Reach-Avoid Verification Based on Convex Optimization

Bai Xue ©, Member, IEEE, Naijun Zhan ©, Martin Fränzle ©, Ji Wang ©, and Wanwei Liu ©

Abstract—In this article, we propose novel sufficient conditions for verifying reach-avoid properties of continuous-time systems modeled 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 (that is, if the system starting from the initial set can reach the target set safely). 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 these new sets 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 property successfully. Finally, several examples demonstrate the theoretical developments and performance of proposed sufficient conditions using semidefinite programming methods.

Index Terms—Ordinary differential equations (ODEs), quantified constraints, reach-avoid verification.

I. INTRODUCTION

CYBER-PHYSICAL technology is integrated into an ever-growing range of physical devices and increasingly pervades our daily life [17]. 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. [8], [25]. Many of the aforementioned applications are safety-critical and require a rigorous guarantee of safe operation.

Among the many possible rigorous guarantees, reach-avoid verification, i.e., verifying whether the system’s dynamics (generally modeled by ODEs) satisfy reach-avoid properties, is definitely in demand. One of the popular methods for reach-avoid verification is computational reachability analysis, which involves the explicit computation of reachable states [3], [12]. In general, the exact computation of reach sets is impossible for dynamical and hybrid systems [13]. Overapproximate reachability analysis, which computes an overapproximation (i.e., superset) of the reach set based on set propagation techniques, is therefore studied in the existing literature for verification purposes (e.g., [5]). Overly pessimistic overapproximations, 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, which is the propagation and accumulation of overapproximation errors through the iterative computation of reach sets. There are many techniques developed in the existing literature for controlling the wrapping effect. One way is to use complex sets such as Taylor models [7], [10] and polynomial zonotopes [2] to overapproximate 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 overapproximate reachability analysis via exploiting the (topological) structure of the system. For instance, appropriate corner points of reach sets, called bracketing systems, are used in [11] and [23] to bound the complete reach sets when the systems under consideration are monotonic; the authors in [26] proposed the set-boundary reachability method for continuous-time systems featuring a locally Lipschitz-continuous vector field.

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 [20] and [21] for safety verification of continuous and hybrid systems. The barrier certificate method was inspired by Lyapunov functions in the 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 [6], [14], 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 reach-avoid verification in [22]. The set of constraints in [21] requires the Lie derivative of barrier certificates to be strictly decreasing along the trajectories of the dynamics. It is strong, limiting its applications as discussed in Section II-B. Recently, a set of new constraints based on the moment theory was presented in [15] for inner approximating the set of all initial states guaranteeing the satisfaction of the reach-avoid property.
It can be straightforwardly extended to reach-avoid verification via supplementing a constraint that the designated initial set is included in the computed inner approximation. The obtained set of constraints overcomes the strong requirement of the one in [21], but it is a special case of the proposed ones in this article.

In this article, we study the reach-avoid verification problem of continuous-time systems modeled by ODEs 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. 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 in Definition 1. We attempt to solve this problem within the framework of optimization-based methods. Generally, such methods are sound but incomplete.

In the following computations, all of constraints for reach-avoid verification are addressed via encoding them into semidefinite programs. The formulated semidefinite programs can be found in https://arxiv.org/pdf/2208.08105. In addition, all of semidefinite programs are formulated using MATLAB package YALMIP [18] and solved by employing the academic version of the semidefinite programming solver MOSEK [19].

II. PRELIMINARIES

In this section, we formally present the concepts of continuous-time systems and reach-avoid verification problem of interest in this article. Before formulating them, let us introduce some basic notions used throughout this article: for a function \( v(x) \), \( \nabla_x v(x) \) denotes its gradient with respect to \( x \); \( \mathbb{R}^{\geq 0} \) 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 \( \partial X \); \( \land \) denotes conjunction, \( \lor \) and \( \exists \) denote the universal and existential quantifiers, respectively; the ring of all multivariate polynomials in a variable \( x \) is denoted by \( \mathbb{R}[x] \); vectors are denoted by boldface letters, and the transpose of a vector \( x \) is denoted by \( \mathbf{x}^\top \).}

\[ \sum \mathbf{x} = \{ p \in \mathbb{R}[\mathbf{x}] | p = \sum_{i=1}^{k} q_i^2, q_i \in \mathbb{R}[\mathbf{x}], i = 1, \ldots, k \}. \]

A. Preliminaries

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

\[ \dot{x} = f(x), x(0) = x_0 \in \mathbb{R}^n \]

where \( \dot{x} = \frac{dx(t)}{dt} \) and \( f(x) = \{ f_1(x), \ldots, f_n(x) \}^\top \) with \( f_i(x) \in \mathbb{R}_p \).

We denote the trajectory of the system CS that originates from \( x_0 \) and is defined over the maximal time interval \( [0, T_{\text{max}}] \) by \( \phi_{\mathbb{R}_p}(\cdot) : [0, T_{\text{max}}) \to \mathbb{R}^n \). Consequently, we have

\[ \phi_{\mathbb{R}_p}(t) := x(t), \forall t \in [0, T_{\text{max}}), \quad \text{and} \quad \phi_{\mathbb{R}_p}(0) = x_0 \]

where \( T_{\text{max}} \) is either a positive value (i.e., \( T_{\text{max}} \in \mathbb{R}^+ \) or \( \infty \)).

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

\[ X = \{ x \in \mathbb{R}^n | h(x) < 0 \} \quad \text{with} \quad \partial X = \{ x \in \mathbb{R}^n | h(x) = 0 \} \]

\[ X_0 = \{ x \in \mathbb{R}^n | l(x) < 0 \} \quad \text{and} \quad X_r = \{ x \in \mathbb{R}^n | g(x) < 0 \} \]

with \( l(x), h(x), g(x) \in \mathbb{R}[x] \), 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 CS 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 \( x_0 \in X_0 \), any trajectory \( \phi_{\mathbb{R}_p}(t) \) of system CS starting at \( \phi_{\mathbb{R}_p}(0) = x_0 \) can eventually enter the target set \( X_r \) eventually while remaining inside the safe set until the first target hit, i.e.,

\[ \forall x_0 \in X_0, \exists T \in \mathbb{R}^{>0}, \exists \phi_{\mathbb{R}_p}(T) \in X_r \] \( \forall t \in [0, T] \), \( \phi_{\mathbb{R}_p}(t) \in X \).

Since the reach-avoid property combines guarantees of safety by staying within the safe set \( X \) with the reachability property of reaching the target set \( X_r \) and thus can formalize many important engineering problems such as autonomous spacecraft rendezvous [9], its verification has turned out to be of fundamental importance in engineering. The problem of interest in this work is on reach-avoid verification, i.e., verifying that system CS 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.

In the following computations, all of constraints for reach-avoid verification are addressed via encoding them into semidefinite programs. The formulated semidefinite programs can be found in https://arxiv.org/pdf/2208.08105. In addition, all of semidefinite programs are formulated using MATLAB package YALMIP [18] and solved by employing the academic version of the semidefinite programming solver MOSEK [19].

B. Existing Methods

For the convenience of comparisons, in this subsection, we recall the sets of quantified constraints in existing literature for verifying the reach-avoid property in Definition 1. The first one is from [22], while the other one is from [15].

Proposition 1 (see [22]): Suppose that there exists a continuously differentiable function \( v(x) : \overline{X} \to \mathbb{R} \) satisfying

\[ v(x) \leq 0 \forall x \in X_0 \] \( (2) \)

\[ v(x) > 0 \forall x \in \partial X \setminus \partial X_r \] \( (3) \)

\[ \nabla_x v(x) : f(x) < 0 \forall x \in \overline{X} \setminus X_r. \] \( (4) \)

Then, the reach-avoid property in Definition 1 holds.

One of drawbacks of constraints (2)–(4) in reach-avoid verification is the strong requirement that the Lie derivative of \( v(x) \) should be strictly decreasing along the trajectories of the system CS over the set \( \overline{X} \setminus X_r \). A straightforward consequence is that these constraints cannot deal with
the case with an equilibrium being inside $\mathcal{X} \setminus \mathcal{X}_r$, since $f(x_0) = 0$ implies $\nabla_x v(x) \cdot f(x) \mid_{x=x_0} = 0$.

Besides, if the reach-avoid property in Definition 1 holds, the initial set $\mathcal{X}_0$ must be a subset of the reach-avoid set $\mathcal{R}A$, which is the set of all initial states guaranteeing the satisfaction of the reach-avoid property, i.e.,

$$\mathcal{R}A = \left\{ x_0 \in \mathbb{R}^n \mid \exists t \in [0, T], \phi_{\mathcal{R}A}(t) \in \mathcal{X}_r \wedge \forall t \in [0, T], \phi_{\mathcal{R}A}(t) \in \mathcal{X}_r \right\}.$$ 

The method for computing underapproximations of the reach-avoid set $\mathcal{R}A$ can be used for reach-avoid verification. By adding the condition $v(x) < 0 \forall x \in \mathcal{X}_0$ into constraint [15, eq. (18)], which is originally developed for underapproximating the reach-avoid set $\mathcal{R}A$, we can obtain a set of quantified constraints as shown in Proposition 2 for reach-avoid verification.

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

$$v(x) < 0 \forall x \in \mathcal{X}_0$$
$$\nabla_x v(x) \cdot f(x) \leq \beta v(x) \forall x \in \mathcal{X} \setminus \mathcal{X}_r$$

and $w(x)$ is defined as follows:

$$w(x) = 0$$
$$w(x) \geq v(x) + 1 \forall x \in \mathcal{X} \setminus \mathcal{X}_r$$

then trajectories entering the set $\mathcal{X}_r$ in finite time.

## III. REACH-AVOID VERIFICATION

This section presents our optimization-based methods for reach-avoid verification. Based on a discount value function, which is defined based on trajectories of a switched system and introduced in Section 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 reach-avoid verification.

### A. Induced Switched Systems

This subsection introduces a switched system, which is built upon system CS. This switched system is constructed by requiring the state of the system CS to stay still when the complement of the safe set $\mathcal{X}_r$ is reached. For the sake of brevity, only trajectories of the induced switched system, also called CTSS, are introduced.

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

$$\dot{x}(t) := 1_{\chi}(x(t)) \cdot f(x(t))$$

with $1_{\chi}(\cdot) : \mathcal{X} \rightarrow \{0, 1\}$ representing the indicator function of the set $\chi$, i.e.,

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

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

$$\widehat{\phi}_{x_0}(t) := x(t) \forall t \in \mathbb{R}_{\geq 0},$$

It is observed that the set $\mathcal{X}$ is an invariant set for the system CTSS. Also, if $x_0 \in \mathcal{X}$ and there exists $T \geq \mathbb{R}_{\geq 0}$ such that $\widehat{\phi}_{x_0}(t) \in \mathcal{X}_r$ for $t \in [0, T]$, we have $\widehat{\phi}_{x_0}(t) = \phi_{\mathcal{R}A}(t) \forall t \in [0, T]$. Also, trajectories of system CTSS evolving in the viable set $\mathcal{X}$ can be classified into three disjoint groups.

1) Trajectories entering the set $\mathcal{X}_r$ in finite time. It is worth remarking here that these trajectories will not leave the safe set $\mathcal{X}_r$ before reaching the target set $\mathcal{X}_r$. Since $\phi_{\mathcal{R}A}(t) = \phi_{\mathcal{R}A}(t) \forall t \in [0, T]$, where $x_0 \in \mathcal{X}$ and $T \in \mathbb{R}_{\geq 0}$ is a time instant such that $\widehat{\phi}_{x_0}(t) \in \mathcal{X}_r$ for $t \in [0, T]$, we conclude that the set of initial states deriving these trajectories equals the reach-avoid set $\mathcal{R}A$.

2) Trajectories entering the set $\partial \mathcal{X}$ in finite time, but never entering the target set $\mathcal{X}_r$.

3) Trajectories staying in the set $\mathcal{X} \setminus \mathcal{X}_r$ for all time.

### B. Discount Value Functions

The discount value function, mentioned above, is introduced in this subsection. With a nonnegative discount factor $\beta$, the discount value function $V(x) : \mathcal{X} \rightarrow \mathbb{R}$ with a nonnegative discount factor $\beta$ is defined in the following form:

$$V(x) := \sup_{t \in \mathbb{R}_{\geq 0}} e^{-\beta t} 1_{\chi_r}(\widehat{\phi}_{x}(t))$$

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

$$V(x) = \begin{cases} 1, & \text{if } x \in \mathcal{R}A \\ 0, & \text{otherwise} \end{cases}$$

If $\beta > 0$, $V(x) = \begin{cases} 0, & \text{if } x \in \mathcal{X} \setminus \mathcal{R}A \\ e^{-\beta \tau_{x}}, & \text{if } x \in \mathcal{R}A \end{cases}$

where $\tau_x = \inf \{ t \in \mathbb{R}_{\geq 0} \mid \widehat{\phi}_{x}(t) \in \mathcal{X}_r \}$ is the first hitting time of the target set $\mathcal{X}_r$.

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

**Proposition 5:** When $\beta = 0$, the one level set of $V(\cdot) : \mathcal{X} \rightarrow \mathbb{R}$ in (11) equals the reach-avoid set $\mathcal{R}A$, i.e., $\{ x \in \mathcal{X} \mid V(x) = 1 \} = \mathcal{R}A$. When $\beta > 0$, the strict zero super level set of $V(\cdot) : \mathcal{X} \rightarrow \mathbb{R}$ in (11) equals $\mathcal{R}A$, i.e., $\{ x \in \mathcal{X} \mid V(x) > 0 \} = \mathcal{R}A$.

In the following, we, respectively, obtain two sets of quantified constraints for reach-avoid verification based on the discount value function $V(x)$ in (11) with $\beta > 0$ and $\beta = 0$. Through thorough analysis on these two sets of constraints, we further obtain a set of more expressive constraints for 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 $\beta > 0$, 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 the system CTSS, if there exists a continuously differential function $v(x) : \mathcal{X} \rightarrow \{0, 1\}$ such that

$$\nabla_x v(x) \cdot f(x) = \beta v(x) \forall x \in \mathcal{X} \setminus \mathcal{X}_r$$

and

$$v(x) = 0 \forall x \in \partial \mathcal{X}_r$$

and

$$v(x) = 1 \forall x \in \mathcal{X}_r$$

then $x \in \mathcal{R}A$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
then $v(x) = V(x)$ for $x \in \mathcal{X}$, and thus, $\{x \in \mathcal{X} \mid v(x) > 0\} = \mathcal{R} \mathcal{A}$, where $V(\cdot) : \mathcal{X} \to \mathbb{R}$ is the value function with $\beta > 0$ in (11).

**Proof.** We first consider that $x \in \mathcal{R} \mathcal{A}$. If $x \in \mathcal{X}_r$, $v(x) = 1 = e^0$ according to constraint (16). Thus, we just consider $x \in \mathcal{R} \mathcal{A} \setminus \mathcal{X}_r$.

From (14), we have that $v(x) = e^{-\beta x} v(\phi_0(x)) \forall t \in [0, \tau_x]$, where $\tau_x$ is the first hitting time of the target set $\mathcal{X}_r$. Due to constraint (16), we further have that $v(x) = e^{-\beta x}$.

Next, we consider that $x \in \mathcal{X} \setminus \mathcal{R} \mathcal{A}$, but its resulting trajectory $\hat{\phi}_0(\tau)$ will stay within the set $\mathcal{X}' \setminus \mathcal{X}_r$ for all time. Due to constraint (14), we have that $v(x) = e^{-\beta x} v(\hat{\phi}_0(\tau))$ for $\tau \in \mathbb{R}_{\geq 0}$. Since $v(\cdot) : \mathcal{X} \to \mathbb{R}$ is bounded, we have $v(x) = 0$.

Finally, we consider that $x \in \mathcal{X} \setminus \mathcal{R} \mathcal{A}$, but its resulting trajectory $\hat{\phi}_0(\tau)$ will touch the set $\partial \mathcal{X}'$ in finite time and never enter the target set $\mathcal{X}_r$. Let $\tau'_x = \inf\{t \in \mathbb{R}_{\geq 0} \mid \hat{\phi}_0(t) \in \partial \mathcal{X}'\}$ be the first hitting time of the set $\partial \mathcal{X}'$.

If $x \in \partial \mathcal{X}'$, $v(x) = 0$ holds from constraint (15). Otherwise, $\tau'_x > 0$. Further, from (14), we have $v(x) = e^{-\beta x} v(\hat{\phi}_0(\tau'_x)) \forall t \in [0, \tau'_x]$. Regarding constraint (15), which implies $v(\phi_0(\tau'_x)) = 0$, we have $v(x) = 0$.

Thus, $v(x) = V(x)$ for $x \in \mathcal{X}$ according to (13) and $\mathcal{R} \mathcal{A} = \{x \in \mathcal{X} \mid v(x) > 0\}$.

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 (14) and (15), respectively.

**Proposition 4:** If there exists a continuously differentiable function $v(x) : \mathcal{X} \to \mathbb{R}$ such that

$$v(x) > 0 \quad \forall x \in \mathcal{X}_0$$

$$\nabla_x v(x) \cdot f(x) \geq \beta v(x) \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_r \tag{17}$$

$$v(x) \leq 0 \quad \forall x \in \partial \mathcal{X}' \tag{19}$$

where $\beta > 0$ 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, the constraint (17)–(19) will be used for discussions and comparisons instead of the constraint (5)–(9) in the sequel.

If an exponential guidance-barrier function $v(x)$ satisfying the 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 $\mathcal{R} = \{x \in \mathcal{X} \mid v(x) > 0\}$ is an underapproximation of the reach-avoid set, i.e., $\mathcal{R} \subseteq \mathcal{R} \mathcal{A}$. Also, due to constraint (18), we conclude that the set $\mathcal{R}$ is an invariant for system $\mathcal{C}$ until it enters the target set $\mathcal{X}_r$, i.e., the boundary $\partial \mathcal{R} = \{x \in \mathcal{X} \mid v(x) = 0\}$ is a barrier, preventing system $\mathcal{C}$ from leaving the set $\mathcal{R}$ and escaping system $\mathcal{C}$ to the target set $\mathcal{X}_r$ safely; furthermore, we observe that if $v(x)$ satisfies constraint (17)–(19), it must satisfy

$$\nabla_x v(x) \cdot f(x) \geq \beta v(x) \quad \forall x \in \mathcal{R} \setminus \mathcal{X}_r \tag{20}$$

which implies $v(\phi_0(t)) \geq e^{\beta t} v(x_0) \quad \forall t \in [0, \tau_{x_0}], \quad x_0 \in R \setminus \mathcal{X}_r$ and $\tau_{x_0} = \inf\{t \in \mathbb{R}_{\geq 0} \mid \phi_0(t) \in \mathcal{X}_r\}$ is the first hitting time of the target set $\mathcal{X}_r$. Consequently, this constraint indicates that trajectories starting from $\mathcal{R} \setminus \mathcal{X}_r$ will approach the target set $\mathcal{X}_r$ at an exponential rate of $\beta$. This is why we term a solution to the set of constraints (17)–(19) exponential guidance-barrier function.

The aforementioned analysis also uncovers a necessary condition such that $v(x)$ is a solution to the constraint (17)–(19), which is $\mathcal{R} \cap \mathcal{X}_r \neq \emptyset$.

The condition of entering the target set at an exponential rate is strict for many cases, limiting the application of the constraint (17)–(19) to reach-avoid verification in practice. On the other hand, since the initial set $\mathcal{X}_0$ should be a subset of the set $\mathcal{R}$, the less conservative the set $\mathcal{R}$ is, the more likely the reach-avoid property is able to be verified. It is concluded from constraint (20) that the smaller $\beta$ is, the less conservative the set $\mathcal{R}$ is inclined to be. This is illustrated in the following example.

**Example 1:** Consider an academic example

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

with $\mathcal{X} = \{(x, y) \mid x^2 + y^2 - 1 < 0\}$, $\mathcal{X}_r = \{(x, y) \mid (x + 0.2)^2 + (y - 0.7)^2 - 0.02 < 0\}$, and $\mathcal{X}_0 = \{(x, y) \mid 0.1 - x < 0, x - 0.5 < 0, -0.8 - y < 0, y + 0.5 < 0\}$.

In this example, we use $\beta = 0.1$ and $\beta = 1$ to illustrate the effect of $\beta$ on reach-avoid verification via solving the constraint (17)–(19). The degree of all polynomials in the resulting semidefinite program is taken the same and is taken in order of $\{2, 4, 6, 8, 10, \ldots, 20\}$. When the reach-avoid property is verified successfully, the computations will terminate. The degree is, respectively, 14 for $\beta = 0.1$ and 20 for $\beta = 1$ when termination. Both of the computed sets $\mathcal{R}$ are showcased in Fig. 1, which almost collide with each other.

It is worth emphasizing here that although the discount factor $\beta$ can arbitrarily approach zero from aforementioned, it cannot be zero in the constraint (17)–(19), since a function $v(x)$ satisfying this constraint with $\beta = 0$ cannot rule out the existence of trajectories, which start from $\mathcal{X}_0$ and stay inside $\mathcal{X} \setminus \mathcal{X}_r$ for ever. Consequently, we do not recommend the use of too small $\beta$ in practical numerical computations in order to avoid numerical issue (i.e., preventing the term $\beta v(x)$ in the right hand of the constraint (18) from becoming zero numerically due to floating point errors).

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

### D. Asymptotic Guidance-BARRIER Function

In this subsection, we elucidate the construction of constraints for reach-avoid verification based on the discount value function $V(x)$ in (11) with $\beta = 0$. 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 $\beta = 0$ as solutions. These equations are presented in Theorem 2.

**Theorem 2:** Given system CTSS, if there exist continuously differential functions $v(x): \mathcal{X} \to \mathbb{R}$ and $w(x): \mathcal{X} \to \mathbb{R}$ satisfying

\[
\nabla_x v(x) \cdot f(x) = 0 \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_c
\]

then $v(x) = V(x)$ for $x \in \mathcal{X}$, and thus, $(x \in \mathcal{X} | v(x) = 1) = \mathcal{R}A$, where $V(\cdot): \mathcal{X} \to \mathbb{R}$ is the value function with $\beta = 0$ in (11).

**Proof:** From (22), we have that

\[
v(x) = v(\hat{\phi}_x(t)) \quad \forall t \in [0, t_x]
\]

where $t_x \in [0, +\infty)$ is the time instant such that $\hat{\phi}_x(t) \in \mathcal{X} \setminus \mathcal{X}_c$, $\forall t \in [0, t_x]$.

For $x \in \mathcal{R}A$, we obtain $v(x) = 1$ due to (25) and (26).

In the following, we consider $x \in \mathcal{X} \setminus \mathcal{R}A$.

We first consider $x \in \mathcal{X} \setminus \mathcal{R}A$, but its trajectory $\hat{\phi}_x(\cdot) \in [0, +\infty)$ stays within the set $\mathcal{X} \setminus \mathcal{X}_c$. From (23), we have that

\[
v(\hat{\phi}_x(t)) = \nabla_y w(y) \cdot f(y) \big|_{y = \hat{\phi}_x(t)}
\]

for $t \geq 0$. Thus, we have that

\[
\int_0^t v(\hat{\phi}_x(t)) dt = \int_0^t \nabla_y w(y) \cdot f(y) \big|_{y = \hat{\phi}_x(t)} dt
\]

for $t \geq 0$ and further $v(x) = w(\hat{\phi}_x(t)) - w(x)$ for $t \geq 0$. Since $w(x)$ is continuously differential function over $\mathcal{X}$, it is bounded over $x \in \mathcal{X}$. Consequently, $v(x) = 0$.

Next, we consider $x \in \mathcal{X} \setminus \mathcal{R}A$, but its trajectory $\hat{\phi}_x(t)$ will touch $\partial \mathcal{X}$ in finite time and never enters the target set $\mathcal{X}_c$. For such $x$, we can obtain that $v(x) = 0$ due to constraints (24) and (26).

Thus, according to (12), $v(x) = V(x)$ for $x \in \mathcal{X}$. Further, from Lemma 3, $\{x \in \mathcal{X} | v(x) = 1\} = \mathcal{R}A$ holds.

Based on the system of equations (22)–(25), we have a set of inequalities as shown in Proposition 5 for computing an asymptotic guidance-barrier function $v(x)$ to ensure the satisfaction of reach-avoid properties in the sense of Definition 1. In Proposition 5, inequalities (28)–(30) are obtained directly by relaxing (22)–(24), respectively.

**Proposition 5:** If there exist a continuously differential function $v(x): \mathcal{X} \to \mathbb{R}$ and a continuously differential function $w(x): \mathcal{X} \to \mathbb{R}$ satisfying

\[
v(x) > 0 \quad \forall x \in \mathcal{X}_0
\]

\[
\nabla_x v(x) \cdot f(x) \geq 0\forall x \in \mathcal{X} \setminus \mathcal{X}_c
\]

\[
v(x) - \nabla_x w(x) \cdot f(x) \leq 0\forall x \in \mathcal{X} \setminus \mathcal{X}_c
\]

\[
v(x) \leq 0\forall x \in \partial \mathcal{X}
\]

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

**Proof:** Let $\mathcal{R} = \{x \in \mathcal{X} | v(x) > 0\}$. We will show that $\mathcal{R} \subseteq \mathcal{R}A$. If it holds, we can obtain the conclusion since $\mathcal{X}_0 \subseteq \mathcal{R}A$, which is obtained from the constraint (27).

Let $x_0 \in \mathcal{R}$. Obviously, $x_0 \in \mathcal{X}$ due to constraint (30). If $x_0 \in \mathcal{X}_c$, $x_0 \in \mathcal{R}A$ holds obviously. Therefore, in the following, we assume $x_0 \in \mathcal{X} \setminus \mathcal{X}_c$. We will prove that there exists $t \in \mathbb{R}^+$ satisfying $\phi_{x_0}(t) \in \mathcal{X} \setminus \mathcal{X}_c$. We are inclined to verify whether there exists a continuously differential function $w(x)$ satisfying

\[
v(x) \geq 0 \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_c
\]

\[
v(x) - \nabla_x w(x) \cdot f(x) \leq 0 \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_c
\]

\[
v(x) \leq 0 \quad \forall x \in \partial \mathcal{X}
\]

holds. Thus, constraint (33) is more expressive than (20), and consequently, is more likely to produce less conservative set $\mathcal{R}$. We in the following continue to use the scenario in Example 1 to illustrate this.

**Example 2:** Consider the scenario in Example 1 again. We solve constraint (27)–(30) to verify the reach-avoid property. The reach-avoid property is verified when polynomials of degree 12 in the resulting semidefinite program are taken.

**Remark 1:** As done in verifying invariance of a set using barrier certificate methods [4], one simple application scenario, reflecting the advantage of constraint (33) over (20) further, is on verifying whether trajectories starting from a given open set $\mathcal{R}$, which may be designed a priori via the Monte-Carlo simulation method, will enter the target set $\mathcal{X}_c$ eventually while staying inside it before the first hit time, where $\mathcal{R} = \{x \in \mathcal{X} | v(x) > 0\}$ with $v(x)$ being continuously differentiable and $\mathcal{X}_c \cap \mathcal{R} \neq \emptyset$. We are inclined to verify whether there exists a continuously differential function $w(x)$ satisfying

\[
v(x) \geq 0 \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_c
\]

\[
v(x) - \nabla_x w(x) \cdot f(x) \leq 0 \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_c
\]
instead of verifying whether there exists $\beta > 0$ such that 
\[
\nabla_v \hat{v}(x) \cdot f(x) \geq \beta \hat{v}(x) \quad \forall x \in \overline{R} \setminus \mathcal{X}_r
\]

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 reach-avoid verification, there is still a defect caused by constraint (28), possibly limiting the application of constraint (27)–(30) to some extent. Unlike constraint (18) in (17)–(19), the constraint (28) not only requires the Lie derivative of the function $v(x)$ along the flow of system CS to be nonnegative over the set $\overline{R} \setminus \mathcal{X}_r$, but also over $\mathcal{X}_r \setminus \mathcal{X}$. One simple solution to remedy this defect is to combine constraints (27)–(30) and (17)–(19) together, and obtain a set of constraints, which is more expressive. These constraints are presented in Proposition 6.

**Proposition 6:** If there exist continuously differentiable functions $v_1(x), v_2(x) : \overline{R} \to R$ and $w(x) : \overline{R} \to R$ satisfying
\[
\begin{align*}
\nabla_v v_1(x) \cdot f(x) &\geq 0 \quad \forall x \in \mathcal{X}_r, \\
v_1(x) - \nabla_v w(x) \cdot f(x) &\leq 0 \quad \forall x \in \mathcal{X}_r, \\
v_2(x) &\leq 0 \quad \forall x \in \partial \mathcal{X}_r
\end{align*}
\]

where $\beta \in (0, +\infty)$, then the reach-avoid property in the sense of Definition 1 holds.

**Proof:** Let $x_0 \in \mathcal{R} = \{ x \in \overline{X} | v_1(x) + v_2(x) > 0 \}$, we have that $x_0 \in \{ x \in \overline{X} | v_1(x) > 0 \}$ or $x_0 \in \{ x \in \overline{X} | v_2(x) > 0 \}$. Following Propositions 4 and 5, we have the conclusion.

Due to constraint (34), $v_1(x)$ 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 the constraint (17)–(19). Thus, constraint (34)–(39) is weaker than both of constraints (27)–(30) and (17)–(19). The set $\mathcal{R} = \{ x \in \overline{X} | v_1(x) + v_2(x) > 0 \}$ is a mix of states entering the target set $\mathcal{X}_r$ at an exponential rate and ones entering the target set $\mathcal{X}_r$ at an asymptotic rate, thus we term $v_1(x) + v_2(x)$ asymptotic guidance-barrier function. However, we cannot guarantee that the set $\mathcal{R}$ still satisfies $\mathcal{R} \cap \mathcal{X}_r \neq \emptyset$ and it is an invariant for system CS until it enters the target set $\mathcal{X}_r$. If an initial state $x_0 \in \mathcal{R}_0$ is a state such that $v_2(x) > 0$, then the trajectory starting from it will stay inside the set $\mathcal{R}$ until it enters the target set $\mathcal{X}_r$, since
\[
\frac{d(v_1 + v_2)}{dt} = \nabla_v v_1(x) \cdot f(x) + \nabla_v v_2(x) \cdot f(x) \geq 0,
\]
holds; otherwise, we cannot have such a conclusion. Instead, we have that $\mathcal{R}_i = \{ x \in \overline{X} | v_2(x) > 0 \}$ satisfies $\mathcal{R}_i \cap \mathcal{X}_r \neq \emptyset$ and is an invariant for system CS until it enters the target set $\mathcal{X}_i$, if $\mathcal{R}_i \neq \emptyset$, where $i \in \{1, 2\}$. Let us illustrate this via an example.

**Example 3:** Consider the scenario in Example 1 again, but solve the constraint (34)–(39) with $\beta = 2$ for reach-avoid verification. The reach-avoid property is verified when polynomials of degree 12 are taken. The computed $\mathcal{R}$ is shown in Fig. 2. For this case, the set $\mathcal{R}_2 = \{ x \in \overline{X} | v_2(x) > 0 \}$ is empty. It is observed from Fig. 2 that the set $\mathcal{R}$ does not intersect $\mathcal{X}_r$, and the system CS leaves it before entering the target set $\mathcal{X}_r$. However, the set $\mathcal{R}_1$ is an invariant for the system CS until it enters the target set $\mathcal{X}_r$ and $\mathcal{R}_1 \cap \mathcal{X}_r \neq \emptyset$.

The other more sophisticated solution of enhancing constraint (27)–(30) is to replace the constraint (28) with
\[
\nabla_v v(x) \cdot f(x) \geq \alpha(x) \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_r
\]
where $\alpha(x) : \overline{X} \to R$ is a continuous function satisfying $\alpha(x) \geq 0$ over $\{ x \in \mathcal{X} \setminus \mathcal{X}_r | v(x) \geq 0 \}$. One instance for $\alpha(x)$ is $\beta(x)v(x)$, where $\beta(x) : \overline{X} \to [0, +\infty)$. The new constraints are formulated in Proposition 7.

**Proposition 7:** If there exists a continuously differentiable function $v(x) : \mathcal{R} \to R$, a continuous function $\alpha(x) : \mathbb{R}^n \to R$ satisfying $\alpha(x) \geq 0$ over $\{ x \in \mathcal{X} \setminus \mathcal{X}_r | v(x) \geq 0 \}$, and a continuously differentiable function $w(x) : \overline{R} \to R$ satisfying
\[
\begin{align*}
v(x) &> 0 \quad \forall x \in \mathcal{X}_r, \\
v(x) - \nabla_v w(x) \cdot f(x) &\leq 0 \quad \forall x \in \mathcal{X}_r, \\
v(x) &\leq 0 \quad \forall x \in \partial \mathcal{X}_r
\end{align*}
\]

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

**Proof:** Let $\mathcal{R} = \{ x \in \overline{X} | v(x) > 0 \}$. From constraint (41), we have that $\phi_{\alpha(t)}(t) \in \mathcal{X}_r$, $\nabla_v \phi_v(x) \cdot f(x) |_{x=\phi_{\alpha(t)}} \geq 0$. Then, following the arguments in the proof of Proposition 5, we have the conclusion.

Constraint (40)–(43) is less strict than constraint (27)–(30) in that the former only requires the Lie derivative of $v(x)$ along the flow of the system CS to be nonnegative over the set $\overline{R} \setminus \mathcal{X}_r$, rather than $\mathcal{X} \setminus \mathcal{X}_r$, due to constraint (41). The constraint (40)–(43) does not impose any restrictions on the Lie derivative of $v(x)$ along the flow of system CS over the set $\mathcal{X} \setminus \mathcal{X}_r$. Moreover, it is more expressive since it degenerates to the constraint (27)–(30) when $\alpha(x) \equiv 0$, and it is more expressive than the constraint (17)–(19), since the former degenerates to the latter when $\alpha(x) = \beta(x) v(x)$ and $w(x) = \frac{1}{\beta}(x)$. Besides, the set $\mathcal{R}$ obtained via solving constraint (40)–(43) will be an invariant for system CS until it enters the target set $\mathcal{X}_r$, and $\mathcal{R} \cap \mathcal{X}_r \neq \emptyset$.

**Example 4:** Consider the system in Example 1 with $\mathcal{X} = \{ (x, y) | x^2 + y^2 - 1 < 0 \}$, $\mathcal{X}_r = \{ (x, y) | (x + 0.2)^2 + (y - 0.7)^2 - 0.02 < 0 \}$, and $\mathcal{X}_0 = \{ (x, y) | 0.1 - x < 0, -0.5 < y < 0, 0.8 - y < 0, y + 0.4 < 0 \}$.

In this example, we use $\alpha(v(x)) = x^2v(x)$ to illustrate the benefits of constraints (40)–(43) on reach-avoid verification. The degree of all polynomials in the resulting semi-definite program is taken the same and in order of $[2, 4, 6, 8, 10, \ldots, 20]$. When the reach-avoid property is verified successfully, the computations terminate. The degree is 12 for termination. In contrast, the degree is 14 when using constraints (17)–(19) with $\beta = 0.1$ and (27)–(30) to verify the reach-avoid property. Consequently, these experiments further support our analysis that the constraint (40)–(43) is more expressive and can provide more chances for verifying the reach-avoid property successfully.

**IV. Examples**

We further demonstrate the theoretical development and performance of the proposed conditions on several examples, i.e., Examples 5–9. In the computations, the degree of unknown polynomials in the resulting semidefinite programs is taken the same and in order of $[2, 4, 6, 8, 10, \ldots, 20]$. When the reach-avoid property is verified successfully, the computations terminate. A return of “Successfully
solved (MOSEK)\(^\text{\textregistered}\) 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 Section III-C, the smaller the discount factor \(\beta\) 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 the constraint (17)-(19) with \(\beta < 0.1\) for more comprehensive and fair comparisons with the proposed methods in the present work. In these experiments, \(\beta = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, \) and \(10^{-6}\) 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 the constraint (40)-(43) over constraints (17)-(19) and (27)-(30) 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 \(\mathcal{X} = \{(x, y)^\top \mid x^2 + y^2 < 1, \ x < 0, \ y < 0\}\) and \(\mathcal{X}_0 = \{(x, y)^\top \mid x < -0.6, \ x < 0.8, -y < 0, y < 0.2\}\), and \(\mathcal{X} = \{(x, y)^\top \mid x^2 + y^2 < 0.01, \ y < 0\}\).

The reach-avoid property is verified when the degree is 8, 8, and 12 for constraints (2)-(4), (27)-(30), and (17)-(19) with \(\beta = 0.1\), respectively. We did not obtain any positive verification result from the constraint (17)-(19) with \(\beta = 1\). However, it can be improved by solving the constraint (34)-(39) with \(\beta = 1\) and the reach-avoid property is verified when the degree is 8. Further, if the constraint (40)-(43) is used with \(\alpha(x) = x^2v(x)\) the degree is 6. Some of the computed sets \(\mathcal{R}\) are illustrated in Fig. 3.

Besides, we also experimented using constraints (17)-(19) and (34)-(39) with \(\beta = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, \) and \(10^{-6}\) for more comprehensive and fair comparisons. The computations terminate when the degree takes 8 for all of these experiments.

**Example 7:** Consider the following system from [24]:

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

with \(\mathcal{X} = \{x \in \mathbb{R}^2 \mid x^2 + y^2 < 4, \ x < 0\}\), \(\mathcal{X}_0 = \{x \in \mathbb{R}^2 \mid x < -1.2, \ y < 0.8\}\), and \(\mathcal{X} = \{x \in \mathbb{R}^2 \mid x^2 + y^2 < 0.3, \ y < 0\}\).

The reach-avoid property in the sense of Definition 1 is not verified using the constraint (2)-(4). Actually, it cannot be verified via solving the constraint (2)-(4), since there exists an equilibrium in the set \(\mathcal{X} \setminus \mathcal{X}_0\).

The reach-avoid property is verified when the degree is 10 for constraints (27)-(30) and (17)-(19) with \(\beta = 1\) and \(\beta = 0.1\). If the constraint (40)-(43) is used with \(\alpha(x) = (2 - y)v(x)\), the degree is 8. Some of the computed sets \(\mathcal{R}\) are illustrated in Fig. 4.

Like Example 6, we also experimented using constraints (17)-(19) and (34)-(39) with \(\beta = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, \) and \(10^{-6}\) for more comprehensive and fair comparisons. The computations terminate when the degree takes 10 for all of these experiments.

**Example 8:** Consider the following system from [24]:

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

with \(\mathcal{X} = \{x \in \mathbb{R}^2 \mid x^2 + y^2 < 1, \ x < 0\}\), \(\mathcal{X}_0 = \{x \in \mathbb{R}^2 \mid (x + 1)^2 + (y - 1.5)^2 < 0.25, \ y < 0\}\), and \(\mathcal{X} = \{x \in \mathbb{R}^2 \mid x^2 + y^2 < 0.01, \ y < 0\}\).

The reach-avoid property is not verified using the constraint (2)-(4), and is verified when the degree is 10 for the constraint (27)-(30). We did not obtain any positive verification result from solving constraint (17)-(19) with \(\beta = 1\) and \(\beta = 0.1\). However, this negative situation can be improved by solving constraint (34)-(39) with \(\beta = 1\) and \(\beta = 0.1\), and the reach-avoid property is verified when the degree is 10. If the constraint (40)-(43) is used with \(\alpha(x) = (x + y)^2v(x)\), the degree is 6. Furthermore, if constraint (40)-(43) is used with \(\alpha(x) = x^2v(x)\), the degree is 4. Some of the computed sets \(\mathcal{R}\) are illustrated in Fig. 5.

Analogously, we also experimented using constraint (17)-(19) and (34)-(39) with \(\beta = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, \) and \(10^{-6}\) for more comprehensive and fair comparisons. The computations terminate when the degree is 10 for all of these experiments.

**Example 9 (Dubin’s Car):** Consider the Dubin’s car: \(\dot{x} = v \cos(\theta), \ \dot{\theta} = \omega, \ v = \cos(\theta) + b \sin(\theta), \ \dot{y} = -2(a \sin(\theta) - b \cos(\theta)) + \theta y\) with \(u_1 = \omega\) and \(u_2 = -v \omega (a \sin(\theta) - b \cos(\theta))\), it is transformed into polynomial dynamics

\[
\begin{cases}
    \dot{x} = u_1 \\
    \dot{y} = u_2 \\
    \dot{z} = y u_1 - x u_2
\end{cases}
\]
with \( u_2 = 2, u_2 = 1 + z - xy, \mathcal{X} = \{ x \in \mathbb{R}^3 | x^2 + y^2 + z^2 < 4 \}, \mathcal{X}_0 = \{ x \in \mathbb{R}^3 | (x + 0.6)^2 + y^2 + (z + 0.6)^2 - 0.02 < 0 \}, \) and \( \mathcal{X}_2 = \{ x \in \mathbb{R}^3 | x^2 + y^2 + z^2 < 4 \}, (x - 1.0)^2 - (y + 0.5)^2 + (z + 0.1)^2 - 0.1 < 0 \} \).

The reach-avoid property is verified when degree is 8 for all constraints (21)-(24), (27)-(30), and (17)-(19) with \( \beta \in \{ 1, 0.1, \ldots, 10^{-5} \} \), (34)-(39) with \( \beta \in \{ 0.1, 0.1, \ldots, 10^{-5} \} \). However, if the constraint (40)-(43) is used with \( \alpha(x) = (1 - x)^e(x) \), the degree is 6.

Examples mentioned previously, i.e., Example 5-9, indicate that when the discount factor is small, the constraint (17)-(19) has the same performance with the constraint (27)-(30), although it performs worse when the discount factor is large. On the other hand, the constraint (34)-(39) indeed is able to improve the performance of the constraint (17)-(19) when the discount factor is large, but it does not improve the constraint (27)-(30) and its performance will be the same with the constraint (17)-(19) when the discount factor is small. However, the constraint (40)-(43) outperforms the former three, i.e., constraints (17)-(19), (27)-(30), and (34)-(39), and constraint (21)-(24). 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 the constraint (40)-(43) 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 this article, we only demonstrate the performance of all of quantified constraints by relaxing them into semidefinite constraints and addressing them within the semidefinite programming framework, which could be solved efficiently via interior point methods in polynomial time. It is worth remarking here that besides semidefinite 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 article and leave these investigations for ones of interest.

V. CONCLUSION

In this article, 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, which is a simplified version of the one in the 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 semialgebraic, 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 constraints.

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,” in Proc. 24th Int. Conf. Hybrid Syst., Comput. Control, 2021, Art. no. 24.
[2] M. Althoff, “Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets,” in Proc. 16th Int. Conf. Hybrid Syst., Comput. Control, 2013, pp. 173–182.
[3] M. Althoff, G. Frehsre, and A. Girard, “Set propagation techniques for reachability analysis,” Automatica, vol. 44, no. 1, pp. 369–395, 2008.
[4] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in Proc. 18th Eur. Control Conf., 2019, pp. 3420–3431.
[5] E. Asarin, O. Bournez, T. Dang, and O. Maler, “Approximate reachability analysis of piecewise-linear dynamical systems,” in Proc. 3rd Int. Workshop Hybrid Syst., Comput. Control, 2000, pp. 20–31.
[6] S. Bak, “t-Barrier certificates: A continuous analogy to k-induction,” IFAC PapersOnLine, vol. 51, no. 16, pp. 145–150, 2018.
[7] M. Berz and K. Makino, “Verified integration of odes and flows using differential algebraic methods on high-order Taylor models,” Reliable Comput., vol. 4, no. 4, pp. 361–369, 1998.
[8] T. Braunl, Embedded Robotics: Mobile Robot Design and Applications With Embedded Systems. Berlin, Germany: Springer, 2008.
[9] L. Breger and J. P. How, “Safe trajectories for autonomous rendezvous of spacecraft,” J. Guid., Control, Dyn., vol. 31, no. 5, pp. 1478–1489, 2008.
[10] X. Chen, E. Abraham, and S. Sankaranarayanan, “Taylor model flowpipe construction for non-linear hybrid systems,” in Proc. IEEE 33rd Real-Time Syst. Symp., 2012, pp. 183–192.
[11] A. Eggers, N. Ramdani, N. S. Nedialkov, and M. Fränzel, “Improving the saturation modulo oode approach to hybrid systems analysis by combining different enclosure methods,” Softw. Syst. Model., vol. 14, no. 1, pp. 121–148, 2015.
[12] M. Fränzel, M. Chen, and P. Kröger, “In memory of oded maler: Automatic reachability analysis of hybrid-state automata,” ACM SIGSOFT News., vol. 6, no. 1, pp. 19–39, 2019.
[13] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, “What’s decidable about hybrid automata?,” J. Comput. Syst. Sci., vol. 57, no. 1, pp. 94–124, 1998.
[14] 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 Proc. Int. Conf. Comput. Aided Verification, 2013, pp. 242–257.
[15] M. Korda, D. Henrion, and C. N. Jones, “Controller design and region of attraction estimation for nonlinear dynamical systems,” IFAC Proc. Vol., vol. 47, no. 3, pp. 2310–2316, 2014.
[16] U. Kulisch, R. Lohner, and A. Facius, Perspectives on Enclosure Methods. Berlin, Germany: Springer, 2001.
[17] E. A. Lee, “Cyber physical systems: Design challenges,” in Proc. IEEE 11th Int. Symp. Object Component-Orbile Real-Time Distrib. Comput., 2008, pp. 363–369.
[18] J. Lobberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” in Proc. IEEE Int. Conf. Robot. Automat., 2004, pp. 284–289.
[19] The MOSEK Optimization Toolbox for Matlab Manual, MOSEKAP, Copenhagen, Denmark, 2010.
[20] S. Prajna and A. Jadbabaie, “Safety verification of hybrid systems using barrier certificates,” in Proc. Int. Workshop Hybrid Syst., Comput. Control, 2004, pp. 477–492.
[21] S. Prajna, A. Jadbabaie, and G. J. Pappas, “A framework for worst-case and stochastic safety verification using barrier certificates,” IEEE Trans. Autom. Control, vol. 52, no. 8, pp. 115–1428, Aug. 2007.
[22] S. Prajna and A. Rantzer, “Convex programs for temporal verification of nonlinear dynamical systems,” SIAM J. Control Optim., vol. 46, no. 3, pp. 999–1021, 2007.
[23] 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 Trans. Autom. Control, vol. 54, no. 10, pp. 2352–2364, Oct. 2009.
[24] W. Tan and A. Packard, “Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming,” IEEE Trans. Autom. Control, vol. 53, no. 2, pp. 565–571, Mar. 2008.
[25] B. C. Williams, M. D. Ingham, S. H. Chung, and P. H. Elliott, “Model-based programming of intelligent embedded systems and robotic space explorers,” Proc. IEEE, vol. 91, no. 1, pp. 212–237, Jan. 2003.
[26] B. Xue, A. Easwaran, N.-J. Cho, and M. Fränzel, “Reach-avoid verification for nonlinear systems based on boundary analysis,” IEEE Trans. Autom. Control, vol. 62, no. 7, pp. 3518–3523, Jul. 2017.