods. Finally, the proposed methods were demonstrated and discussed in several examples.

In the future work, we would investigate the convergence of the proposed method and extend the proposed method to the reach-avoid sets generation problem of controlled discrete/continuous-time systems (e.g., [7]).

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. A. Ahmadi and A. Majumdar. Dsos and sdsos optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.

[3] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on Optimization*, 5(1):13–51, 1995.

[4] K. J. Åström and B. Wittenmark. *Computer-controlled systems: theory and design*. Courier Corporation, 2013.

[5] M. L. Borgne, A. Benveniste, and P. Le Guernic. Polynomial dynamical systems over finite fields. In *Algebraic Computing in control*, pages 212–222. Springer, 1991.

[6] D. Drzajic, N. Kariotoglou, M. Kamgarpour, and J. Lygeros. A semidefinite programming approach to control synthesis for stochastic reach-avoid problems. In *ARCH@ CPSWeek*, pages 134–143, 2016.

[7] C. Fan, Z. Qin, U. Mathur, Q. Ning, S. Mitra, and M. Viswanathan. Controller synthesis for linear system with reach-avoid specifications. *IEEE Transactions on Automatic Control*, 2021.

[8] M. Fränzle, M. Chen, and P. Kröger. In memory of oded maler: automatic reachability analysis of hybrid-state automata. *ACM SIGLOG News*, 6(1):19–39, 2019.

[9] O. Galor. *Discrete dynamical systems*. Springer Science & Business Media, 2007.

[10] 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*, pages 4283–4290. IEEE, 2017.

[11] A. Halanay and V. Rasvan. *Stability and stable oscillations in discrete time systems*, volume 2. CRC Press, 2000.

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

[13] N. Kariotoglou, S. Summers, T. Summers, M. Kamgarpour, and J. Lygeros. Approximate dynamic programming for stochastic reachability. In *ECC*, pages 584–589. IEEE, 2013.

[14] J. C. Knight. Safety critical systems: challenges and directions. In *ICSE*, pages 547–550, 2002.

[15] M. Kot. Discrete-time travelling waves: ecological examples. *Journal of mathematical biology*, 30(4):413–436, 1992.

[16] M. Kot and W. M. Schaffer. Discrete-time growth-dispersal models. *Mathematical Biosciences*, 80(1):109–136, 1986.

[17] E. A. Lee. Cyber physical systems: Design challenges. In *ISORC*, pages 363–369. IEEE, 2008.

[18] J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *IEE ICRA (IEEE Cat. No. 04CH37508)*, pages 284–289. IEEE, 2004.

[19] 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.

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

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

[22] C. P. Neuman and V. D. Tourassis. Discrete dynamic robot models. *IEEE transactions on systems, man, and cybernetics*, SMC-15(2):193–204, 1985.

[23] S. Shao, M. Chen, and P. Shi. *Robust Discrete-Time Flight Control of UAV with External Disturbances*, volume 317. Springer, 2021.

[24] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

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

[26] K.-C. Toh, M. J. Todd, and R. H. Tütüncü. Sdpt3—a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581, 1999.

[27] B. Xue, R. Li, N. Zhan, and M. Fänzle. Reach-avoid analysis for stochastic discrete-time systems. In *ACC*, pages 4867–4873. IEEE, 2021.

[28] B. Xue and N. Zhan. Robust invariant sets computation for discrete-time perturbed nonlinear systems. *IEEE Transactions on Automatic Control*, 67:1053–1060, 2021.

[29] B. Xue, N. Zhan, and M. Fränzle. Inner-approximating reach-avoid sets for discrete-time polynomial systems. In *CDC*, pages 457–476. IEEE, 2020.

[30] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0). *Optimization Methods and Software*, 18(4):491–505, 2003.

VI. Appendix

The proof of Lemma 1:

Proof. We first prove that $\mathtt{RA} \subseteq \{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\}$.

Assume that $\boldsymbol{x}_0 \in \mathtt{RA}$. Then, according to Proposition 1, the following conclusion holds:

$$\exists N \in \mathbb{N}. \forall \pi \in \Pi. \exists k \in \mathbb{N}_{\leq N}. \widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k) \in \mathtt{TR}.$$

Consequently, $\sup_{l\in\mathbb{N}} \alpha^l 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l)) \geq \alpha^N$ for $\pi \in \Pi$. Therefore, $V(\boldsymbol{x}_0) > 0$, implying that

$$\boldsymbol{x}_0 \in \{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\}$$

and thus $\mathtt{RA} \subseteq \{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\}$.

We in the following show that $\{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\} \subseteq \mathtt{RA}$.

Assume that $\boldsymbol{x}_0 \in \{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\}$. Thus, $V(\boldsymbol{x}_0) > 0$. Without loss of generality, we assume that $V(\boldsymbol{x}_0) = \epsilon_0 > 0$. Therefore, $\sup_{l\in\mathbb{N}} \alpha^l 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l)) \geq \epsilon_0$ for $\pi \in \Pi$. As a matter of fact that $\alpha^l \geq \alpha^l 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l))$ for $\pi \in \Pi$ and $\lim_{l\to\infty} \alpha^l = 0$, we conclude that there exists $N \in \mathbb{N}$ such that every possible trajectory $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}^n$ starting from $\boldsymbol{x}_0$ will hit the

target set $\mathtt{TR}$ within the finite time horizon $\mathbb{N}_{\leq N}$, irrespective of disturbances. According to Proposition 1, we have that $\boldsymbol{x}_0 \in \mathtt{RA}$. Therefore, $\{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\} \subseteq \mathtt{RA}$ holds.

Thus, we conclude that $\{\boldsymbol{x} \in X \mid V(\boldsymbol{x}) > 0\} = \mathtt{RA}$. □

The proof of Lemma 2:

Proof.

$$W(k, \boldsymbol{x}_0) := \inf_{\pi \in \Pi} \max\left\{\alpha^k V(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k)), \sup_{l \in \mathbb{N}_{\leq k-1}} \alpha^l 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l))\right\}.$$

We will prove that for $\epsilon > 0$, $|W(k, \boldsymbol{x}_0) - V(\boldsymbol{x}_0)| < \epsilon$.

According to the definition of $V(\boldsymbol{x}_0)$, i.e., (4), for any ϵ_1, there exists a disturbance trajectory $\pi' = (\boldsymbol{d}'(i))_{i \in \mathbb{N}} \in \Pi$ such that

$$V(\boldsymbol{x}) \geq \sup_{i \in \mathbb{N}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi'}_{\boldsymbol{x}_0}(i)) - \epsilon_1.$$

We then introduce two disturbance trajectories $\pi_1 = (\boldsymbol{d}_1(i))_{i \in \mathbb{N}} \in \Pi$ and $\pi_2 = (\boldsymbol{d}_2(i))_{i \in \mathbb{N}} \in \Pi$ with $\boldsymbol{d}_1(j) = \boldsymbol{d}'(j)$ for $j = 0, \ldots, k-1$ and $\boldsymbol{d}_2(j) = \boldsymbol{d}'(j+k)$ for $j \in \mathbb{N}$ respectively. Thus we obtain that

$$\begin{aligned}
W(k, \boldsymbol{x}_0) &\leq \max\left\{\alpha^k V(\boldsymbol{y}), \sup_{l \in \mathbb{N}_{\leq k-1}} \alpha^l 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi_1}_{\boldsymbol{x}_0}(l))\right\} \\
&\leq \max\Big\{ \sup_{i \in \mathbb{N}_{\geq k}} \{\alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi_2}_{\boldsymbol{y}}(i-k))\}, \\
&\qquad \sup_{i \in \mathbb{N}_{\leq k-1}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi_1}_{\boldsymbol{x}_0}(i))\Big\} \\
&= \max\Big\{ \sup_{i \in \mathbb{N}_{\geq k}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi'}_{\boldsymbol{x}_0}(i)), \\
&\qquad \sup_{i \in \mathbb{N}_{\leq k-1}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi'}_{\boldsymbol{x}_0}(i))\Big\} \\
&= \sup_{i \in \mathbb{N}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi'}_{\boldsymbol{x}_0}(i)) \\
&\leq V(\boldsymbol{x}_0) + \epsilon_1,
\end{aligned}$$

where $\boldsymbol{y} = \widehat{\boldsymbol{\phi}}^{\pi_1}_{\boldsymbol{x}_0}(k)$. Therefore,

$$V(\boldsymbol{x}_0) \geq W(k, \boldsymbol{x}_0) - \epsilon_1. \tag{21}$$

On the other hand, by the definition of $W(k, \boldsymbol{x}_0)$, for every $\epsilon_1 > 0$, there exists $\pi_1 = (\boldsymbol{d}_1(i))_{i \in \mathbb{N}} \in \Pi$ such that

$$W(k, \boldsymbol{x}_0) \geq \max\left\{\alpha^k V(\widehat{\boldsymbol{\phi}}^{\pi_1}_{\boldsymbol{x}_0}(k)), \sup_{i \in \mathbb{N}_{\leq k-1}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi_1}_{\boldsymbol{x}_0}(i))\right\} - \epsilon_1.$$

Also, by the definition of $V(\boldsymbol{x}_0)$, i.e., (4), for every $\epsilon_1 > 0$, there exists $\pi_2 = (\boldsymbol{d}_2(i))_{i \in \mathbb{N}} \in \Pi$ such that

$$V(\boldsymbol{y}) \geq \sup_{i \in \mathbb{N}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi_2}_{\boldsymbol{y}}(i)) - \epsilon_1,$$

where $\boldsymbol{y} = \widehat{\boldsymbol{\phi}}^{\pi_1}_{\boldsymbol{x}_0}(k)$. We define $\pi = (\boldsymbol{d}(i))_{i \in \mathbb{N}}$ such that $\boldsymbol{d}(i) = \boldsymbol{d}_1(i)$ for $i = 0, \ldots, k-1$ and $\boldsymbol{d}(i+k) = \boldsymbol{d}_2(i)$ for $i \in \mathbb{N}$. Obviously, $\pi \in \Pi$. Then, it follows

$$\begin{aligned}
W(k, \boldsymbol{x}_0) &\geq \max\Big\{ \sup_{i \in \mathbb{N}_{\geq k}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi_2}_{\boldsymbol{y}}(i-k)), \\
&\qquad \sup_{i \in \mathbb{N}_{\leq k-1}} \{\alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi_1}_{\boldsymbol{x}_0}(i))\}\Big\} - 2\epsilon_1 \\
&\geq \sup_{i \in \mathbb{N}} \alpha^i 1_{\mathtt{TR}}(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(i)) - 2\epsilon_1 \geq V(\boldsymbol{x}_0) - 2\epsilon_1.
\end{aligned}$$

Therefore,

$$V(\boldsymbol{x}_0) \leq W(k, \boldsymbol{x}_0) + 2\epsilon_1. \tag{22}$$

Combining (21) and (22), we finally have $|V(\boldsymbol{x}_0) - W(k, \boldsymbol{x}_0)| \leq \epsilon = 2\epsilon_1$. Since ϵ_1 is arbitrary, $V(\boldsymbol{x}_0) = W(k, \boldsymbol{x}_0)$ holds for $\boldsymbol{x}_0 \in \mathbb{R}^n$ and $k \in \mathbb{N}$. This completes the proof. □

The proof of Theorem 1:

Proof. The conclusion that the function $V(\boldsymbol{x})$ in (4) is a solution to the Bellman type equation (5) can be assured by Lemma 2 with $k = 1$: When $k = 1$,

$$V(\boldsymbol{x}) = \inf_{\pi \in \Pi} \max\left\{1_{\mathtt{TR}}(\boldsymbol{x}), \alpha V(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}}(1))\right\},$$

which is equal to $\max\left\{1_{\mathtt{TR}}(\boldsymbol{x}), \alpha \inf_{\boldsymbol{d} \in D} V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}))\right\}$. That is,

$$V(\boldsymbol{x}) - \max\left\{1_{\mathtt{TR}}(\boldsymbol{x}), \alpha \inf_{\boldsymbol{d} \in D} V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}))\right\} = 0,$$

which is equivalent to

$$\min\left\{V(\boldsymbol{x}) - 1_{\mathtt{TR}}(\boldsymbol{x}), V(\boldsymbol{x}) - \alpha \inf_{\boldsymbol{d} \in D} V(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}))\right\} = 0.$$

Below we just need to prove the uniqueness of bounded solutions to the Bellman type equation (5) with $\alpha \in (0, 1)$.

Assume that there exists another function $V'(\cdot) : \widehat{X} \to \mathbb{R}$ satisfying the equation (5) with $\alpha \in (0, 1)$, and there exists $\boldsymbol{y} \in \widehat{X}$ such that $V(\boldsymbol{y}) \neq V'(\boldsymbol{y})$.

Since both $V'(\cdot) : \widehat{X} \to \mathbb{R}$ and $V(\cdot) : \widehat{X} \to \mathbb{R}$ with $\alpha \in (0, 1)$ satisfy equation (5), we have that

$$\begin{aligned}
|V(\boldsymbol{y}) - V'(\boldsymbol{y})| = \Big| &\inf_{\boldsymbol{d} \in D} \max\{1_{\mathtt{TR}}(\boldsymbol{y}), \alpha V(\widehat{\boldsymbol{f}}(\boldsymbol{y}, \boldsymbol{d}))\} \\
&- \inf_{\boldsymbol{d} \in D} \max\{1_{\mathtt{TR}}(\boldsymbol{y}), \alpha V'(\widehat{\boldsymbol{f}}(\boldsymbol{y}, \boldsymbol{d}))\}\Big| \\
&\leq \alpha \sup_{\boldsymbol{d} \in D} \Big|V(\widehat{\boldsymbol{f}}(\boldsymbol{y}, \boldsymbol{d})) - V'(\widehat{\boldsymbol{f}}(\boldsymbol{y}, \boldsymbol{d}))\Big|.
\end{aligned}$$

Therefore,

$$|V(\boldsymbol{y}) - V'(\boldsymbol{y})| \leq \alpha^k \sup_{\pi \in \Pi} \Big|V(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{y}}(k)) - V'(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{y}}(k))\Big|, \forall k \in \mathbb{N}.$$

Also, due to the fact that the functions $V(\boldsymbol{x})$ and $V'(\boldsymbol{x})$ are bounded over $\boldsymbol{x} \in \widehat{X}$, we conclude that $|V(\boldsymbol{y}) - V'(\boldsymbol{y})| = 0$, which contradicts $V(\boldsymbol{y}) \neq V'(\boldsymbol{y})$.

Consequently, $V(\cdot) : \widehat{X} \to \mathbb{R}$ with $\alpha \in (0, 1)$ is the unique bounded solution to equation (5). □

The proof of Theorem 2:

Proof. According to (6), we have

$$\begin{aligned}
&|V_{i+1}(\boldsymbol{x}) - V_i(\boldsymbol{x})| \\
&= |\max\{1_{\mathtt{TR}}(\boldsymbol{x}), \alpha \inf_{\boldsymbol{d}_i \in D} V_i(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}_i))\} \\
&- \max\{1_{\mathtt{TR}}(\boldsymbol{x}), \alpha \inf_{\boldsymbol{d}_i \in D} V_{i-1}(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}_i))\}| \\
&\leq \max\{0, |\alpha \sup_{\boldsymbol{d}_i \in D} \Big(V_i\big(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}_i)\big) - V_{i-1}\big(\widehat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{d}_i)\big)\Big)|\} \\
&\leq \max\Big\{0, |\alpha^i \sup_{\boldsymbol{d}_1 \in D} \cdots \sup_{\boldsymbol{d}_i \in D} \Big(V_1(g(\boldsymbol{x}, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_i)) - \\
&\qquad V_0(g(\boldsymbol{x}, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_i))\Big)\Big|\}
\end{aligned} \tag{23}$$

where

$$g(\boldsymbol{x},\boldsymbol{d}_1,\cdots,\boldsymbol{d}_i)=\underbrace{\boldsymbol{f}\Big(\cdots\boldsymbol{f}}_{i}(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{d}_i),\boldsymbol{d}_{i-1}),\cdots,\boldsymbol{d}_1\Big).$$

Therefore, for $\forall l\in\mathbb{N}$ and $\forall k\in\mathbb{N}$, we have

$$\begin{aligned}
&|V_{l+k}(\boldsymbol{x})-V_l(\boldsymbol{x})|\\
&=|V_{l+k}(\boldsymbol{x})-V_{l+k-1}(\boldsymbol{x})+V_{l+k-1}(\boldsymbol{x})-V_{l+k-2}(\boldsymbol{x})\\
&\qquad\qquad+\cdots+V_{l+1}(\boldsymbol{x})-V_l(\boldsymbol{x})|\\
&\le|V_{l+k}(\boldsymbol{x})-V_{l+k-1}(\boldsymbol{x})|+|V_{l+k-1}(\boldsymbol{x})-V_{l+k-2}(\boldsymbol{x})|\\
&\qquad\qquad+\cdots+|V_{l+1}(\boldsymbol{x})-V_l(\boldsymbol{x})|\\
&\le\max\Big\{0,|\alpha^l|\sum_{j=0}^{k}\Big|\alpha^j\sup_{\boldsymbol{d}_1\in D}\cdots\sup_{\boldsymbol{d}_{l+j}\in D}\Big(V_1(g(\boldsymbol{x},\boldsymbol{d}_1,\ldots,\boldsymbol{d}_{l+j}))\\
&\qquad\qquad-V_0(g(\boldsymbol{x},\boldsymbol{d}_1,\ldots,\boldsymbol{d}_{l+j}))\Big)\Big|\Big\}
\end{aligned}\tag{24}$$

Moreover, since $V_0(\boldsymbol{x})$ and $1_{\texttt{TR}}(\boldsymbol{x})$ are bounded over $\widehat{X}$, therefore, $V_1(\boldsymbol{x})$ is bounded as well. Thus, according to (23), (24) and $\alpha\in(0,1)$, we have that $V_i(\boldsymbol{x})$ uniformly approximates a function $V'(\boldsymbol{x})$ over $\widehat{X}$ as i tends to infinity. In the rest we just need to prove that $V'(\boldsymbol{x})=V(\boldsymbol{x})$ over $\boldsymbol{x}\in\widehat{X}$. This conclusion can be assured by replacing $V_{i+1}(\boldsymbol{x})-V_i(\boldsymbol{x})$ in (23) with $V_{i+1}(\boldsymbol{x})-V(\boldsymbol{x})$, resulting in that $V_{i+1}(\boldsymbol{x})$ uniformly approximates $V(\boldsymbol{x})$ over $\widehat{X}$ as i tends to infinity, where $V(\boldsymbol{x})$ is the function in (4) with $\alpha\in(0,1)$. □

The proof of Lemma 3:

Proof. The fact that the function $V(\boldsymbol{x}):\widehat{X}\to\mathbb{R}$ in (4) with $\alpha\in(0,1)$ is the solution to the system of equations (8)-(10) was already discussed aforementioned.

We just show the uniqueness. That is, if there exists a function $v(\cdot):\widehat{X}\to\mathbb{R}$ satisfying the system of equations (8)-(10), $v(\boldsymbol{x})=V(\boldsymbol{x})$ over $\boldsymbol{x}\in\widehat{X}$, where $V(\cdot):\widehat{X}\to\mathbb{R}$ is the function in (4).

According to the definition of $V(\boldsymbol{x})$ in (4), we have that

$$V(\boldsymbol{x})=\begin{cases}\alpha^L, & \text{if } \boldsymbol{x}\in\texttt{RA}\setminus\texttt{TR},\\ 1, & \text{if } \boldsymbol{x}\in\texttt{TR},\\ 0, & \text{if } \boldsymbol{x}\notin\widehat{X}\setminus\texttt{RA},\end{cases}$$

where $L=\sup_{\pi\in\Pi}\{k\in\mathbb{N}\mid\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k)\in\texttt{TR}\}$.

We just need to prove that the function $v(\boldsymbol{x})$ over $\boldsymbol{x}\in\widehat{X}$ also satisfies the condition:

$$v(\boldsymbol{x})=\begin{cases}\alpha^L, & \text{if } \boldsymbol{x}\in\texttt{RA}\setminus\texttt{TR},\\ 1, & \text{if } \boldsymbol{x}\in\texttt{TR},\\ 0, & \text{if } \boldsymbol{x}\notin\widehat{X}\setminus\texttt{RA},\end{cases}$$

which implies that $v(\boldsymbol{x})=V(\boldsymbol{x})$ over $\boldsymbol{x}\in\widehat{X}$. We in the following prove that this condition holds. Assume that

$$L_\pi=\inf\{l\in\mathbb{N}\mid\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(l)\in\texttt{TR}\}.$$

1) $\boldsymbol{x}\in\texttt{TR}\cup[\widehat{X}\setminus X]$: it is obvious that $v(\boldsymbol{x})=V(\boldsymbol{x})$;
2) $\boldsymbol{x}\in\texttt{RA}\setminus\texttt{TR}$: since $\boldsymbol{x}\in\texttt{RA}$, $\sup_{\pi\in\Pi}L_\pi=\max_{\pi\in\Pi}L_\pi<\infty$, we conclude that

$$L=\sup_{\pi\in\Pi}L_\pi.\tag{25}$$

According to constraint (8), we have that

$$v(\boldsymbol{x})\le\alpha v(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d})),\forall\boldsymbol{d}\in D.\tag{26}$$

According to (25), there exists $\pi\in\Pi$ such that $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(L)\in\texttt{TR}$. Also, because of constraint (9), we have that $v(\boldsymbol{x})\le\alpha^L$. In the following, we show that

$$\forall\epsilon>0.\alpha^L-\epsilon\le v(\boldsymbol{x}).$$

According to constraint (8), it holds that

$$\begin{aligned}&\forall\epsilon_1>0.\exists\boldsymbol{d}_{\epsilon_1}\in D.\\&\alpha v(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1}))-\epsilon_1\le v(\boldsymbol{x})\le\alpha v(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1})).\end{aligned}\tag{27}$$

If $L=1$, implying that $\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d})\in\texttt{TR}$ for $\boldsymbol{d}\in D$, from (27), we thus have that $v(\boldsymbol{x})=\alpha$. The proof is completed; otherwise(i.e, $L>1$), according to constraints (27) and (9), we have that

$$\exists\delta_1>0.\forall\epsilon_1\in(0,\delta_1).\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1})\notin\texttt{TR}.\tag{28}$$

Therefore,

$$\begin{aligned}&\forall\epsilon_1\in(0,\delta_1).\forall\epsilon_2>0.\exists\boldsymbol{d}_{\epsilon_1,\epsilon_2}\in D.\\&\alpha^2v(\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1}),\boldsymbol{d}_{\epsilon_1,\epsilon_2}))-\epsilon_1-\alpha\epsilon_2\le v(\boldsymbol{x})\bigwedge\\&v(\boldsymbol{x})\le v(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1}))\le\alpha^2v(\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1}),\boldsymbol{d}_{\epsilon_1,\epsilon_2})).\end{aligned}\tag{29}$$

If $L=2$, implying that $\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1}),\boldsymbol{d})\in\texttt{TR}$ for $\boldsymbol{d}\in D$ when $\epsilon_1\in(0,\delta_1)$, thus, from (29), we have that

$$v(\boldsymbol{x})=\alpha^2.$$

The proof is completed; otherwise, according to constraints (29) and (9), we have that

$$\exists\delta_2>0.\forall\epsilon_1,\epsilon_2\in(0,\delta_2).\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d}_{\epsilon_1}),\boldsymbol{d}_{\epsilon_1,\epsilon_2})\notin\texttt{TR}.$$

Continuing the above deduction, we finally obtain that

$$\begin{aligned}&\bigwedge_{l=1}^{L-1}\forall\epsilon_l\in(0,\delta_{L-1}).\forall\epsilon_L>0.\exists\boldsymbol{d}_{\epsilon_1,\ldots,\epsilon_L}\in D.\\&\alpha^Lv(\widehat{\boldsymbol{\phi}}^{\pi_{\epsilon_L}}_{\boldsymbol{x}})-\sum_{l=1}^{L}\alpha^{l-1}\epsilon_l\le v(\boldsymbol{x})\le\alpha^Lv(\widehat{\boldsymbol{\phi}}^{\pi_{\epsilon_L}}_{\boldsymbol{x}}),\end{aligned}$$

where $\pi_{\epsilon_L}(l)=\boldsymbol{d}_{\epsilon_1,\ldots,\epsilon_l}$ for $l=1,\ldots,L$. For arbitrary but fixed $\epsilon>0$, taking $\delta_l\le(1-\alpha)\epsilon$ for $l=1,\ldots,L-1$, we have that $\sum_{l=1}^{L}\alpha^{l-1}\epsilon_l\le\epsilon$, i.e.

$$\alpha^Lv(\widehat{\boldsymbol{\phi}}^{\pi_{\epsilon_L}}_{\boldsymbol{x}})-\epsilon\le v(\boldsymbol{x})\le\alpha^Lv(\widehat{\boldsymbol{\phi}}^{\pi_{\epsilon_L}}_{\boldsymbol{x}}).$$

Also, due to (25), we have that $v(\boldsymbol{x})=\alpha^L=V(\boldsymbol{x})$.

3) $\boldsymbol{x}\in X\setminus\texttt{RA}$: we conclude that

$$\forall N\in\mathbb{N}.\exists\pi\in\Pi.\forall k\in\mathbb{N}_{\le N}.\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k)\in\widehat{X}\setminus\texttt{TR}.$$

Also, from constraints (8) and (10), we have that

$$v(\boldsymbol{x})=\alpha\inf_{\boldsymbol{d}\in D}v(\widehat{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{d})),\forall\boldsymbol{x}\in\widehat{X}\setminus\texttt{TR}.$$

Assume that the sets Π_1 and Π_2 satisfy

$$\forall\pi\in\Pi_1.\exists k\in\mathbb{N}.\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k)\in\texttt{TR}$$

and

$$\forall \pi \in \Pi_2. \forall k \in \mathbb{N}. \widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(k) \in \widehat{X} \setminus \texttt{TR},$$

respectively. Then, $\Pi_1 \cup \Pi_2 = \Pi$ holds.

3a) If $\Pi_2 = \emptyset$, we have that for arbitrary $M \in \mathbb{N}$, there exists $\pi \in \Pi_1$ such that $L_\pi \geq M$.

Therefore, according to constraint (8), we have that

$$v(\boldsymbol{x}) = \inf_{\pi \in \Pi_1} \{\alpha^{L_\pi} v(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}}(L_\pi))\}.$$

Consequently, $v(\boldsymbol{x}) = \inf_{\pi \in \Pi_1} \alpha^{L_\pi}$, implying that $v(\boldsymbol{x}) = 0 = V(\boldsymbol{x})$.

3b) If $\Pi_2 \neq \emptyset$, according to constraint (8), we have that

$$v(\boldsymbol{x}) = \inf\{\inf_{\pi \in \Pi_1} \{\alpha^{L_\pi} v(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}}(L_\pi))\}, \inf_{\pi \in \Pi_2} \{\alpha^{l} v(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}}(l))\}\}$$

for $l \in \mathbb{N}$. Thus,

$$v(\boldsymbol{x}) = \inf\{\inf_{\pi \in \Pi_1} \alpha^{L_\pi}, \inf_{\pi \in \Pi_2} \{\lim_{l \to \infty} \alpha^{l} v(\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}}(l))\}\},$$

implying that $v(\boldsymbol{x}) = \inf\{\inf_{\pi \in \Pi_1} \alpha^{L_\pi}, 0\} = 0$. Therefore, $v(\boldsymbol{x}) = 0 = V(\boldsymbol{x})$.

In summary, $V(\boldsymbol{x})$ is the unique bounded solution to the system of equations (8)-(10). □

The proof of Corollary 1:

Proof. Assume that $\boldsymbol{x}_0 \in \{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) > 0\}$. Without loss of generality, we further assume that $\epsilon_0 = v(\boldsymbol{x}_0) > 0$. Obviously, $\boldsymbol{x}_0 \in X$ from constraint (13).

Firstly, we show that all trajectories $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}$ starting from $\boldsymbol{x}_0$ do not leave the safe set X before the first target hitting time. This conclusion can be assured by constraints (11) and (13) and the fact that $v(\boldsymbol{x}_0) > 0$.

Then we show that there exists $N \in \mathbb{N}$ such that all trajectories $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}$ starting from $\boldsymbol{x}_0$ will hit the target set $\texttt{TR}$ within the finite time horizon $\mathbb{N}_{\leq N}$, regardless of disturbances.

1) Firstly, we show that all trajectories $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}$ starting from $\boldsymbol{x}_0$ will hit the target set $\texttt{TR}$ in finite time, regardless of disturbances. If this is not true, then there exists at least a trajectory driven by a disturbance trajectory π_0 such that $\widehat{\boldsymbol{\phi}}^{\pi_0}_{\boldsymbol{x}_0}(l) \in X \setminus \texttt{TR}$ for $l \in \mathbb{N}$. According to constraint (11), we have that

$$v(\widehat{\boldsymbol{\phi}}^{\pi_0}_{\boldsymbol{x}_0}(l)) \geq \frac{v(\boldsymbol{x}_0)}{\alpha^l} \geq \frac{\epsilon_0}{\alpha^l}$$

for $l \in \mathbb{N}$ and thus $\lim_{l \to \infty} v(\widehat{\boldsymbol{\phi}}^{\pi_0}_{\boldsymbol{x}_0}(l)) = \infty$, contradicting the fact that $v(\boldsymbol{x})$ is bounded over $\boldsymbol{x} \in \widehat{X}$. Therefore, all trajectories $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}$ starting from $\boldsymbol{x}_0$ will hit the target set $\texttt{TR}$ in finite time, regardless of the disturbance trajectory π.

2) Secondly, we show the existence of the upper bound $N \in \mathbb{N}$. Assume that $|v(\boldsymbol{x})| \leq M$ over $\boldsymbol{x} \in \widehat{X}$. Then, there exists $N' \in \mathbb{N}$ such that $\alpha^l M \leq \frac{\epsilon_0}{2}$ for $l \geq N'$. According to constraint (11) and $v(\boldsymbol{x}_0) = \epsilon_0$, any trajectory $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}$ for $\pi \in \Pi$ will leave the set $X \setminus \texttt{TR}$ within the finite time horizon $\mathbb{N}_{\leq N'}$. Due to the fact that all trajectories $\widehat{\boldsymbol{\phi}}^{\pi}_{\boldsymbol{x}_0}(\cdot) : \mathbb{N} \to \mathbb{R}$ for $\pi \in \Pi$ do not leave the set X before the first target hitting time, they will hit the target set $\texttt{TR}$ within the finite time horizon $\mathbb{N}_{\leq N'}$. The existence of the upper bound is shown.

Thus, $\boldsymbol{x}_0 \in \texttt{RA}$ and further $\{\boldsymbol{x} \in \widehat{X} \mid v(\boldsymbol{x}) > 0\} \subseteq \texttt{RA}$. The proof is completed. □

Changyuan Zhao received the B.Sc. degree in applied mathematics from University of Science and Technology of China, Hefei, China, in 2020. He is currently pursuing the Master degree in the Institute of Software, CAS, Beijing, China.

His research interests involve formal verification of hybrid systems and AI.

Shuyuan Zhang received the B.E. degree in automation from Northeastern University, Shenyang, China, in 2018. He is currently pursuing the Doctoral degree in control theory and control engineering with Beihang University, Beijing, China.

His major research fields include consensus control for multi-agent systems, and synchronization in complex networks.

Lei Wang received the B.E. degree in automation, the B.S. degree in applied mathematics, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2004, 2004, and 2009, respectively.

From 2014 to 2015, he was a Senior Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, and as a Senior Research Associate Professor with the Department of Mechanical Engineering, from 2015 to 2016. He is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His current research interests include modeling, control and optimization in complex networked systems.

Dr. Wang is currently the Associate Editor of the *ISA Transactions* (USA).

Bai Xue received the B.Sc. degree in information and computing science from Tianjin University of Technology and Education, Tianjin, China, in 2008, and the Ph.D. degree in applied mathematics from Beihang University, Beijing, China,in 2014.

He is currently a Research Professor with the Institute of Software, Chinese Academy of Sciences, Beijing, China, since September, 2021. Prior to joining the Institute of Software as an associate research professor in November 2017, he worked as a Research Fellow with the Centre for High Performance Embedded Systems, Nanyang Technological University, from May, 2014 to September, 2015, and as a postdoctoral with the Department für Informatik, Carl von Ossietzky Universität Oldenburg, from November, 2015 to October, 2017. His research interests involve formal verification of hybrid systems and AI.