

Reach-avoid Verification Based on Convex Optimization

Bai Xue^{1,2}, Naijun Zhan^{1,2}, Martin Fränzle³, Ji Wang⁴ and Wanwei Liu⁴

Abstract—In this paper we propose novel optimization-based methods for verifying reach-avoid (or, eventuality) properties of continuous-time systems modelled by ordinary differential equations. Given a system, an initial set, a safe set and a target set of states, we say that the reach-avoid property holds if for all initial conditions in the initial set, any trajectory of the system starting at them will eventually, i.e. in unbounded yet finite time, enter the target set while remaining inside the safe set until that first target hit. Based on a discount value function, two sets of quantified constraints are derived for verifying the reach-avoid property via the computation of exponential/asymptotic guidance-barrier functions (they form a barrier escorting the system to the target set safely at an exponential or asymptotic rate). It is interesting to find that one set of constraints whose solution is termed exponential guidance-barrier functions is just a simplified version of the existing one derived from the moment based method, while the other one whose solution is termed asymptotic guidance-barrier functions is completely new. Furthermore, built upon this new set of constraints, we derive a set of more expressive constraints, which includes the aforementioned two sets of constraints as special instances, providing more chances for verifying the reach-avoid properties successfully. When the involved datum are polynomials, i.e., the initial set, safe set and target set are semi-algebraic, and the system has polynomial dynamics, the problem of solving these sets of constraints can be framed as a semi-definite optimization problem using sum-of-squares decomposition techniques and thus can be efficiently solved in polynomial time via interior point methods. Finally, several examples demonstrate the theoretical developments and performance of proposed methods.

Index Terms—Ordinary Differential Equations; Reach-avoid Verification; Quantified Constraints

I. INTRODUCTION

Cyber-physical technology is integrated into an ever-growing range of physical devices and increasingly pervades our daily life [27]. Examples of such systems range from intelligent highway systems, to air traffic management systems, to computer and communication networks, to smart houses and smart supplies, etc. [11], [41]. Many of the above-mentioned applications are safety-critical and require a rigorous guarantee of safe operation.

Among the many possible rigorous guarantees, verifying whether the system's dynamics (generally modelled by ordinary differential equations) satisfy required properties is definitely in demand. Besides the more traditional properties such as stability and input-output performance [22], properties of interest also includes safety and reachability. Safety verification answers the problem whether a system starting from any initial condition in some prescribed set cannot evolve to some unsafe region in the state space. On the other hand, reachability verification aims to show that for some or all initial conditions in some prescribed set, the system will evolve to some target region in the state space [36]. In existing literature,

there are various extensions of safety and reachability such as reach-avoid properties to be verified. Reach-avoid verification is to verify whether all trajectories of the system starting from a specified initial set will enter a target set within a bounded time horizon or eventually while staying inside a safe set before the target hitting time. Since the reach-avoid property combines guarantees of safety by avoiding a set of unsafe states with the reachability property of reaching a target set and thus can formalize many important engineering problems such as collision avoidance [31] as well as target surveillance [19], its verification has turned out to be of fundamental importance in engineering. Mechanically providing mathematical guarantees for such scenarios is challenging but interesting due to practical applications mentioned above.

One of the popular methods for the reach-avoid verification is computational reachability analysis, which involves the explicit computation of reachable states. It has been widely studied over the last three decades in several disciplines including control theory, computer science and applied mathematics [3], [17]. In general, the exact computation of reach sets is impossible for dynamical and hybrid systems [18], [21]. Over-approximate reachability analysis is therefore studied in existing literature for verification purposes (e.g., [6]). Over-approximate reachability analysis computes an over-approximation (i.e., super-set) of the reach set based on set propagation techniques, which is an extension of traditional numerical methods for solving ordinary differential equations using set arithmetic rather than point arithmetic. If the computed over-approximation does not intersect with unsafe states and is included in the target set, the reach-avoid property holds. Overly pessimistic over-approximations, however, render many properties unverifiable in practice, especially for large initial sets and/or large time horizons. This pessimism mainly arises due to the wrapping effect [26], which is the propagation and accumulation of over-approximation errors through the iterative computation of reach sets. There are many techniques developed in existing literature for controlling the wrapping effect. One way is to use complex sets such as Taylor models [9], [12] and polynomial zonotopes [2] to over-approximate the reach set. On the other hand, as the extent of the wrapping effect correlates strongly with the size of the initial set, another way is to exploit subsets of the initial set for performing over-approximate reachability analysis via exploiting the (topological) structure of the system. For instance, appropriate corner points of reach sets, called bracketing systems, are used in [14], [37] to bound the complete reach sets when the systems under consideration are monotonic; [42] proposed the set-boundary reachability method for continuous-time systems featuring a locally Lipschitz-continuous vector field, which just need to perform over-approximate reachability analysis of the initial set's boundary for the reach-avoid verification.

Another popular method is the optimization based method, which transforms the verification problem into a problem of determining the existence of solutions to a set of quantified constraints. This method avoids the explicit computation of reach sets and thus can handle verification with unbounded time horizons. A well-known method is the barrier certificate method, which was originally proposed in [34], [35] for safety verification of continuous and hybrid systems.

1. State Key Lab. of Computer Science, Institute of Software Chinese Academy of Sciences, Beijing, China ({xuebai,znj}@ios.ac.cn)

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

3. Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany (martin.fraenzle@uni-oldenburg.de)

4. National University of Defense Technology, Changsha, China ({wj,wliu}@nudt.edu.cn; jiwang@ios.ac.cn)

The barrier certificate method was inspired by Lyapunov functions in control theory and relies on the computation of barrier certificates, which are a function of state satisfying a set of quantified inequalities on both the function itself and its Lie derivative along the flow of the system. In the state space, the zero level set of a barrier certificate separates an unsafe region from all system trajectories starting from a set of legally initial states and thus the existence of such a function provides an exact certificate/proof of system safety. Afterwards, a number of different kinds of barrier certificates were developed such as exponential barrier certificates and vector barrier certificates in the literature [8], [13], [23], [39], which mainly differ in their expressiveness. This method was also extended to reachability verification of continuous and hybrid systems. For instance, it was extended to the reach-avoid verification in [36].

In this paper we study the reach-avoid verification problem of continuous-time systems modelled by ordinary differential equations in the framework of the optimization based method. The reach-avoid verification problem of interest is that given an initial set, a safe set and a target set, we verify whether any trajectory starting from the initial set will eventually enter the target set while remaining inside the safe set until the first target hit. The reach-avoid verification problem in our method is transformed into a problem of searching for so called guidance-barrier functions. The reason that we term guidance-barrier function is that the boundary of its certain sub-level set forms a barrier, escorting the system to the target set safely. Based on a discount value function, whose certain (sub) level set equals the set of all initial states enabling the satisfaction of reach-avoid properties, with the discount factor being larger than and equal to zero we first respectively derive two sets of quantified constraints whose solutions are termed exponential and asymptotic guidance barrier functions. If a solution to any of these two sets of constraints is found, the reach-avoid property is guaranteed. Based on the set of constraints associated with asymptotic guidance-barrier functions, we further construct a set of more expressive constraints, which admits more solutions and formulates the aforementioned two sets of constraints as its special instances, and thus offers more possibilities of verifying the reach-avoid property successfully. When the datum are polynomials, i.e., the system has polynomial dynamics, the initial set, target set and safe set are semi-algebraic sets, the problem of solving these constraints can be reduced to a semi-definite programming problem which can be solved efficiently in polynomial time via interior point methods. Finally, several examples are provided to demonstrate theoretical developments of the proposed methods, and the experiment results show that they outperform existing ones.

The main contributions of this work are summarized below.

- 1) A novel unified framework is proposed for the reach-avoid verification of continuous-time systems modelled by ordinary differential equations such that the verification problem is transformed into a problem of searching for exponential/asymptotic guidance-barrier functions. In this framework, two sets of quantified inequalities are derived based on a discount value function. One is a simplified version of the existing one from the moment based method in [25] however, the other one is completely new.
- 2) The differences and respective benefits of the aforementioned two sets of constraints are discussed in detail. Based on these discussions, a set of more expressive constraints is further developed such that the aforementioned two sets of constraints are its special cases.

Related Work

Among the many possible extensions beyond reachability analysis, reach-avoid analysis has turned out to be of fundamental importance

in engineering, as it can formalize many important engineering problems such as collision avoidance [31] and target surveillance [19]. The reach-avoid problem comes in the two variants of computing a reach-avoid set and of verifying reach-avoid properties for systems featuring a given initial state set [42]. A reach-avoid set, also known as the capture basin in viability theory [7], is the maximal set of initial states such that a system starting from them is guaranteed to (eventually or within a given time horizon) reach a given target set while avoiding a given unsafe set till the target hit. The computation of reach-avoid sets is closely related to the reach-avoid verification, especially the computation of their under-approximations (i.e., subsets). If the given initial set is included in the computed under-approximation, it is evident that the reach-avoid property holds.

[36] presented a set of quantified constraints for the reach-avoid verification of continuous-time systems. One of constraints is that the Lie derivative of the computed function along the flow of the system is strictly negative over a set which is the closure of safe set minus target set. This condition is strict and thus limits its application. For instance, it is impossible for this constraint to be satisfied when the system has equilibrium states in this set. The derived constraints in the present work remedy this shortcoming.

On the other hand, the problem of computing reach-avoid sets of continuous-time systems has been investigated in the Hamilton-Jacobi reachability framework, e.g., [10], [15], [16], [30], [32], which links reach-avoid sets with viscosity solutions to Hamilton-Jacobi equations and finally reduces the problem of computing reach-avoid sets to the problem of addressing Hamilton-Jacobi equations. However, traditional numerical methods for solving Hamilton-Jacobi equations require gridding the state space such that the computed reach-avoid set is just an approximation, neither an over-approximation nor under-approximation, which cannot be used to reason formally on the system. More recently, via relaxing Hamilton-Jacobi equations, a method exploiting semi-definite programming for under-approximating reach-avoid sets has been suggested in [43]. Besides, moment-based programming methods were proposed for over- as well as under-approximating reach-avoid sets in [20], [24], [29], [38], [45]. All aforementioned works are confined in that they compute reach-avoid sets over specified bounded time horizons and thus cannot be used to study the reach-avoid verification problem in this work. Studies on computing under-approximations of reach-avoid sets over unbounded time horizons are rare by the best of our knowledge, with only moment-based programming methods attempting to address them [25]. The constraints developed in the present work include the ones in [25] as a special case and thus have more capabilities for the reach-avoid verification.

A set of quantified constraints, derived by reduction from a pertinent set of equations, was recently derived in [44]. It addresses the computation of under-approximations of reach-avoid sets for *discrete-time systems*. The present work is an extension of the work [44] via following the line of constructing quantified constraints. However, the construction is completely different, since the value function in [44] relies on a long-run average cost, which is constructed based on the polynomial function defining the target set. Such a characterization is not applicable to continuous-time systems, due to different topological dynamics between continuous- and discrete-time systems. For example, if trajectories for discrete-time systems starting from states outside a target set enter the target set, they may not touch the boundary of the target set. However, this assertion does not hold for continuous-time systems. Consequently, we define a value function of a new form, which is built upon the indicator function of the target set and a discount factor, and this new value function facilitates the gain of several sets of constraints for the reach-avoid verification of continuous-time systems.

This paper is structured as follows. In Section II, basic notions used throughout this paper and the reach-avoid verification problem of interest are introduced. Then our methods for the reach-avoid verification are elucidated in Section III. After demonstrating the performance of proposed methods on a series of examples in Section IV, we conclude this paper in Section V.

II. PRELIMINARIES

In this section we formally present the concepts of continuous-time systems and reach-avoid verification problem of interest in this paper. Before formulating them, let us introduce some basic notions used throughout this paper: for a function $v(\mathbf{x})$, $\nabla_{\mathbf{x}}v(\mathbf{x})$ denotes its gradient with respect to \mathbf{x} ; $\mathbb{R}_{\geq 0}$ ($\mathbb{R}_{>0}$) stands for the set of nonnegative (positive) real values in \mathbb{R} with \mathbb{R} being the set of real numbers; the closure of a set X is denoted by \overline{X} , the complement by X^c and the boundary by $@X$; the ring of all multivariate polynomials in a variable \mathbf{x} is denoted by $\mathbb{R}[\mathbf{x}]$; vectors are denoted by boldface letters, and the transpose of a vector \mathbf{x} is denoted by \mathbf{x}^T ; $\Sigma[\mathbf{x}]$ is used to represent the set of sum-of-squares polynomials over variables \mathbf{x} , i.e.,

$$\Sigma[\mathbf{x}] = \{ \sum_{i=1}^k p_i^2(\mathbf{x}) \mid p_i \in \mathbb{R}[\mathbf{x}], i = 1, \dots, k \}$$

All of semi-definite programs in this paper are formulated using Matlab package YALMIP [28] and solved by employing the academic version of the semi-definite programming solver MOSEK [33], and the computations were performed on an i7-P51s 2.6GHz CPU with 32GB RAM running Windows 10 and MATLAB2020a.

A. Preliminaries

The continuous-time system of interest (or, **CToIS**) is a system whose dynamics are described by an ODE of the following form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}); \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n; \quad (1)$$

where $\mathbf{x} = \frac{d\mathbf{x}(t)}{dt}$ and $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}); \dots; f_n(\mathbf{x}))^T$ with $f_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$.

We denote the trajectory of system **CToIS** that originates from $\mathbf{x}_0 \in \mathbb{R}^n$ and is defined over the maximal time interval $[0; T_{\mathbf{x}_0})$ by $\mathbf{x}_0(\cdot) : [0; T_{\mathbf{x}_0}) \rightarrow \mathbb{R}^n$. Consequently,

$$\mathbf{x}_0(t) := \mathbf{x}(t); \forall t \in [0; T_{\mathbf{x}_0}); \text{ and } \mathbf{x}_0(0) = \mathbf{x}_0;$$

where $T_{\mathbf{x}_0}$ is either a positive value (i.e., $T_{\mathbf{x}_0} \in \mathbb{R}_{>0}$) or ∞ .

Given a bounded safe set X , an initial set X_0 and a target set X_r , where

$$X = \{ \mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) < 0 \} \text{ with } @X = \{ \mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) = 0 \};$$

$$X_0 = \{ \mathbf{x} \in \mathbb{R}^n \mid l(\mathbf{x}) < 0 \}; \text{ and}$$

$$X_r = \{ \mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) < 0 \}$$

with $l(\mathbf{x}), h(\mathbf{x}), g(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, and $X_0 \subseteq X$ and $X_r \subseteq X$, the reach-avoid property of interest is defined as follows.

Definition 1 (Reach-Avoid Property): Given system **CToIS** with the safe set X , initial set X_0 and target set X_r , we say that the reach-avoid property holds if for all initial conditions $\mathbf{x}_0 \in X_0$, any trajectory $\mathbf{x}(t)$ of system **CToIS** starting at $\mathbf{x}_0(0) = \mathbf{x}_0$ satisfies

$$\mathbf{x}_0(t) \in X \quad \forall t \in [0; T]; \quad \mathbf{x}_0(t) \notin X_r$$

for some $T > 0$.

The problem of interest in this work is the reach-avoid verification, i.e., verifying that system **CToIS** satisfies the reach-avoid property in Definition 1. We attempt to solve this problem within the framework

of optimization based methods. Generally, such methods are sound but incomplete.

It is worth remarking here that the proposed methods in the subsequent sections can be extended straightforwardly to the case with the initial set X_0 and target set X_r being a finite union of multiple sets, i.e., $X = \bigcup_{i=1}^{k_1} X_i$, $X_0 = \bigcup_{k=1}^{k_1} X_{0,k}$, and $X_r = \bigcup_{j=1}^{j_1} X_{r,j}$, where $X_i = \{ \mathbf{x} \in \mathbb{R}^n \mid h_i(\mathbf{x}) < 0 \}$, $X_{0,k} = \{ \mathbf{x} \in \mathbb{R}^n \mid l_k(\mathbf{x}) < 0 \}$ and $X_{r,j} = \{ \mathbf{x} \in \mathbb{R}^n \mid g_j(\mathbf{x}) < 0 \}$, via using $h(\mathbf{x}) = \max_{i \in \{1, \dots, k_1\}} h_i(\mathbf{x})$, $l(\mathbf{x}) = \max_{k \in \{1, \dots, k_1\}} l_k(\mathbf{x})$ and $g(\mathbf{x}) = \max_{j \in \{1, \dots, j_1\}} g_j(\mathbf{x})$. For ease of exposition, we mainly take $k_1 = j_1 = 1$ in the sequel, if not explicitly stated otherwise.

B. Existing Methods

For the convenience of comparisons, in this subsection we recall two sets of quantified constraints in existing literature for verifying the reach-avoid property in Definition 1. The first one is from [36], while the other one is from [25].

Proposition 1: [36] Suppose that there exists a continuously differentiable function $v(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying

$$v(\mathbf{x}) > 0; \forall \mathbf{x} \in X_0 \quad (2)$$

$$v(\mathbf{x}) > 0; \forall \mathbf{x} \in @X \cap \overline{X_r}; \quad (3)$$

$$\nabla_{\mathbf{x}} v(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) < 0; \forall \mathbf{x} \in \overline{X} \cap \overline{X_r}; \quad (4)$$

Then the reach-avoid property in Definition 1 holds.

One of the drawbacks of constraints (2)-(4) in the reach-avoid verification is that it cannot deal with the case with an equilibrium being inside $\overline{X} \cap \overline{X_r}$, since $\mathbf{f}(\mathbf{x}_0) = 0$ implies $\nabla_{\mathbf{x}} v(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = 0$.

Besides, if the reach-avoid property in Definition 1 holds, the initial set X_0 must be a subset of the reach-avoid set RA , which is the set of all initial states guaranteeing the satisfaction of the reach-avoid property, i.e.

$$RA = \left\{ \mathbf{x}_0 \in \mathbb{R}^n \mid \begin{array}{l} \exists t \in [0; T_{\mathbf{x}_0}) : \mathbf{x}_0(t) \in X_r \\ \forall t \in [0; T_{\mathbf{x}_0}) : \mathbf{x}_0(t) \in X \end{array} \right\}$$

Therefore, the method for computing under-approximations of the reach-avoid set RA can be used for the reach-avoid verification. By adding the condition

$$v(\mathbf{x}) < 0; \forall \mathbf{x} \in X_0$$

into constraint [25, (18)], which is originally developed for under-approximating the reach-avoid set RA , we can obtain a set of quantified constraints as shown in Proposition 2 for the reach-avoid verification.

Proposition 2: Suppose that there exists a continuously differentiable function $v(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ and a continuous function $w(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying

$$v(\mathbf{x}) < 0; \forall \mathbf{x} \in X_0 \quad (5)$$

$$\nabla_{\mathbf{x}} v(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) - v(\mathbf{x}); \forall \mathbf{x} \in \overline{X} \cap \overline{X_r}; \quad (6)$$

$$w(\mathbf{x}) > 0; \forall \mathbf{x} \in \overline{X} \cap \overline{X_r}; \quad (7)$$

$$w(\mathbf{x}) - v(\mathbf{x}) + 1; \forall \mathbf{x} \in \overline{X} \cap \overline{X_r}; \quad (8)$$

$$v(\mathbf{x}) > 0; \forall \mathbf{x} \in @X; \quad (9)$$

where $\epsilon > 0$ is a user-defined value, then the reach-avoid property in Definition 1 holds.

III. REACH-AVOID VERIFICATION

This section presents our optimization based methods for the reach-avoid verification. Based on a discount value function, which is defined based on trajectories of a switched system and introduced in Subsection III-A, two sets of quantified constraints are first respectively derived when the discount factor is respectively equal to zero and larger than zero. Once a solution (termed exponential or asymptotic guidance-barrier function) to any of these two sets of constraints is found, the reach-avoid property in Definition 1 is verified successfully. Furthermore, inspired by the set of constraints obtained when the discount factor is zero, a set of more expressive constraints is constructed for the reach-avoid verification.

A. Induced Switched Systems

This subsection introduces a switched system, which is built upon system CTols. This switched system is constructed by requiring the state of system CTols to stay still when the complement of the safe set X is reached. For the sake of brevity, only trajectories of the induced switched system, also called CSPS, are introduced.

Definition 2: Given system CSPS with an initial state $x_0 \in \bar{X}$, if there is a function $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ with $x(0) = x_0$ such that it satisfies the dynamics defined by $\dot{x} = \phi(x)$, where

$$\phi(x) := 1_X(x) f(x); \quad (10)$$

with $1_X(\cdot) : \mathbb{R}^n \rightarrow \{0, 1\}$ representing the indicator function of the set X , i.e.,

$$1_X(x) := \begin{cases} 1; & \text{if } x \in X; \\ 0; & \text{if } x \notin X; \end{cases}$$

then the trajectory $b_{x_0}(\cdot) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$, induced by x_0 , of system CSPS is defined as follows:

$$b_{x_0}(t) := x(t); \forall t \in \mathbb{R}_{\geq 0};$$

It is observed that the set \bar{X} is an invariant set for system CSPS. Also, if $x_0 \in X$ and there exist $T \in \mathbb{R}_{\geq 0}$ such that $b_{x_0}(t) \in X$ for $t \in [0, T]$, we have that

$$b_{x_0}(t) = x_0(t); \forall t \in [0, T];$$

Also, trajectories of system CSPS evolving in the viable set \bar{X} can be classified into three disjoint groups:

- 1) trajectories entering the set X_r in finite time. It is worth remarking here that these trajectories will not leave the safe set X before reaching the target set X_r . Due to the fact that
- $$b_{x_0}(t) = x_0(t); \forall t \in [0, T]$$
- where $x_0 \in X$ and $T \in \mathbb{R}_{\geq 0}$ is a time instant such that $b_{x_0}(t) \in X$ for $t \in [0, T]$, we conclude that the set of initial states deriving these trajectories equals the reach-avoid set
- 2) trajectories entering the set X_r in finite time, but never entering the target set X_r ;
 - 3) trajectories staying in the set $X \setminus X_r$ for all time.

B. Discount Value Functions

The discount value function aforementioned is introduced in this subsection. With a non-negative discount factor the discount value function $V(x) : \bar{X} \rightarrow \mathbb{R}$ with a non-negative discount factor is defined in the following form:

$$V(x) := \sup_{t \in \mathbb{R}_{\geq 0}} e^{-t} 1_{X_r}(b_x(t)); \quad (11)$$

where $1_{X_r}(\cdot) : \mathbb{R}^n \rightarrow \{0, 1\}$ is the indicator function of the target set X_r . Obviously, $V(x)$ is bounded over the set \bar{X} . Moreover, if

$$V(x) = \begin{cases} 1; & \text{if } x \in X_r; \\ 0; & \text{otherwise} \end{cases} \quad (12)$$

$$V(x) = \begin{cases} 0; & \text{if } x \in \bar{X} \setminus X_r; \\ e^{-x}; & \text{if } x \in X_r; \end{cases} \quad (13)$$

$$x = \inf_{t \in \mathbb{R}_{\geq 0}} b_x(t) \in X_r$$

is the first hitting time of the target set X_r .

From (12) and (13), we have that following proposition.

Proposition 3: When $\gamma = 0$, the one level set of $V(\cdot) : \bar{X} \rightarrow \mathbb{R}$ in (11) is equal to the reach-avoid set RA , i.e.,

$$\{x \in \bar{X} \mid V(x) = 1\} = RA;$$

When $\gamma > 0$, the strict zero super level set of $V(\cdot) : \bar{X} \rightarrow \mathbb{R}$ in (11) is equal to the reach-avoid set RA , i.e.,

$$\{x \in \bar{X} \mid V(x) > 0\} = RA;$$

In the following we respectively obtain two sets of quantified constraints for the reach-avoid verification based on the discount value function $V(x)$ in (11) with $\gamma > 0$ and $\gamma = 0$. Through thorough analysis on these two sets of constraints, we further obtain a set of more expressive constraints for the reach-avoid verification.

C. Exponential Guidance-barrier Functions

In this subsection we introduce the construction of quantified constraints based on the discount value function $V(x)$ in (11) with $\gamma > 0$, such that the reach-avoid verification problem is transformed into a problem of determining the existence of an exponential guidance-barrier function.

The set of constraints is derived from a system of equations admitting the value function $V(x)$ as solutions, which is formulated in Theorem 1.

Theorem 1: Given system CSPS if there exists a continuously differential function $v(x) : \bar{X} \rightarrow [0, 1]$ such that

$$5 \quad v(x) f(x) = v(x); \forall x \in \bar{X} \setminus X_r; \quad (14)$$

$$v(x) = 0; \forall x \in X; \quad (15)$$

$$v(x) = 1; \forall x \in X_r; \quad (16)$$

then $v(x) = V(x)$ for $x \in \bar{X}$ and thus

$$\{x \in \bar{X} \mid v(x) > 0\} = RA;$$

where $V(\cdot) : \bar{X} \rightarrow \mathbb{R}$ is the value function with $\gamma > 0$ in (11).

Proof: We first consider that $x \in X_r$. If $x \in X_r$, $v(x) = 1 = e^0$ according to constraint (16). Therefore, we just need to consider $x \in X \setminus X_r$.

From (14), we have that

$$v(x) = e^{-x} v(b_x(\cdot)); \forall x \in [0, x];$$

where x is the first hitting time of the target set X_r . Due to constraint (16), we further have that

$$v(x) = e^{-x};$$

Next, we consider that $x \in \bar{X} \setminus X_r$, but its resulting trajectory $b_x(\cdot)$ will stay within the set $X \setminus X_r$ for all time. Due to constraint (14), we have that

$$v(x) = e^{-x} v(b_x(\cdot))$$

for $x \in \mathbb{R}^n$: Since $v(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is bounded, we have that

$$v(x) = 0 :$$

Finally, we consider that $x \in \mathbb{R}^n$ is a solution to the reach-avoid problem, but its resulting trajectory $b_x(\cdot)$ will touch the set X_r in finite time and never enter the target set X_r . Let

$$t_x^0 = \inf_{t \in \mathbb{R}^+} \{ t \mid b_x(t) \in X_r \}$$

be the first hitting time of the set X_r .

If $x \in \mathbb{R}^n$, $v(x) = 0$ holds from constraint (15). Otherwise, $v(x) > 0$. Further, from (14) we have

$$\dot{v}(x) = -\alpha v(x) - \beta \mathbb{1}_{x \in X_r}$$

Regarding constraint (15), which implies that $v(x) = 0$, we have $\dot{v}(x) = 0$.

Thus,

$$\dot{v}(x) = -\alpha v(x)$$

for $x \in \mathbb{R}^n$ according to (13) and $\dot{v}(x) = -\alpha v(x) > 0$.

Via relaxing the set of equations (14)-(16), we can obtain a set of inequalities for computing what we call exponential guidance-barrier function, whose existence ensures the satisfaction of the reach-avoid property in Definition 1. This set of inequalities is formulated in Proposition 4, in which inequalities (18) and (19) are obtained directly by relaxing equations (14) and (15), respectively.

Proposition 4: If there exists a continuously differentiable function $v(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$$v(x) > 0; \forall x \in X_0; \tag{17}$$

$$\exists \alpha > 0, \forall x \in \mathbb{R}^n, \dot{v}(x) \leq -\alpha v(x); \tag{18}$$

$$v(x) = 0; \forall x \in X_r; \tag{19}$$

where $\alpha > 0$ is a user-defined value, then the reach-avoid property in the sense of Definition 1 holds.

Comparing constraints (17)-(19) and (5)-(9), we find that the former is just a simplified version of the latter, i.e., if a function $v(x)$ satisfies constraints (5)-(9), $v(x)$ satisfies (17)-(19). The former can be obtained by removing constraints (7) and (8) and reversing the inequality sign in the rest of constraints in the latter. Therefore, we did not give a proof of Proposition 4 here. Also, due to its more concise form, constraint (17)-(19) will be used for discussions and comparisons instead of constraint (5)-(9) in the sequel.

If an exponential guidance-barrier function $v(x)$ satisfying constraint (17)-(19) is found, the reach-avoid property in Definition 1 holds, which is justified via Proposition 4. Moreover, it is observed that the set

$$R = \{ x \in \mathbb{R}^n \mid v(x) > 0 \}$$

is an under-approximation of the reach-avoid set, Reach-avoid . Also, due to constraint (18), we conclude that the set R is an invariant for system (1) until it enters the target set X_r , i.e., the boundary $\partial R = \{ x \in \mathbb{R}^n \mid v(x) = 0 \}$ is a barrier, preventing system (1) from leaving the set R and escorting system (1) to the target set X_r safely; furthermore, we observe that if $v(x)$ satisfies constraint (17)-(19), it must satisfy

$$\exists \alpha > 0, \forall x \in \mathbb{R}^n, \dot{v}(x) \leq -\alpha v(x); \tag{20}$$

This constraint indicates that trajectories starting from R will approach the target set X_r at an exponential rate of α . This is why we term a solution to the set of constraints (17)-(19) exponential guidance-barrier function. The above analysis also uncovers a necessary condition such that $v(x)$ is a solution to constraint (17)-(19), which is

$$R \cap X_r = \emptyset;$$

The condition of entering the target set at an exponential rate is strict for many cases, limiting the application of constraint (17)-(19) to the reach-avoid verification in practice. On the other hand, since the initial set X_0 should be a subset of the set R , the less conservative the set R is, the more likely the reach-avoid property is able to be verified. It is concluded from constraint (20) that the smaller α is, the less conservative the set R is inclined to be. This is illustrated in the following example.

Example 1: Consider an academic example,

$$\begin{cases} \dot{x} = 0.5x - 0.5y + 0.5xy \\ \dot{y} = 0.5y + 0.5 \end{cases} \tag{21}$$

with $X = \{ (x; y) \mid x^2 + y^2 - 1 < 0 \}$, $X_r = \{ (x; y) \mid (x + 0.2)^2 + (y - 0.7)^2 - 0.02 < 0 \}$ and $X_0 = \{ (x; y) \mid x < 0; x - 0.5 < 0; 0.8 - y < 0; y + 0.5 < 0 \}$.

In this example we use $\alpha = 0.1$ and $\beta = 1$ to illustrate the effect of α on the reach-avoid verification. Constraint (17)-(19) is solved by relaxing it into a semi-definite program, which is presented in Appendix, via sum-of-squares decomposition techniques. The degree of all polynomials in the resulting semi-definite program is taken to be the same and is taken in order for $\alpha = 0.1$; 2, 4, 6, 8, 10, ..., 20g. When the reach-avoid property is verified successfully, the computations will terminate. The degree is respectively 4 for $\alpha = 0.1$ and 20 for $\alpha = 1$ when termination. Both of the computed sets are showcased in Fig. 1, which almost collide with each other.

Fig. 1. Green, orange and black curve - ∂X_r , ∂X_0 and ∂X ; blue and red curve - ∂R , which are computed via respectively solving constraints (17)-(19) with $\alpha = 0.1$, and (17)-(19) with $\alpha = 1$.

It is worth emphasizing here that although the discount factor α can arbitrarily approach zero from above, it cannot be zero in constraint (17)-(19), since a function $v(x)$ satisfying this constraint with $\alpha = 0$ cannot rule out the existence of trajectories, which start from R and stay inside $X \setminus X_r$ for ever. Consequently, we do not recommend the use of too small α in practical numerical computations in order to avoid numerical issue (i.e., preventing the term $v(x)$ in the right hand of constraint (18) from becoming zero numerically due to floating point errors).

Although a set of constraints for the reach-avoid verification when $\alpha = 0$ cannot be obtained directly from (17)-(19), we will obtain one from the discount function (11) with $\alpha = 0$ in the sequel, expecting to remedy the shortcoming of the strict requirement of exponentially entering the set X_r when $\alpha > 0$.

D. Asymptotic Guidance-barrier Functions

In this subsection we elucidate the construction of constraints for the reach-avoid verification based on the discount value function $V(x)$ in (11) with $\alpha = 0$. In this case, the reach-avoid verification

problem is transformed into a problem of determining the existence of an asymptotic guidance-barrier function.

The set of constraints is constructed via relaxing a system of equations admitting the value function $v(x)$ in (11) with $\alpha = 0$ as solutions. These equations are presented in Theorem 2.

Theorem 2: Given system (1) and $w(x) : \bar{X} \rightarrow \mathbb{R}$ satisfying

$$5_x v(x) - f(x) = 0; 8x \in \bar{X} \cap \bar{X}_r; \quad (22)$$

$$v(x) = 5_x w(x) - f(x); 8x \in \bar{X} \cap \bar{X}_r; \quad (23)$$

$$v(x) = 0; 8x \in \mathcal{X}; \quad (24)$$

$$v(x) = 1; 8x \in X_r; \quad (25)$$

then $v(x) = V(x)$ for $x \in \bar{X}$ and thus

$$f(x) \in \bar{X} \cap \bar{X}_r \implies v(x) = 1 \text{ g} = \text{RA};$$

where $V(\cdot) : \bar{X} \rightarrow \mathbb{R}$ is the value function with $\alpha = 0$ in (11).

Proof: From (22), we have that

$$v(x) = v(b_x(\cdot)); 8 \in [0; x]; \quad (26)$$

where $x \in R_0$ is the time instant such that

$$b_x(\cdot) \in \bar{X} \cap \bar{X}_r; 8 \in [0; x];$$

For $x \in \text{RA}$, we can obtain that $v(x) = 1$ due to constraints (25) and (26).

In the following we consider $x \in \bar{X} \cap \text{RA}$.

We first consider $x \in \bar{X} \cap \text{RA}$, but its trajectory $b_x(\cdot) : R_0 \rightarrow \mathbb{R}^n$ stays within the set $\bar{X} \cap \bar{X}_r$. From (23), we have that

$$v(b_x(\cdot)) = 5_y w(y) - f(y) \Big|_{y=b_x(\cdot)}$$

for $\tau \in R_0$. Thus, we have that

$$\int_0^\tau v(b_x(\cdot)) d = \int_0^\tau 5_y w(y) - f(y) \Big|_{y=b_x(\cdot)} d$$

for $t \in R_0$ and further

$$v(x) = \frac{w(b_x(t)) - w(x)}{t}$$

for $t \in R_0$. Since $w(x)$ is continuously differentiable function over \bar{X} , it is bounded over $x \in \bar{X}$. Consequently $v(x) = 0$.

Next, we consider $x \in \bar{X} \cap \text{RA}$, but its trajectory $b_x(\cdot)$ will touch \mathcal{X} in finite time and never enters the target set. For such x , we can obtain that $v(x) = 0$ due to constraints (26) and (24).

Thus, according to (12),

$$v(x) = V(x)$$

for $x \in \bar{X}$. Further, from Lemma 3,

$$f(x) \in \bar{X} \cap \bar{X}_r \implies v(x) = 1 \text{ g} = \text{RA}$$

holds.

Based on the system of equations (22)-(25), we have a set \mathcal{X} of and $R \setminus X_r \in \mathbb{R}^n$. Inspired by the notion of asymptotic stability inequalities as shown in Proposition 5 for computing an asymptotic stability analysis, if system (1) starting from the set $R = f(x) \in \bar{X} \cap \bar{X}_r$ with $v(x)$ satisfying constraint (17)-(19), enters the avoid properties in the sense of Definition 1. In Proposition 5, target set X_r at an exponential rate, then system (1) starting from inequalities (28), (29) and (30) are obtained directly by relaxing the set $R = f(x) \in \bar{X} \cap \bar{X}_r$ with $v(x)$ satisfying constraint (27)-(30), would enter the target set X_r in an asymptotic sense. This is the reason that we term a function $v(x)$ satisfying constraint (27)-(30) asymptotic guidance-barrier function.

$$v(x) > 0; 8x \in X_r; \quad (27)$$

$$5_x v(x) - f(x) = 0; 8x \in \bar{X} \cap \bar{X}_r; \quad (28)$$

$$v(x) = 5_x w(x) - f(x) = 0; 8x \in \bar{X} \cap \bar{X}_r; \quad (29)$$

$$v(x) = 0; 8x \in \mathcal{X}; \quad (30)$$

then the reach-avoid property in the sense of Definition 1 holds.

Proof: Let $R = f(x) \in \bar{X} \cap \bar{X}_r$. We will show that

$$R \text{ RA} :$$

If it holds, we can obtain the conclusion since

$$X_0 \text{ RA} ;$$

which is obtained from constraint (27).

Let $x_0 \in R$. Obviously, $x_0 \in X$ due to constraint (30). If $x_0 \in X_r$, $x_0 \in \text{RA}$ holds obviously. Therefore, in the following we assume $x_0 \in R \cap X_r$. We will prove that there exists $\tau \in R_0$ satisfying

$$x_0(t) \in X_r \quad \forall t \in [0; \tau]; x_0(\tau) \in R :$$

Assume that there exists $\tau \in R_0$ such that

$$x_0(t) \in \mathcal{X} \quad \forall t \in [0; \tau]; x_0(\tau) \in R \cap X_r :$$

From (28), we have that

$$v(x_0(\tau)) - v(x_0) > 0$$

for $\tau \in [0; \tau]$, contradicting that $v(x) = 0$ for $x \in \mathcal{X}$.

Therefore, either

$$x_0(\tau) \in R \cap X_r; 8 \tau \in R_0 \quad (31)$$

or

$$\exists t \in R_0: x_0(t) \in X_r \quad \forall t \in [0; t]; x_0(t) \in R \quad (32)$$

holds.

Assume that (31) holds. From (28), we have that

$$v(x_0(\tau)) - v(x_0) > 0; 8 \tau \in R_0; \quad (33)$$

From (29), we have that for $\tau \in R_0$,

$$v(x_0(\tau)) = 5_x w(x) - f(x) \Big|_{x=x_0(\tau)}; \quad (34)$$

$$w(x_0(\tau)) - w(x_0) = \tau v(x_0); 8 \tau \in R_0;$$

implying that

$$\lim_{\tau \rightarrow +1} w(x_0(\tau)) = w(x_0) + 1 :$$

Since $v(\cdot) : \bar{X} \rightarrow \mathbb{R}$ is continuously differentiable and \bar{X} is compact, $w(\cdot) : \bar{X} \rightarrow \mathbb{R}$ is bounded. This is a contradiction. Thus, (32) holds, implying that $x \in \text{RA}$ and thus $R \text{ RA}$. ■

According to Proposition 5, the reach-avoid property in the sense of Definition 1 can be verified via searching for a feasible solution $v(x)$ to the set of constraints (27)-(30). Also, from the proof of Proposition 5, we find that the set $R = f(x) \in \bar{X} \cap \bar{X}_r$ will also be an invariant for system (1) until it enters the target set X_r .

Proposition 5: If there exist a continuously differentiable function $v(x) : \bar{X} \rightarrow \mathbb{R}$ and a continuously differentiable function $w(x) : \bar{X} \rightarrow \mathbb{R}$ satisfying constraint (27)-(30) asymptotic guidance-barrier function.

Comparing constraints (17)-(19) and (27)-(30), it is easy to find that constraint (27)-(30) to some extent. Unlike constraint (18) in (17)-(19), an auxiliary function $w(x)$ is introduced when $\alpha = 0$ in constraint (19), constraint (28) not only requires the Lie derivative of function (18). This constraint having $v(x)$, i.e., (29), excludes trajectories $v(x)$ along the flow of system $\dot{C}TolS$ to be non-negative over the starting from X_0 and staying inside X_r for ever. On the other set $R \cap \bar{X}_r$, but also over $\bar{X}_r \setminus X_r$. One simple solution to hand, constraint (28) keeps trajectories, starting from inside it remedy this defect is to combine constraints (27)-(30) and (17)-(19) before entering the target set. Also, we observe that constraint together, and obtain a set of constraints which is more expressive. (17)-(19) can be derived from the set of constraints (27)-(30) via these constraints are presented in Proposition 6. taking $w(x) = \frac{1}{5}v(x)$. In such circumstances, constraint (28) is Proposition 6: If there exist continuously differentiable functions redundant and should be removed. Further, if there exists a function $v_1(x); v_2(x) : \bar{X} \rightarrow \mathbb{R}$ and $w(x) : \bar{X} \rightarrow \mathbb{R}$ satisfying $v(x)$ satisfying constraint (20), there must exist a function $v(x)$, which can take $\frac{1}{5}v(x)$ for instance, such that

$$\begin{aligned} 5 \times v(x) - f(x) &\leq 0; \forall x \in \bar{R} \cap \bar{X}_r; \\ v(x) - 5 \times w(x) - f(x) &\leq 0; \forall x \in \bar{R} \cap \bar{X}_r \end{aligned} \quad (35)$$

holds. Thus, constraint (35) is more expressive than (20) and consequently is more likely to produce less conservative results in the following continue to use the scenario in Example 1 to illustrate this.

Example 2: Consider the scenario in Example 1 again. Similarly where $\alpha \in (0; +1)$, then the reach-avoid property in the sense of we also solve constraint (27)-(30) via relaxing it into a semi-definite Definition 1 holds. program, which is shown in Appendix. The reach-avoid property is Proof: Let $x_0 \in R = \{x \in \bar{X} \mid v_1(x) + v_2(x) > 0\}$, we have verified when polynomials of degree 2 in the resulting semi-definite that $x_0 \in \{x \in \bar{X} \mid v_1(x) > 0\}$ or $x_0 \in \{x \in \bar{X} \mid v_2(x) > 0\}$. program are taken. The computation is showcased in Fig. 2. Following Proposition 4 and 5, we have the conclusion. ■

Due to constraint (36) $v_1(x)$ may not be an asymptotic guidance-barrier function satisfying constraint (27)-(30). Similarly, $v_2(x)$ may not be an exponential guidance-barrier function satisfying constraint (17)-(19). Thus, constraint (36)-(41) is weaker than both of constraints (27)-(30) and (17)-(19). The set $\{x \in \bar{X} \mid v_1(x) + v_2(x) > 0\}$ is a mix of states entering the target set at an exponential rate and ones entering the target set at an asymptotic rate, thus we term $v_1(x) + v_2(x)$ asymptotic guidance-barrier function. However, we cannot guarantee that the set still satisfies $R \cap \bar{X}_r \subseteq \bar{X}_r$; and it is an invariant for system $\dot{C}TolS$ until it enters the target set X_r . If an initial state $x_0 \in X_0$ is a state such that $v_2(x) > 0$, then the trajectory starting from it will stay inside the set until it enters the target set X_r , since

$$\frac{d(v_1 + v_2)}{dt} = 5 \times v_1(x) - f(x) + 5 \times v_2(x) - f(x) - v_2(x); \forall x \in \bar{X} \cap \bar{X}_r$$

Fig. 2. Green, orange and black curve- \bar{X}_r , \bar{X}_0 and \bar{X} ; purple curve - \bar{R} computed via solving constraint (27)-(30).

Remark 1: As done in verifying invariance of a set using barrier certificate methods [4], one simple application scenario, reflecting the advantage of constraint (35) over (20) further, is on verifying whether trajectories starting from a given open set, which may be designed a priori via the Monte-Carlo simulation method, will enter the target set X_r eventually while staying inside it before the first target hit, where

$$R = \{x \in R^n \mid b(x) > 0\}$$

with $b(x)$ being continuously differentiable and $R \subseteq \bar{X}$. We are inclined to verifying whether there exists a continuously differentiable function $w(x)$ satisfying

$$\begin{aligned} 5 \times b(x) - f(x) &\leq 0; \forall x \in \bar{R} \cap \bar{X}_r; \\ b(x) - 5 \times w(x) - f(x) &\leq 0; \forall x \in \bar{R} \cap \bar{X}_r; \end{aligned}$$

instead of verifying whether there exists $\alpha \geq 0$ such that

$$5 \times b(x) - f(x) - \alpha b(x); \forall x \in \bar{R} \cap \bar{X}_r;$$

because the former is more expressive than the latter.

Although there are some benefits on the use of constraints (27)-(30) over (17)-(19) for the reach-avoid verification, there is still a defect caused by constraint (28), possibly limiting the application of Proposition 7: If there exists a continuously differentiable function $v(x) : \bar{X} \rightarrow \mathbb{R}$, a continuous function $(\cdot) : R^n \rightarrow \mathbb{R}$ satisfying

holds; otherwise, we cannot have such a conclusion. Instead, we have that $R_i = \{x \in \bar{X} \mid v_i(x) > 0\}$ satisfies $R_i \setminus X_r \subseteq \bar{X}_r$; and is an invariant for system $\dot{C}TolS$ until it enters the target set X_r , if $R_i \subseteq \bar{X}_r$; where $i \in \{1; 2\}$. Let's illustrate this via an example.

Example 3: Consider the scenario in Example 1 again. Similarly, we solve constraint (36)-(41) with $\alpha = 2$ via relaxing it into a semi-definite program, which is shown in Appendix. The reach-avoid property is verified when polynomials of degree 2 are taken. The computed \bar{R} is shown in Fig. 3. For this case, the set $\{x \in \bar{X} \mid v_2(x) > 0\}$ is empty. It is observed from Fig. 3 that the set does not intersect \bar{X}_r , and system $\dot{C}TolS$ leaves it before entering the target set X_r . However, the set R_1 is an invariant for system $\dot{C}TolS$ until it enters the target set X_r and $R_1 \setminus X_r \subseteq \bar{X}_r$.

The other more sophisticated solution of enhancing constraint (27)-(30) is to replace constraint (28) with

$$5 \times v(x) - f(x) - (\cdot); \forall x \in \bar{X} \cap \bar{X}_r;$$

where $(\cdot) : \bar{X} \rightarrow \mathbb{R}$ is a continuous function satisfying $(x) \leq 0$ over $\{x \in \bar{X} \cap \bar{X}_r \mid v(x) \leq 0\}$. One instance for (x) is $(x)v(x)$, where $(\cdot) : \bar{X} \rightarrow [0; +1]$. The new constraints are formulated in

experiments further support our theoretical analysis that constraint (42)-(45) is more expressive and provides more chances for verifying the reach-avoid property successfully.

Fig. 3. Green, orange and black curve- ∂X_r , ∂X_0 and ∂X ; blue and red curve - ∂R_1 and ∂R computed via solving constraint (36)-(41).

(x) 0 over $f(x) \geq \overline{X} \wedge \overline{X}_r \wedge v(x) \geq 0$, and a continuously differentiable function $w(x) : \overline{X} \rightarrow \mathbb{R}$ satisfying

$$v(x) > 0; \quad (42)$$

$$L_{f(x)} v(x) \geq 0; \quad (43)$$

$$v(x) \geq w(x) \geq 0; \quad (44)$$

$$v(x) \geq 0; \quad (45)$$

then the reach-avoid property in the sense of Definition 1 holds.

Proof: Let $R = \{x \in \overline{X} \mid v(x) > 0\}$. From constraint (43), we have that if $x_0(t) \in R \cap \overline{X}_r$,

$$L_{f(x)} v(x) \big|_{x=x_0(t)} \geq 0.$$

Then, following the arguments in the proof of Proposition 5, we have the conclusion.

Constraint (42)-(45) is less strict than constraint (27)-(30) in that the former only requires the Lie derivative $L_{f(x)} v(x)$ along the flow of system CTolS to be non-negative over the set \overline{X}_r rather than $\overline{X} \cap \overline{X}_r$, due to constraint (43). Constraint (42)-(45) does not impose any restrictions on the Lie derivative $L_{f(x)} v(x)$ along the flow of system CTolS over the set $\overline{X} \cap \overline{R}$. Moreover, it is more expressive since it degenerates to constraint (27)-(30) when $\epsilon = 0$, and it is more expressive than constraint (17)-(19), since the former degenerates to the latter when

$$v(x) = v(x)$$

and

$$w(x) = \frac{1}{\epsilon} v(x).$$

Besides, the set R obtained via solving constraint (42)-(45) will be an invariant for system CTolS until it enters the target set ∂X_r , and $R \setminus X_r \subseteq \mathbb{R}^n$.

Example 4: Consider the system in Example 1 with $\epsilon = 0.1$, $f(x; y) = x^2 + y^2 - 1 < 0$, $X_r = \{x > 0.2\} \cup \{y > 0.2\}$, $X_0 = \{x < 0.1\} \cup \{y < 0.1\}$, $\partial X_r = \{x = 0.2\} \cup \{y = 0.2\}$, $\partial X_0 = \{x = 0.1\} \cup \{y = 0.1\}$.

In this example we use $v(x) = x^2 + y^2$ to illustrate the benefits of constraints (42)-(45) on the reach-avoid verification. The constraint

(42)-(45) is solved by relaxing it into a semi-definite program which is presented in Appendix. The degree of all polynomials in the resulting semi-definite program is taken the same and in order of $\{2, 4, 6, 8, 10, \dots, 20\}$. When the reach-avoid property is verified successfully, the computations terminate. The degree for termination, and the computed set is showcased in Fig. 4. In contrast, the degree is 14 when using constraints (17)-(19) with $\epsilon = 0.1$ and (27)-(30) to verify the reach-avoid property. Consequently, these

Fig. 4. Green, orange and black curve- ∂X_r , ∂X_0 and ∂X ; purple curve - ∂R , which is computed via solving constraint (42)-(45).

IV. EXPERIMENTS

We further demonstrate the theoretical development and performance of the proposed methods on several examples, i.e., Examples 5-9. All of constraints (2)-(4), (17)-(19), (27)-(30), (36)-(41) and (42)-(45) are addressed via encoding them into semi-definite programs. The formulated semi-definite programs are presented in Appendix. In the computations, the degree of unknown polynomials in the resulting semi-definite programs is taken the same and in order of $\{2, 4, 6, 8, 10, \dots, 20\}$. When the reach-avoid property is verified successfully, the computations terminate. A return of 'Successfully solved (MOSEK)' from YALMIP will denote that a feasible solution is found, and the reach-avoid property is successfully verified.

Example 5: Consider the scenario in Example 4. As analyzed in Subsection III-C, the smaller the discount factor is in constraint (17)-(19), the more likely the reach-avoid property is able to be verified. Thus, in this example, we supplement some experiments involving constraint (17)-(19) with $\epsilon < 0.1$ for more comprehensive and fair comparisons with the proposed methods in the present work. In these experiments, $\epsilon = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ are used.

For all of these experiments, the computations terminate when the degree takes 14. All the results, including the ones in Example 4, further validate the benefits of constraint (42)-(45) over constraints (27)-(30) and (17)-(19) in terms of stronger expressiveness.

We also experimented using constraint (2)-(4), and the reach-avoid property is verified when the degree is 14.

Example 6 (Van der Pol Oscillator) Consider the reversed-time Van der Pol oscillator given by

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

with $X = \{x > 0.2\} \cup \{y > 0.2\}$, $X_0 = \{x < 0.1\} \cup \{y < 0.1\}$, $\partial X_r = \{x = 0.2\} \cup \{y = 0.2\}$, $\partial X_0 = \{x = 0.1\} \cup \{y = 0.1\}$.

The reach-avoid property is verified when the degree is 8, 8 and 12 for constraints (2)-(4), (27)-(30) and (17)-(19) with $\epsilon = 0.1$, respectively. We did not obtain any positive verification result from constraint (17)-(19) with $\epsilon = 1$. However, it can be improved by solving constraint (36)-(41) with $\epsilon = 1$ and the reach-avoid property is verified when the degree is 8. Further, if constraint (42)-(45) is used with $v(x) = x^2 + y^2$, the degree is 6. Some of the computed sets are illustrated in Fig. 5.

Besides, we also experimented using constraints (17)-(19) and (36)-(41) with $\gamma = 10^{-2}; 10^{-3}; 10^{-4}; 10^{-5}; 10^{-6}$ for more comprehensive and fair comparisons. The computations terminate when the degree takes 8 for all of these experiments.

Fig. 5. Green, black and orange curve - \mathcal{X}_r , \mathcal{X} and \mathcal{X}_0 ; blue, red, cyan and purple curve - \mathcal{R} , which is computed via solving constraints (27)-(30) when the degree is 8, (17)-(19) when the degree is 12 and $\gamma = 0:1$, (36)-(41) when the degree is 8 and $\gamma = 1$, and (42)-(45) when the degree is 6 and $\psi(x) = x^2v(x)$.

Example 7: Consider the following system from [40],

$$\begin{cases} \dot{x} = 0.42x - 1.05y - 2.3x^2 - 0.56xy - x^3 \\ \dot{y} = 1.98x + xy \end{cases}$$

with $X = \{x \in \mathbb{R}^2 \mid (x)^2 + (y)^2 - 4 < 0\}$, $X_0 = \{x \in \mathbb{R}^2 \mid (x - 1:2)^2 + (y - 0:8)^2 - 0:1 < 0\}$ and $X_r = \{x \in \mathbb{R}^2 \mid (x + 1:2)^2 + (y + 0:5)^2 - 0:3 < 0\}$.

The reach-avoid property in the sense of Definition 1 is not verified using constraint (2)-(4). Actually, it cannot be verified via solving constraint (2)-(4), since there exists an equilibrium in the set X_r .

The reach-avoid property is verified when the degree is 10 for constraints (27)-(30), and (17)-(19) with $\gamma = 1$ and $\gamma = 0:1$. If constraint (42)-(45) is used with $\psi(x) = (2 - y)v(x)$, the degree is 8. Some of the computed sets are illustrated in Fig. 6.

Like Example 6, we also experimented using constraints (17)-(19) and (36)-(41) with $\gamma = 10^{-2}; 10^{-3}; 10^{-4}; 10^{-5}; 10^{-6}$ for more comprehensive and fair comparisons. The computations terminate when the degree takes 8 for all of these experiments.

Fig. 6. Green, black and orange curve - \mathcal{X}_r , \mathcal{X} and \mathcal{X}_0 ; blue, red, cyan and purple curve - \mathcal{R} , which is computed via solving constraints (27)-(30) when the degree is 10, (17)-(19) when the degree is 10 and $\gamma = 1$, (17)-(19) when the degree is 10 and $\gamma = 0:1$, and (42)-(45) when the degree is 8 and $\psi(x) = (2 - y)v(x)$.

$$\begin{cases} \dot{x} = y \\ \dot{y} = (1 - x^2)x - y \end{cases}$$

with $X = \{x \in \mathbb{R}^2 \mid \frac{x^2}{4} + \frac{y^2}{9} - 1 < 0\}$, $X_0 = \{x \in \mathbb{R}^2 \mid (x+1)^2 + (y - 1:5)^2 - 0:25 < 0\}$ and $X_r = \{x \in \mathbb{R}^2 \mid x^2 + y^2 - 0:01 < 0\}$.

The reach-avoid property is not verified using constraint (2)-(4), and is verified when the degree is 10 for constraint (27)-(30). We did not obtain any positive verification result from solving constraint (17)-(19) with $\gamma = 1$ and $\gamma = 0:1$. However, this negative situation can be improved by solving constraint (36)-(41) with $\gamma = 1$ and $\gamma = 0:1$, and the reach-avoid property is verified when the degree is 10. If constraint (42)-(45) is used with $\psi(x) = (x + y)^2v(x)$, the degree is 6. Furthermore, if constraint (42)-(45) is used with $\psi(x) = x^4v(x)$, the degree is 4.

Analogously, we also experimented using constraint (17)-(19) and (36)-(41) with $\gamma = 10^{-2}; 10^{-3}; 10^{-4}; 10^{-5}; 10^{-6}$ for more comprehensive and fair comparisons. The computations terminate when the degree takes 8 for all of these experiments.

Fig. 7. Green, black and orange curve - \mathcal{X}_r , \mathcal{X} and \mathcal{X}_0 ; blue, red, cyan and purple curve - \mathcal{R} , which is computed via solving constraints (27)-(30) when the degree is 10, (36)-(41) when the degree is 10 and $\gamma = 1$, (42)-(45) when the degree is 6 and $\psi(x) = (x + y)^2v(x)$, and (42)-(45) when the degree is 4 and $\psi(x) = x^4v(x)$.

Example 9 (Dubin's Car): Consider the Dubin's car, a multi-input system: $\dot{a} = v \cos(\theta)$, $\dot{b} = v \sin(\theta)$, $\dot{\theta} = \omega$. By the change of variables, $x = a$, $y = a \cos(\theta) + b \sin(\theta)$, $z = 2(a \sin(\theta) - b \cos(\theta)) + y$, and $u_1 = \omega$, $u_2 = v$ ($a \sin(\theta) - b \cos(\theta)$), it is transformed into polynomial dynamics:

$$\begin{cases} \dot{x} = u_1; \\ \dot{y} = u_2; \\ \dot{z} = yu_1 - xu_2; \end{cases} \quad (46)$$

with $u_1 = 2$; $u_2 = 1 + z - xy$, $X = \{x \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 - 4 < 0\}$, $X_0 = \{x \in \mathbb{R}^3 \mid (x + 0:6)^2 + y^2 + (z + 0:6)^2 - 0:02 < 0\}$ and $X_r = \{x \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 - 4 < 0; (x - 1:0)^2 + (y + 0:5)^2 + (z + 0:1)^2 - 0:1 < 0\}$.

The reach-avoid property is verified when degree is 8 for all of constraints (2)-(4), (27)-(30), (17)-(19) with $\gamma = 1; 0:1; \dots; 10^{-6}$, (36)-(41) with $\gamma = 1; 0:1; \dots; 10^{-6}$. However, if constraint (42)-(45) is used with $\psi(x) = (1 - x)^2v(x)$, the degree is 6.

Examples above, i.e., Example 5-9, indicate that when the discount factor is small, constraint (17)-(19) has the same performance with constraint (27)-(30), although it performs worse when the discount factor is large. On the other hand, constraint (36)-(41) indeed is able to improve the performance of constraint (17)-(19) when the discount factor is large, but it does not improve constraint (27)-(30) and its

performance will be the same with constraint (17)-(19) when the discount factor is small. However, constraint (42)-(45) outperforms the former three, i.e., constraints (17)-(19), (27)-(30) and (36)-(41), and constraint (2)-(4). It is indeed more expressive and has more feasible solutions, providing more chances for verifying the reach-avoid property in the sense of Definition 1 successfully. Besides, from Example 8 we observe that the performance of constraint (42)-(45) is affected by the choice of the function $\alpha(x)$, and an appropriate choice will be more conducive to the reach-avoid verification. However, how to determine such a function in an optimal sense is still an open problem, which will be investigated in the future work. In practice, engineering experiences and insights may facilitate the choice.

In the present work we only demonstrate the performance of all quantified constraints by relaxing them into semi-definite constraints and addressing them within the semi-definite programming framework, which could be solved efficiently via interior point methods in polynomial time. It is worth remarking here that besides semi-definite programs for implementing these constraints, other methods such as counterexample-guided inductive synthesis methods combining machine learning and SMT solving techniques (e.g., [1]) can also be used to solve these constraints. We did not show their performance in this present work and leave these investigations for ones of interest.

V. CONCLUSION AND FUTURE WORK

In this paper we studied the reach-avoid verification problem of continuous-time systems within the framework of optimization based methods. At the beginning of our method, two sets of quantified inequalities were derived respectively based on a discount value function with the discount factor being larger than zero and equal to zero, such that the reach-avoid verification problem is transformed into a problem of searching for exponential/asymptotic guidance-barrier functions. The set of constraints associated with asymptotic guidance-barrier functions is completely novel and has certain benefits over the other one associated with exponential guidance-barrier functions, which is a simplified version of the one in existing literature. Furthermore, we enhanced the new set of constraints such that it is more expressive than the aforementioned two sets of constraints, providing more chances to verify the satisfaction of reach-avoid properties successfully. When the datum involved are polynomials, i.e., the initial set, safe set and target set are semi-algebraic, and the system has polynomial dynamics, the problem of solving these sets of constraints can be efficiently addressed using convex optimization. Finally, several examples demonstrated the theoretical developments and benefits of the proposed approaches.

Like the extension of barrier functions for safety verification to control barrier functions for synthesizing safe controllers [5], in the future work we will extend exponential/asymptotic guidance-barrier functions to control exponential/asymptotic guidance-barrier functions for synthesizing provably reach-avoid controllers.

REFERENCES

- [1] A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, and A. Peruffo. Fossil: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks. *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control* pages 1–11, 2021.
- [2] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. *Proceedings of the 16th international conference on Hybrid systems: computation and control* pages 173–182, 2013.
- [3] M. Althoff, G. Frehse, and A. Girard. Set propagation techniques for reachability analysis. *Annual Review of Control, Robotics, and Autonomous Systems* 4(1), 2021.
- [4] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and application. *2019 18th European control conference (ECC)* pages 3420–3431. IEEE, 2019.
- [5] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control* 62(8):3861–3876, 2016.
- [6] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. *HSCC'00* pages 20–31. Springer, 2000.
- [7] J.-P. Aubin. Viability kernels and capture basins of sets under differential inclusions. *SIAM Journal on Control and Optimization* 40(3):853–881, 2001.
- [8] S. Bak. t-barrier certificates: a continuous analogy to k-inductivity. *IFAC-PapersOnLine* 51(16):145–150, 2018.
- [9] M. Berz and K. Makino. Verified integration of odes and flows using differential algebraic methods on high-order taylor models. *Reliable computing* 4(4):361–369, 1998.
- [10] O. Bokanowski, N. Forcadel, and H. Zidani. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM Journal on Control and Optimization* 48(7):4292–4316, 2010.
- [11] T. Bräunl. *Embedded Robotics: Mobile Robot Design and Applications with Embedded systems*. Springer Science & Business Media, 2008.
- [12] X. Chen, E. Abraham, and S. Sankaranarayanan. Taylor model owpipe construction for non-linear hybrid systems. *2012 IEEE 33rd Real-Time Systems Symposium* pages 183–192. IEEE, 2012.
- [13] L. Dai, T. Gan, B. Xia, and N. Zhan. Barrier certificates revisited. *Journal of Symbolic Computation* 60:62–86, 2017.
- [14] A. Eggers, N. Ramdani, N. S. Nedialkov, and M. Fränzle. Improving the sat modulo ode approach to hybrid systems analysis by combining different enclosure methods. *Software & Systems Modeling* 14(1):121–148, 2015.
- [15] 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.
- [16] J. F. Fisac and S. S. Sastry. The pursuit-evasion-defense differential game in dynamic constrained environments. *CDC'15*, pages 4549–4556. IEEE, 2015.
- [17] M. Fränzle, M. Chen, and P. König. In memory of oded maler: automatic reachability analysis of hybrid-state automata. *ACM SIGLOG News* 6(1):19–39, 2019.
- [18] T. Gan, M. Chen, Y. Li, B. Xia, and N. Zhan. Reachability analysis for solvable dynamical systems. *IEEE Trans. Automat. Contr.* 63(7):2003–2018, 2018.
- [19] J. H. Gillula and C. J. Tomlin. Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. *ICRA'12*, pages 2723–2730. IEEE, 2012.
- [20] D. Henrion and M. Korda. Convex computation of the region of attraction of polynomial control systems. *IEEE Trans. Automat. Contr.* 59(2):297–312, 2013.
- [21] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *J. Comput. Syst. Sci.* 57(1):94–124, 1998.
- [22] H. K. Khalil. *Nonlinear systems third edition*. Prentice Hall, 115, 2002.
- [23] H. Kong, F. He, X. Song, W. N. Hung, and M. Gu. Exponential-condition-based barrier certificate generation for safety verification of hybrid systems. In *International Conference on Computer Aided Verification*, pages 242–257. Springer, 2013.
- [24] M. Korda, D. Henrion, and C. N. Jones. Inner approximations of the region of attraction for polynomial dynamical systems. *IFAC Proceedings Volumes* 46(23):534–539, 2013.
- [25] M. Korda, D. Henrion, and C. N. Jones. Controller design and region of attraction estimation for nonlinear dynamical systems. *IFAC Proceedings Volumes* 47(3):2310–2316, 2014.
- [26] U. Kulisch, R. Lohner, and A. Facius. *Perspectives on Enclosure Methods*. Springer Science & Business Media, 2001.
- [27] E. A. Lee. *Cyber physical systems: Design challenges*. Springer, 2018.
- [28] J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *CACSD'04* pages 284–289. IEEE, 2004.
- [29] A. Majumdar, R. Vasudevan, M. M. Tobenkin, and R. Tedrake. Convex optimization of nonlinear feedback controllers via occupation measures. *The International Journal of Robotics Research* 33(9):1209–1230, 2014.
- [30] K. Margellos and J. Lygeros. Hamilton-jacobi formulation for reach-avoid differential games. *IEEE Trans. Automat. Contr.* 56(8):1849–1861, 2011.

- [31] 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.
- [32] I. Mitchell and C. J. Tomlin. Level set methods for computation in hybrid systems. In *HSCC'00*, pages 310–323. Springer, 2000.
- [33] A. Mosek. The mosek optimization toolbox for matlab manual, 2015.
- [34] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *HSCC'04*, pages 477–492. Springer, 2004.
- [35] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, 2007.
- [36] S. Prajna and A. Rantzer. Convex programs for temporal verification of nonlinear dynamical systems. *SIAM Journal on Control and Optimization*, 46(3):999–1021, 2007.
- [37] N. Ramdani, N. Meslem, and Y. Candau. A hybrid bounding method for computing an over-approximation for the reachable set of uncertain nonlinear systems. *IEEE Transactions on automatic control*, 54(10):2352–2364, 2009.
- [38] V. Shia, R. Vasudevan, R. Bajcsy, and R. Tedrake. Convex computation of the reachable set for controlled polynomial hybrid systems. In *CDC'14*, pages 1499–1506. IEEE, 2014.
- [39] A. Sogokon, K. Ghorbal, Y. K. Tan, and A. Platzer. Vector barrier certificates and comparison systems. In *International Symposium on Formal Methods*, pages 418–437. Springer, 2018.
- [40] W. Tan and A. Packard. Stability region analysis using polynomial and composite polynomial lyapunov functions and sum-of-squares programming. *IEEE Trans. Automat. Contr.*, 53(2):565–571, 2008.
- [41] B. C. Williams, M. D. Ingham, S. H. Chung, and P. H. Elliott. Model-based programming of intelligent embedded systems and robotic space explorers. *Proceedings of the IEEE*, 91(1):212–237, 2003.
- [42] B. Xue, A. Easwaran, N.-J. Cho, and M. Fränzle. Reach-avoid verification for nonlinear systems based on boundary analysis. *IEEE Trans. Automat. Contr.*, 62(7):3518–3523, 2016.
- [43] B. Xue, M. Fränzle, and N. Zhan. Inner-approximating reachable sets for polynomial systems with time-varying uncertainties. *IEEE Trans. Automat. Contr.*, 65(4):1468–1483, 2020.
- [44] B. Xue, N. Zhan, and M. Fränzle. Inner-approximating reach-avoid sets for discrete-time polynomial systems. In *CDC'20*, pages 867–873. IEEE, 2020.
- [45] P. Zhao, S. Mohan, and R. Vasudevan. Control synthesis for nonlinear optimal control via convex relaxations. In *ACC'17*, pages 2654–2661. IEEE, 2017.

VI. APPENDIX

The semi-definite program for solving constraints (2)-(4):

$$\begin{aligned} v(\mathbf{x}) + \sum_{i=1}^{k_1} s_{0,i}(\mathbf{x}) l_i(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} v(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + s_{1,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{2,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ v(\mathbf{x}) \quad \rho(\mathbf{x}) h(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ j &= 1; \dots; j_1; \end{aligned}$$

where $v(\mathbf{x}); \rho(\mathbf{x}) \geq R[\mathbf{x}]$, and $s_{0,i}(\mathbf{x}) \geq \epsilon^P [\mathbf{x}]$, $i = 0; \dots; k_1$, $s_{1,j}; s_{2,j} \geq \epsilon [\mathbf{x}]$, $j = 1; \dots; j_1$, and $\epsilon > 0$ is a user-defined value enforcing the strict positiveness of $v(\mathbf{x})$ and $5_{\times} v(\mathbf{x}) \quad \mathbf{f}(\mathbf{x})$. In the computations of this paper we take $\epsilon = 10^{-6}$.

The semi-definite program for solving constraints (17)-(19):

$$\begin{aligned} v(\mathbf{x}) + \sum_{i=1}^{k_1} s_{0,i}(\mathbf{x}) l_i(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} v(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + s_{1,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{2,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ v(\mathbf{x}) \quad \rho(\mathbf{x}) h(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ j &= 1; \dots; j_1; \end{aligned}$$

where $v(\mathbf{x}); \rho(\mathbf{x}) \geq R[\mathbf{x}]$, and $s_{0,i}(\mathbf{x}) \geq \epsilon^P [\mathbf{x}]$, $i = 0; \dots; k_1$, $s_{1,j}; s_{2,j} \geq \epsilon [\mathbf{x}]$, $j = 1; \dots; j_1$, and $\epsilon > 0$ is a user-defined value enforcing the strict positiveness of $v(\mathbf{x})$. In the computations of this paper we take $\epsilon = 10^{-6}$.

The semi-definite program for solving constraints (27)-(30):

$$\begin{aligned} v(\mathbf{x}) + \sum_{i=1}^{k_1} s_{0,i}(\mathbf{x}) l_i(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} v(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + s_{1,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{2,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} w(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + s_{3,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{4,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ v(\mathbf{x}) \quad \rho(\mathbf{x}) h(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ j &= 1; \dots; j_1; \end{aligned}$$

where $v(\mathbf{x}); w(\mathbf{x}); \rho(\mathbf{x}) \geq R[\mathbf{x}]$, and $s_{0,i}(\mathbf{x}) \geq \epsilon^P [\mathbf{x}]$, $i = 1; \dots; k_1$, and $s_{i,j} \geq \epsilon [\mathbf{x}]$, $i = 1; \dots; 4; j = 1; \dots; j_1$, $\epsilon > 0$ is a user-defined value enforcing the strict positiveness of $v(\mathbf{x})$ over the initial set X_0 . In the computations of this paper we take $\epsilon = 10^{-6}$.

The semi-definite program for solving constraints (36)-(41):

$$\begin{aligned} v_1(\mathbf{x}) + v_2(\mathbf{x}) + \sum_{i=1}^{k_1} s_{0,i}(\mathbf{x}) l_i(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} v_1(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + s_{1,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{2,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} w(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + v_1(\mathbf{x}) + s_{3,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{4,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} v_2(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + v_2(\mathbf{x}) + s_{5,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{6,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ v_1(\mathbf{x}) \quad \rho(\mathbf{x}) h(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ v_2(\mathbf{x}) \quad \rho(\mathbf{x}) h(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ j &= 1; \dots; j_1; \end{aligned}$$

where $v(\mathbf{x}); w(\mathbf{x}); \rho(\mathbf{x}) \geq R[\mathbf{x}]$, and $s_{0,i}(\mathbf{x}) \geq \epsilon^P [\mathbf{x}]$, $i = 1; \dots; k_1$, and $s_{i,j} \geq \epsilon [\mathbf{x}]$, $i = 1; \dots; 6; j = 1; \dots; j_1$, $\epsilon > 0$ is a user-defined value enforcing the strict positiveness of $v(\mathbf{x})$ over the initial set X_0 . In the computations of this paper we take $\epsilon = 10^{-6}$.

The semi-definite program for solving constraints (42)-(45):

$$\begin{aligned} v(\mathbf{x}) + \sum_{i=1}^{k_1} s_{0,i}(\mathbf{x}) l_i(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} v(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + s_{1,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{2,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ 5_{\times} w(\mathbf{x}) \quad \mathbf{f}(\mathbf{x}) + v(\mathbf{x}) + s_{3,j}(\mathbf{x}) h(\mathbf{x}) \quad s_{4,j}(\mathbf{x}) g_j(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ v(\mathbf{x}) \quad \rho(\mathbf{x}) h(\mathbf{x}) &\geq \epsilon \quad [\mathbf{x}]; \\ j &= 1; \dots; j_1; \end{aligned}$$

where $v(\mathbf{x}); w(\mathbf{x}); \rho(\mathbf{x}) \geq R[\mathbf{x}]$, and $s_{0,i}(\mathbf{x}) \geq \epsilon^P [\mathbf{x}]$, $i = 1; \dots; k_1$, and $s_{i,j} \geq \epsilon [\mathbf{x}]$, $i = 1; \dots; 4; j = 1; \dots; j_1$, $\epsilon > 0$ is a user-defined value enforcing the strict positiveness of $v(\mathbf{x})$ over the initial set X_0 . In the computations of this paper we take $\epsilon = 10^{-6}$.

