Robust Invariant Sets Computation for Switched Discrete-Time Polynomial Systems

Bai Xue and Naijun Zhan
State Key Lab. of Computer Science, Institute of Software, CAS, China
Email: {xuebai,znj}@ios.ac.cn

Abstract

In this paper we systematically study the problem of computing robust invariant sets for switched discrete-time polynomial systems subject to state constraints from theoretical and computational perspectives. A robust invariant set of interest in this paper is a set of states such that every possible trajectory starting from it never violates a specified state constraint, regardless of actions of perturbations. We show that the maximal robust invariant set can be characterized as the zero level set of the unique bounded solution to a modified Bellman equation. The uniqueness property of the solution facilitates use of existing numerical methods to solve the Bellman equation for an appropriate number of state variables in order to obtain an approximation of the maximal robust invariant set. Especially when there is only one subsystem in the switched system, the solution to the derived equation can be Lipschitz continuous. We further relax the equation into a system of inequalities and encode these inequality constraints using sum of squares decomposition for polynomials. This results in the computation of robust invariant sets by solving a single semi-definite program, which can be addressed via interior point methods in polynomial time. Finally, three examples demonstrate the performance of our methods.

1 Introduction

The computation of robust invariant sets is central to validation of systems such as programs (or discrete-time systems), physical systems (or continuous-time systems) or hybrid systems [5]. A robust invariant set of interest in this paper refers to a set of states such that every possible trajectory initialized in it never violates a specified state constraint irrespective of the actual perturbation. It goes by numerous other names in the literatures, e.g., infinite-time reachability tubes [27] and invariance kernels in viability theory [3]. Due to its widespread applications, robust invariant sets generation has been the subject of extensive research over past several decades, thereby producing a large amount of works on this topic, e.g., Lyapunov function methods [8, 14, 13, 11], fixed point methods [17, 29] and viability theory [3].

The present work studies the robust invariant sets generation problem by exploiting the link to optimal control through solutions of Bellman equations. Bellman equations are a class of equations widely used in discrete-time continuous-time optimal control theory [4]. Establishing the link to optimal control through viscosity solutions of Hamilton-Jacobi equations, which are widely used in optimal control theory [4], is an attractive means in studying the reachability problem of continuous-time systems, e.g., [3, 22, 24]. One advantage of such methods is the existence of well-developed numerical methods [9, 23, 27, 1] for solving Hamilton-Jacobi equations or Bellman

---

*This is an extended version of [33]. The robust invariant set in the present paper is equivalent to the robust nontermination set in [33] when the program of interest in [33] is transformed into switched discrete-time systems. In this version we further theoretically characterize the maximal robust invariant set as the zero level set of the unique bounded solution to a Bellman equation, thus facilitating use of existing numerical methods to obtain an approximation of the maximal robust invariant set. Especially if there is only one subsystem in the switched system, we further explore the continuity property of the solution. Moreover, we compare the performances of the methods in the current paper and the ones in [33] on three illustrative examples.

1A switched system is defined by a family of subsystems and a switching rule orchestrating the switching between subsystems.
equations with appropriate number of state variables, rendering possible the gain of an approximation of the maximal robust invariant set. Despite the rich literature regarding nonlinear continuous-time systems, studies on the computation of robust invariant sets for its counterpart, i.e. discrete-time systems, are relatively sparse, especially for switched discrete-time nonlinear systems. The switched discrete-time systems is defined by a family of subsystems and a switching rule orchestrating the switching between subsystems. However, the importance of discrete-time systems is self-evident in real applications. The widely known application is the simulation of a continuous-time system by a digital computer. Most of existing works on the computation of robust invariant sets for discrete-time systems focus on linear systems, e.g. [15, 26, 12, 28, 5, 2, 30, 31]. Only a small amount is on nonlinear cases. [10] proposed an algorithm for computing a polytopic robust invariant set for perturbation-affine discrete-time nonlinear systems. [20] considered the estimation of domains of attraction for switched and hybrid nonlinear systems based on semi-definite programs under the assumption that a Lyapunov function is given. [18] estimated robust invariant sets for perturbation-affine systems with competing inputs (control and perturbation) based on fixed point algorithms.

In this paper we study the generation of robust invariant sets for switched discrete-time nonlinear systems subject to state constraints. We firstly define a bounded value function with a positive-valued parameter such that its zero sublevel set is equal to the maximal robust invariant set. Then the value function is reduced to a bounded solution to a modified Bellman equation. When the value of the parameter is strictly between 0 and 1, the solution is unique. The uniqueness property of the bounded solution facilitate use of existing numerical methods such as value iteration methods to solve the Bellman equation for an appropriate number of state variables in order to obtain an approximation of the maximal robust invariant set. When the parameter is equal to one, the inferred Bellman equation does not feature this uniqueness property. Especially, when there is only one subsystem in the switched system, the value function is either Lipschitz continuous when the parameter value is less than one or lower semicontinuous. As to the case that the parameter value is equal to one, we relax the Bellman equation into a system of inequalities and construct a convex optimization based method such that a robust invariant set could be synthesized via addressing a single semi-definite program, which falls within the convex programming category. Finally, three illustrative examples demonstrate the performance of our approaches.

The main mathematical tools we use are dynamic programming, which allows us to reduce the maximal robust invariant set of switched discrete-time nonlinear systems to the zero level set of the unique bounded solution to a Bellman equation, and sum-of-squares decomposition for polynomials, which facilitates the generation of robust invariant sets via solving a single semi-definite program. The main contributions of this paper are summarized as follows:

1. We for the first time infer that the maximal robust invariant set of switched discrete-time polynomial systems subject to state constraints can be characterized as the zero level set of the unique bounded solution to a Bellman equation. Existing well-developed numerical methods can be employed to address this equation for obtaining an approximation of the maximal robust invariant set. To the best of our knowledge this is the first possibility to calculate the maximal robust invariant set for switched discrete-time nonlinear systems subject to state constraints available in the literature. Although we only consider systems of the polynomial type in the present work, this approach is applicable for more general nonlinear systems.

2. The existing numerical methods for solving the derived equation generally require partitioning the state space and thus exponential cost in the dimension of the problem. In order to overcome such limitation, on the basis of the derived Bellman equation we construct a semi-definite program, which can be efficiently solved by interior-point methods in polynomial time, to synthesize robust invariant sets.

Organization of the paper. The structure of this paper is as follows. In Section 2, basic notions used throughout this paper and the problem of interest are introduced. Then we elucidate our approach for synthesizing robust invariant sets in Section 3. After demonstrating our approach on three illustrative examples in Section 4, we end up with concluding this paper in Section 5.

2 Preliminaries

The following basic notations will be used throughout the rest of this paper: \( \mathbb{N} \) stands for the set of nonnegative integers and \( \mathbb{R} \) for the set of real numbers; \( \mathbb{R}[\cdot] \) denotes the ring of polynomials in variables given by the argument,
Definition 1. A switched discrete-time polynomial system (SD) subject to state and perturbation inputs, and the notion of the (maximal) robust invariant set of interest in this paper.

2.1 Problem Description

In this section we describe the switched discrete-time polynomial system subject to state constraints and perturbation inputs, and the notion of the (maximal) robust invariant set of interest in this paper.

Definition 1. A switched discrete-time polynomial system (SD) subject to state and perturbation inputs is a quintuple $(x_0, X_0, \mathcal{X}, D, L)$ with

- $x_0 \in \Omega$ is the initial state;
- $X_0 \subseteq \mathbb{R}^n$ is the state constraint set, which is a compact set. A path can evolve complying with the discrete dynamics only if its current state is in $X_0$;
- $\mathcal{X} := \{X_i, i = 1, \ldots, k\}$ with $X_i \cap X_j = \emptyset$ if $i \neq j$ and $\bigcup_{i=1}^k X_i = \mathbb{R}^n$;
- $D \subseteq \mathbb{R}^m$ is the set of perturbation inputs;
- $L := \{f_i(x, d), i = 1, \ldots, k\}$,

where $X_0 = \{x \in \mathbb{R}^n \mid \bigwedge_{i=1}^n h_{0,i}(x) \leq 0\}$, $X_i = \{x \in \mathbb{R}^n \mid \bigwedge_{j=1}^n h_{i,j}(x) > 0\}$ with $\triangleright \in \{\leq, <\}$, $i = 1, \ldots, k$, and $D = \{d \in \mathbb{R}^m \mid \bigwedge_{i=1}^{n+1} h_{k+1,i}(d) \leq 0\}$. Also, $f_i(x, d) \in \mathbb{R}[x, d]$, $i = 1, \ldots, k$; $h_{i,j}(x) \in \mathbb{R}[x, i = 0, \ldots, k, j = 1, \ldots, n_i; h_{k+1,j}(x) \in \mathbb{R}[d], j = 1, \ldots, n_{k+1}$.

Before defining the trajectory to SD, we define an input policy controlling a trajectory.

Definition 2. An input policy $\pi$ is an ordered sequence $\{d(i), i \in \mathbb{N}\}$, where $d(\cdot) \in \mathbb{N} \mapsto D$, and $\Pi$ is defined as the set of input policies, i.e. $\Pi = \{\pi | d(\cdot) : \mathbb{N} \mapsto D\}$.

Under an input policy $\pi$, the trajectory $\phi_{x_0}^D : \mathbb{N} \mapsto \mathbb{R}^n$ to SD starting from an initial state $x_0$ follows the discrete dynamics defined by

$$\phi_{x_0}^D(l + 1) = f(\phi_{x_0}^D(l), d(l)), \quad (1)$$

where $\phi_{x_0}^D(0) = x_0$, $\phi_{x_0}^D(l) \in X_0$ for $l \in \mathbb{N}$ and

$$f(x, d) = 1_{X_i} \cdot f_i(x, d) + \cdots + 1_{X_k} \cdot f_k(x, d) \quad (2)$$

with $1_{X_i} : X_i \mapsto \{0, 1\}$, $i = 1, \ldots, k$, representing the indicator function of the set $X_i$, i.e.

$$1_{X_i} := \begin{cases} 1, & \text{if } x \in X_i, \\ 0, & \text{if } x \notin X_i. \end{cases} \quad (3)$$

Now, we define our problem of deciding a set of initial states such that SD starting from it never leave the set $X_0$.

Definition 3 (Maximal Robust Invariant Set). The maximal robust invariant set $R_0$ is a subset of $X_0$ such that every possible trajectory of system SD starting from it never leaves $X_0$, i.e.

$$R_0 = \{x_0 \mid \phi_{x_0}^D(l) \in X_0, \forall l \in \mathbb{N}, \forall \pi \in \mathcal{D}\}. \quad (3)$$

A robust invariant set is a subset of the maximal robust invariant set $R_0$.

3 Robust Invariant Sets Generation

In this section we elucidate our approach of addressing the robust invariant sets generation problem for SD. We firstly in Subsection 3.1 characterize the maximal robust invariant set $R_0$ as the unique bounded solution to a modified Bellman equation, which can be solved by existing numerical methods. Furthermore, a computationally tractable semi-definite programming method is proposed to synthesize robust invariant sets in Subsection 3.2.
3.1 Characterization of $\mathcal{R}_0$

In this subsection we firstly introduce a bounded value function to characterize the maximal robust invariant set $\mathcal{R}_0$ and then formulate it as a bounded solution to a modified Bellman equation. The properties of the solution such as uniqueness and continuity are explored as well.

Let $h_{0,j}(x) = \frac{h_{\alpha,j}(x)}{1 + h_{\alpha,j}(x)}$. Thus, $-1 < h_{0,j}(x) < 1$ over $x \in \mathbb{R}^n$ for $j = 1, \ldots, n_0$. For $x \in \mathbb{R}^n$, given a scalar value $\alpha \in (0, 1]$, the value function $V : \mathbb{R}^n \mapsto \mathbb{R}$ is defined by:

$$V(x) := \sup_{\pi \in \mathcal{D}} \sup_{l \in \mathbb{N}} \max_{j \in \{1, \ldots, n_0\}} \left\{ \alpha^l h_{0,j}^l(\phi_x^\pi(l)) \right\}. \quad (4)$$

Obviously, $-1 \leq V(x) \leq 1$ holds for $x \in \mathbb{R}^n$.

Different from [33], we introduce a parameter $\alpha$ into the construction of the value function [4]. This enables us to reduce it a unique bounded solution to a Bellman equation, which will be shown later.

The following theorem shows the relation between the value function $V$ and the maximal robust invariant set $\mathcal{R}_0$, that is, the zero sublevel set of $V(x)$ is equal to the maximal robust invariant set $\mathcal{R}_0$.

**Theorem 1.** $\mathcal{R}_0 = \{x \in \mathbb{R}^n \mid V(x) \leq 0\}$, where $\mathcal{R}_0$ is the maximal robust invariant set as in Definition [3].

Typically, when $\alpha \in (0, 1)$, $V(x) \geq 0$ for $x \in \mathbb{R}^n$ and thus $\mathcal{R}_0 = \{x \in \mathbb{R}^n \mid V(x) = 0\}$.

**Proof.** Let $y \in \mathcal{R}_0$. According to Definition [3] we have that

$$h_{0,j}(\phi_y^\pi(i)) \leq 0, \forall i \in \mathbb{N}, \forall \pi \in \mathcal{D} \text{ and } \forall j \in \{1, \ldots, n_0\}. \quad (5)$$

holds, implying that

$$h_{0,j}^l(\phi_y^\pi(i)) \leq 0, \forall i \in \mathbb{N}, \forall \pi \in \mathcal{D}, \forall j \in \{1, \ldots, n_0\}$$

and thus $V(y) \leq 0$. Therefore, $y \in \{x \mid V(x) \leq 0\}$.

On the other side, if $y \in \{x \in \mathbb{R}^n \mid V(x) \leq 0\}$, then $V(y) \leq 0$, implying that [5] holds and consequently $y \in \mathcal{R}_0$. Therefore, $\mathcal{R}_0 = \{x \in \mathbb{R}^n \mid V(x) \leq 0\}$.

As to the case of $\alpha \in (0, 1)$, it is evident that $V(x) \geq 0$ for $x \in \mathbb{R}^n$ since

$$\lim_{l \to \infty} \alpha^l h_{0,j}^l(\phi_x^\pi(l)) = 0, \forall x \in \mathbb{R}^n, \forall \pi \in \mathcal{D}.$$

Therefore, $\mathcal{R}_0 = \{x \in \mathbb{R}^n \mid V(x) = 0\}$ if $\alpha \in (0, 1)$. \hfill \square

From Theorem [1] the maximal robust invariant set $\mathcal{R}_0$ can be constructed by computing $V(x)$. When there is only one subsystem in the system SD, $V(x)$ is lower semicontinuous if $\alpha = 1$ and Lipschitz continuous if $\alpha \in (0, 1)$, respectively. However, $V(x)$ may not feature such property when switching exists.

**Lemma 1.** Suppose there is only one subsystem in SD, i.e. $\mathcal{X} = \{X_i\}$ with $X_i = \mathbb{R}^n$ in SD, $i \in \{1, \ldots, k\}$. If $\alpha = 1$, $V(x)$ in [4] is lower semicontinuous over $x \in \mathbb{R}^n$.

**Proof.** We need to prove that for every $\epsilon > 0$ there exists $\delta > 0$ such that

$$V(x) - \epsilon < V(y), \forall y \text{ satisfying } \|y - x\| < \delta.$$

To see this, consider $x \in \mathbb{R}^n$. For every $\epsilon > 0$ there exist $K \in [0, \infty)$ and $\pi \in \mathcal{D}$ such that

$$\max_{j \in \{1, \ldots, n_0\}} h_{0,j}^l(\phi_x^\pi(K)) - V(x) |< \frac{\epsilon}{2}.$$ 

Since $h_{0,j}(x)$, $j = 1, \ldots, n_0$, and $f(x, d) = f_i(x, d)$ is locally Lipschitz uniformly over $d \in D$, on a finite-horizon $[0, K]$, there exists $\delta > 0$ such that

$$\max_{j \in \{1, \ldots, n_0\}} h_{0,j}^l(\phi_x^\pi(K)) - \max_{j \in \{1, \ldots, n_0\}} h_{0,j}^l(\phi_y^\pi(K)) |< \frac{\epsilon}{2} \quad (6)$$
for \( y \) satisfying \( \|x - y\| < \delta \). Therefore, we have that
\[
|V(x) - \max_{j \in \{1, \ldots, n_0\}} \alpha^j h^\prime_{0,j}(\phi^\prime_y(K))| < \epsilon.
\]

Also, since \( V(y) \geq \max_{j \in \{1, \ldots, n_0\}} \alpha^j h^\prime_{0,j}(\phi^\prime_y(K)) \),
\[
V(x) - \epsilon < V(y)
\]
holds for \( y \) satisfying \( \|y - x\| < \delta \). The proof is completed.

**Lemma 2.** Let \( \mathcal{X} = \{X_i\} \) in \( \mathcal{SD} \), where \( i \in \{1, \ldots, k\} \), i.e. \( X_i = \mathbb{R}^n \). If \( \alpha \in (0, 1) \), \( V(x) \) in [4] is locally Lipschitz continuous over \( x \in \mathbb{R}^n \).

**Proof.** First, it is obvious that for \( y \in B(x, \delta) \), where \( B(x, \delta) = \{y \mid \|y - x\| \leq \delta\} \) with \( \delta > 0 \),
\[
|V(x) - V(y)| \leq \sup_{\tau \in \mathcal{D}} \sup_{l \in [0, K]} \max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))|.
\]
(7)

As \( -1 < h^\prime_{0,j}(\cdot) : \mathbb{R}^n \to \mathbb{R} < 1 \) over \( \mathbb{R}^n \) for \( j = 1, \ldots, n_0 \) and \( \alpha \in (0, 1) \), this implies that the supremum
\[
\sup_{\tau \in \mathcal{D}} \sup_{l \in [0, K]} \max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))| = 0,
\]
(8)

is attained on a finite interval \([0, K] \cap \mathbb{N}\). The conclusion can be justified as follows: If the supremum is zero, then
\[
\max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))| \equiv 0, \forall \pi \in \mathcal{D}, \forall l \in \mathbb{N}.
\]
Therefore, the supremum can be attained on any finite time interval \([0, K] \cap \mathbb{N}\). Otherwise, assume that the supremum equals a positive value \( \epsilon_1 \). Since
\[
\max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))| \leq 2\alpha^j, \forall \pi \in \mathcal{D},
\]
there exists a finite value \( K > 0 \) such that
\[
\sup_{\tau \in \mathcal{D}} \sup_{l \in [0, K]} \max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))| \leq 2\alpha^j \leq \frac{\epsilon_1}{2}.
\]
Therefore, \( \epsilon_1 \) is attained on the finite time interval \([0, K], \mathcal{i.e.} \)
\[
\sup_{\tau \in \mathcal{D}} \sup_{l \in [0, K]} \max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))| = 0
\]
\[
\sup_{\tau \in \mathcal{D}} \sup_{l \in [0, K]} \max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))|.
\]
Since \( h^\prime_{0,j}(x) \) is locally Lipschitz continuous since \( h_{0,j}(x) \) and \( f(x, d) = f_i(x, d) \) is locally Lipschitz continuous over \( x \) uniformly over \( d \in D \), we have that
\[
|V(x) - V(y)| \leq \sup_{\tau \in \mathcal{D}} \sup_{l \in [0, K]} \max_{j \in \{1, \ldots, n_0\}} |\alpha^j h^\prime_{0,j}(\phi^\prime_z(l)) - \alpha^j h^\prime_{0,j}(\phi^\prime_y(l))| \leq \sup_{\tau \in \mathcal{D}} \sup_{l \in [0, K]} \max_{j \in \{1, \ldots, n_0\}} |h^\prime_{0,j}(\phi^\prime_z(l)) - h^\prime_{0,j}(\phi^\prime_y(l))| \leq \max_{j \in \{1, \ldots, n_0\}} L_{h^\prime_{0,j}} \sup_{k \in [0, K]} L^j_k \|x - y\| \leq \max_{j \in \{1, \ldots, n_0\}} L_{h^\prime_{0,j}} \|x - y\|,
\]
(9)
where \( L_{h^\prime_{0,j}} \) and \( L^j_k \) are respectively the Lipschitz constants of \( h^\prime_{0,j} \) and \( f \) over \( \Omega(B(x, \delta), K), \Omega(B(x, \delta), K) \) is the reachable set of \( B(x, \delta) \) within the time interval \([0, K] \cap \mathbb{N}\), i.e.
\[
\Omega(B(x, \delta), K) = \{x' \mid x' = \phi^\prime_y(l), \forall y \in B(x, \delta), \forall \pi \in \mathcal{D}, \forall l \in [0, K] \cap \mathbb{N}\}.
\]
(10)
[9] shows the desired Lipschitz continuity.
Moreover, \( V(x) \) satisfies the following dynamic programming principle:

**Lemma 3.** For \( x \in \mathbb{R}^n \) and \( l \in \mathbb{N} \), we have:

\[
V(x) = \sup_{\pi \in \mathcal{D}} \max \left\{ \alpha^i V(\phi_x^\pi(l)) \right\}, \quad \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \right\}.
\]

(11)

**Proof.** Let

\[
W(l, x) : = \sup_{\pi \in \mathcal{D}} \max \left\{ \alpha^i V(\phi_x^\pi(l)) \right\}, \quad \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \right\}.
\]

(12)

We will prove that for \( \epsilon > 0 \), \( |W(l, x) - V(x)| < \epsilon \).

According to the definition of \( V(x) \), i.e. (11), for any \( \epsilon_1 \), there exists an input policy \( \pi' \) such that

\[
V(x) \leq \sup_{i \in \mathbb{N}} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \} + \epsilon_1.
\]

We then introduce two input policies \( \pi_1 \) and \( \pi_2 \) with \( d_1(j) = d(j) \) for \( j = 0, \ldots, l - 1 \) and \( d_2(j) = d'(j) + l \) for \( j \in \mathbb{N} \), respectively. Now, let \( y = \phi_x^{\pi_1}(l) \), then we obtain that

\[
W(l, x) \geq \sup_{i \in \mathbb{N}} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \}
\]

\[
\geq \max \left\{ \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \} \right\} \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \}
\]

\[
= \max \left\{ \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \} \right\} \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi(\phi_x^\pi(i)) \}
\]

\[
\geq V(x) - \epsilon_1.
\]

Therefore,

\[
V(x) \leq W(l, x) + \epsilon_1.
\]

14

On the other side, by the definition of \( W(l, x) \), for every \( \epsilon_1 > 0 \), there exists \( \pi_1 \in \mathcal{D} \) such that

\[
W(l, x) \leq \max \left\{ \alpha^i V(\phi_x^\pi_1(l)) \right\}, \quad \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi_1(\phi_x^\pi_1(i)) \} + \epsilon_1.
\]

(15)

Also, by the definition of \( V(x) \), i.e. (11), for every \( \epsilon_1 > 0 \), there exists a \( \pi_2 \) such that

\[
V(y) \leq \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi_2(\phi_x^\pi_2(i)) \} + \epsilon_1,
\]

where \( y = \phi_x^{\pi_1}(l) \). We define \( \pi \in \mathcal{D} \) such that \( d(i) = d_1(i) \) for \( i = 0, \ldots, l - 1 \) and \( d(i + l) = d_2(i) \) for \( i \in \mathbb{N} \). Then, it follows

\[
W(l, x) \leq 2\epsilon_1 + \max \left\{ \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi_1(\phi_x^\pi_1(i)) \} \right\} \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi_2(\phi_x^\pi_2(i)) \}
\]

\[
\leq \max \left\{ \sup_{i \in [0,l]} \max_{j \in \{1,\ldots,n_0\}} \{ \alpha^i h_{0,j}^\pi_1(\phi_x^\pi_1(i)) \} + 2\epsilon_1 \right\}
\]

\[
\leq V(x) + 2\epsilon_1.
\]

Combining (14) and (16), we finally have \( |V(x) - W(l, x)| \leq \epsilon = 2\epsilon_1 \). Since \( \epsilon_1 \) is arbitrary, \( V(x) = W(l, x) \) holds. This completes the proof.

Based on Lemma 3, we derive a central equation of this paper, to which \( V(x) \) is a solution. The equation is formally formulated in Theorem 2.
Theorem 2. The value function $V(x) : \mathbb{R}^n \to \mathbb{R}$ in (17) is a bounded solution to the modified Bellman functional equation
\[
\min \left\{ \inf_{d \in D} (V(x) - \alpha V(f(x, d))), V(x) - \max_{j \in \{1, \ldots, n\}} h'_{0,j}(x) \right\} = 0.
\]
Moreover,
1. $V(x)$ is the unique bounded solution to (17) when $\alpha \in (0, 1)$.
2. if there is only one subsystem in $S_D$, i.e. $\mathcal{X} = \{X_i\}$ with $X_i = \mathbb{R}^n$ in $S_D$, $i \in \{1, \ldots, k\}$, $V(x)$ with $\alpha \in (0, 1)$ is the unique Lipschitz continuous solution to (17).

Proof. It is obvious that (17) is the special case of (11) when $l = 1$. In the following we just prove the statement for $\alpha \in (0, 1)$.

(1). Assume that $U(x)$ is a bounded solution to (17) as well, and there exists $y \in \mathbb{R}^n$ such that $U(y) \neq V(y)$. Without loss of generality, we assume that $U(y) < V(y)$, i.e. there exists $\delta > 0$ such that $V(y) - U(y) = \delta$.

Since $U(y) - \max_{j \in \{1, \ldots, n\}} h'_{0,j}(y) \geq 0,$
\[
V(y) - \max_{j \in \{1, \ldots, n\}} h'_{0,j}(y) > 0
\]
holds. Consequently, we have that $V(y) = \alpha \sup_{d \in D} V(f(y, d))$. Also, due to the fact that $U(y) \geq \alpha \sup_{d \in D} U(f(y, d))$,
\[
\alpha \sup_{d \in D} V(f(y, d)) - \alpha \sup_{d \in D} U(f(y, d)) \geq \delta
\]
holds, implying that
\[
\alpha \sup_{d \in D} (V(f(y, d)) - U(f(y, d))) \geq \delta.
\]
Therefore, for $1 < \beta < \frac{1}{\alpha}$, there exists $d \in D$ such that
\[
\alpha \sup_{d \in D} (V(f(y, d)) - U(f(y, d))) \geq \beta \alpha \delta.
\]
Let $d_1$ satisfy
\[
V(f(y, d_1)) - U(f(y, d_1)) \geq \beta \delta,
\]
and $y_1 = f(y, d_1)$. It is obvious that
\[
V(y_1) - U(y_1) \geq \beta \delta.
\]
Via repeating the above procedure, we can construct a sequence $\{y_j\}_{j=0}^\infty$ satisfying $V(y_j) - U(y_j) \geq \beta^j \delta$ for each $j$. Thus,
\[
V(y_j) \geq U(y_j) + \beta^j \delta, \forall j \in \mathbb{N}.
\]
Since $U$ is bounded over $\mathbb{R}^n$ and
\[
\lim_{j \to \infty} \beta^j \delta = \infty,
\]
we have that $V(y_j)$ approach infinity when $j$ tends to infinity. Therefore, $V(y) = U(y)$. Based on the above deduction technique with $U(y)$ and $V(y)$ reversed, a similar contradiction can be gained for the case that $U(y) > V(y)$.

Above all, we conclude that the value function $V(x) : \mathbb{R}^n \to \mathbb{R}$ in (17) is the unique bounded solution to (17).

(2). When $\mathcal{X} = \{X_i\}$ in $S_D$, where $i \in \{1, \ldots, k\}$, i.e. $X_i = \mathbb{R}^n$, and $\alpha \in (0, 1)$, the continuity property of $V(x) : \mathbb{R}^n \to \mathbb{R}$ is assured by Lemma 2. Uniqueness is guaranteed via (1). Therefore, $V(x) : \mathbb{R}^n \to \mathbb{R}$ in (17) is a unique bounded and Lipschitz continuous solution to (17) when $\alpha \in (0, 1)$. 

From Theorem 2 we obtain that the maximal robust invariant set $\mathcal{R}_0$ can be approximated by addressing (17). A technique for addressing (17) with $\alpha \in (0, 1)$ is the value iteration algorithm in the framework of reinforcement learning.
Theorem 3. Assume that the sequence of functions \( \{V_i\}_{i \in \mathbb{N}} \) is generated by the value iteration algorithm starting from some bounded function \( V_0 : \mathbb{R}^n \to \mathbb{R} \) according to

\[
V_{i+1}(x) = \sup_{d \in D} \max \left\{ \alpha V_i(f(x, d)), \max_{j \in \{1, \ldots, n_0\}} h_{0,j}(x) \right\}, \forall x \in \mathbb{R}^n, \forall i \in \mathbb{N},
\]

then \( V_i(x) \) uniformly approximates \( V(x) \) over \( \mathbb{R}^n \) if \( \alpha \in (0, 1) \) as \( i \) tends to infinity, where \( V(x) \) is the unique bounded solution to (17).

Proof. According to (18), we have

\[
\begin{align*}
V_{i+1}(x) - V_i(x) & \leq \sup_{d \in D} \max \left\{ \alpha V_i(f(x, d)), \max_{j \in \{1, \ldots, n_0\}} h_{0,j}(x) \right\} - \sup_{d \in D} \max \left\{ \alpha V_{i-1}(f(x, d)), \max_{j \in \{1, \ldots, n_0\}} h_{0,j}(x) \right\} \\
& \leq \sup_{d \in D} \max \{\alpha(V_i(f(x, d)) - V_{i-1}(f(x, d))), 0\} \\
& \leq \sup_{d \in D} \max \{\alpha^{i} V_i(f^{i}(x, d)) - V_{0}(f^{i}(x, d)), 0\},
\end{align*}
\]

and

\[
\begin{align*}
V_{i+1}(x) - V_i(x) & \geq - \inf_{d \in D} \min \left\{ -\alpha V_i(f(x, d)), - \max_{j \in \{1, \ldots, n_0\}} h_{0,j}(x) \right\} + \inf_{d \in D} \min \left\{ -\alpha V_{i-1}(f(x, d)), - \max_{j \in \{1, \ldots, n_0\}} h_{0,j}(x) \right\} \\
& \geq \inf_{d \in D} \min \{\alpha(V_i(f(x, d)) - V_{i-1}(f(x, d))), 0\} \\
& \geq \inf_{d \in D} \min \{\alpha^{i} V_i(f^{i}(x, d)) - V_{0}(f^{i}(x, d)), 0\},
\end{align*}
\]

where \( f^{i}(\cdot) = f \circ f \circ \cdots \circ f(\cdot) \). Moreover, since \( V_0 \) and \( \max_{j \in \{1, \ldots, n_0\}} h_{0,j} \) are bounded over \( \mathbb{R}^n \), therefore, \( V_1 \) is bounded as well. Consequently, according to (19), (20) and \( \alpha \in (0, 1) \), we have that \( V_i(x) \) uniformly approximates a function \( V^{'}(x) \) over \( \mathbb{R}^n \) as \( i \) tends to infinity. In the following we just need to prove that \( V^{'}(x) = V(x) \) over \( x \in \mathbb{R}^n \). This conclusion can be assured by replacing \( V_i \) in (19) and (20) with \( V \), resulting in the fact that \( V_i(x) \) uniformly approximates the function \( V(x) \) over \( \mathbb{R}^n \) as \( i \) tends to infinity. \(\square\)

The value iteration algorithm for addressing (17) with \( \alpha \in (0, 1) \) is presented as follows:

1. Set \( V_0(x) = 0 \) over \( x \in \mathbb{R}^n \) and \( l := 0 \); Decide on a grid \( \Lambda = \{x_1, \ldots, x_N\} \) on a compact set \( \overline{\mathcal{B}} \supseteq X_0 \) for the state variable \( x \), and decide on a grid \( \Delta = \{d_1, \ldots, d_M\} \) in \( D \) for the perturbation variable \( d \).

2. Formulate an initial guess for the value function \( V_0(x) \) and choose a stopping criterion \( \epsilon > 0 \);

3. For each \( x \in \Lambda, i = 1, \ldots, N \), compute

\[
x_{i,j}^{'} = f(x_i, d_j), \forall j = 1, \ldots, M,
\]

then compute a interpolated value function at each \( x_{i,j}^{'} \) and compute

\[
V_{i+1}(x_i) = \max_{d \in \Delta} \max \left\{ \alpha V_i(x_i), \max_{j \in \{1, \ldots, n_0\}} h_{0,j}(x_i) \right\}.
\]

4. If \( \max_{x \in \Lambda} |V_{i+1}(x) - V_i(x)| < \epsilon \), go to step 5; otherwise, \( l := l + 1 \) and go back to 2);

5. Compute the final solution as \( V(x) \approx V_{i+1}(x) \).

The termination of the value iteration algorithm is guaranteed by Theorem 3. In this way the value of \( V(x) \) can be calculated on the grid points of the set \( \overline{\mathcal{B}} \). However, this is a computationally expensive process and as the size of the state and perturbation spaces grows, it becomes intractable.

Remark 1. When \( \alpha = 1 \), we cannot guarantee the convergence of \( \{V_k\}_{k \in \mathbb{N}} \) in (18). Even if the sequence \( \{V_k\}_{k \in \mathbb{N}} \) converges, it is not guaranteed that \( \lim_{k \to \infty} V_k(x) = V(x) \), where \( V(x) \) is the value function in (4), since \( \{V_k\}_{k \in \mathbb{N}} \) may not have a unique solution.
3.2 Semi-definite Programming Implementation

When the dimension of the product space of state and perturbation spaces is appropriate, value iteration methods could be employed to solve (17), rendering possible the gain of an approximation of the maximal robust invariant set. However, the limitations of such methods are quite strict due to the curse of dimensionality, consequently highlighting the need of alternative approximation methods. We in this section present a semi-definite programming based method to synthesize robust invariant sets.

From (17) we observe that if a continuous function \( u(x) : \mathbb{R}^n \to \mathbb{R} \) satisfies (17), then it satisfies

\[
\begin{cases}
  u(x) - \alpha u(f(x, d)) \geq 0, \forall d \in D, \forall x \in \mathbb{R}^n, \\
  u(x) - h_{0,j}(x) \geq 0, \forall x \in \mathbb{R}^n, \forall j \in \{1, \ldots, n_0\}.
\end{cases}
\]

\( (21) \)

**Corollary 1.** For any function \( u(x) : \mathbb{R}^n \to \mathbb{R} \) satisfying (21), \( \{x \in \mathbb{R}^n \mid u(x) \leq 0\} \) is a robust invariant set when \( \alpha \in (0, 1] \). Furthermore, if \( \alpha \in (0, 1) \), \( u(x) \geq 0 \) over \( x \in \mathbb{R}^n \) and consequently

\[\{x \in \mathbb{R}^n \mid u(x) \leq 0\} = \{x \in \mathbb{R}^n \mid u(x) = 0\} .\]

**Proof.** Its proof is presented in Appendix.

Lemma 4 \( (22) \). Suppose \( f'(x) = 1_{F_1} \cdot f'_1(x) + \cdots + 1_{F_{k'}} \cdot f'_{k'}(x) \) and \( g'(x) = 1_{G_1} \cdot g'_1(x) + \cdots + 1_{G_{l'}} \cdot g'_{l'}(x) \), where \( x \in \mathbb{R}^n, k', l' \in \mathbb{N} \), and \( F_i, G_j \subseteq \mathbb{R}^n, i = 1, \ldots, k', j = 1, \ldots, l' \). Also, \( F_1, \ldots, F_{k'} \) and \( G_1, \ldots, G_{l'} \) are respectively disjoint. Then, \( f' \leq g' \) if and only if (pointwise)

\[
\bigwedge_{i=1}^{k'} \bigwedge_{j=1}^{l'} \left[ F_i \land G_j \Rightarrow f'_i \leq g'_j \right] \land \bigwedge_{i=1}^{k'} \left[ F_i \land \bigwedge_{j=1}^{l'} \neg G_j \Rightarrow f'_i \leq 0 \right] \land \bigwedge_{j=1}^{l'} \left[ \bigwedge_{i=1}^{k'} \neg F_i \land G_j \Rightarrow 0 \leq g'_j \right].
\]

\( (22) \)

Consequently, according to Lemma 4, the equivalent constraint without indicator functions of (21) is formulated below:

\[
\bigwedge_{i=1}^{k} \left[ u(x_0) - \alpha u(f(x_0, d)) \geq 0, \forall d \in D, \forall x_0 \in X_1 \right] \land \\
\bigwedge_{j=1}^{n_0} \left[ u(x_0) - h_{0,j}(x_0) \geq 0, \forall x_0 \in \mathbb{R}^n \right].
\]

\( (23) \)

We define the set \( \Omega(X_0) \) of states being reachable from the set \( X_0 \) within one step computation, i.e.,

\[\Omega(X_0) := \{x \mid x = f(x_0, d), x_0 \in X_0, d \in D\} \cup X_0,\]

which can be obtained by solving semi-definite programs or linear programs [16] [21]. Herein, we assume that it was already given. In addition, we define \( \sum[y] \) to be the set of sum of squares (SOS) polynomials over variables \( y \), i.e.,

\[\sum[y] := \{p \in \mathbb{R}[y] \mid p = \sum_{i=1}^{k} q_i^2, q_i \in \mathbb{R}[y], i = 1, \ldots, k\}.\]
Therefore, when employing the semi-definite program (24), the state spaces for Example 1, 2 and 3 are respectively restricted to \( B \) as a semi-definite programming solver. For the value iteration method, uniform grids are adopted for state and control the performance of our methods are presented in Table 1. All computations were performed on an i7-7500U processor with 32GB RAM running Windows 10. For numerical implementation of the semi-definite program (24), we formulate the sum of squares program (24) using the MATLAB package YALMIP\(^2\) and use Mosek\(^3\) as a semi-definite programming solver. For the value iteration method, uniform grids are adopted for state and perturbation spaces. The state spaces for Example 1, 2 and 3 are respectively restricted to \([-1,1] \times [-1,1], [-1,1] \times [-1,1]\) and \([-1,1] \times [-1,1]\).

Example 1. In this example, \( f_1(x, y) = (0.4x + 0.6y; dx + 0.9y) \), \( X_0 = \{(x, y) \mid x^2 + y^2 \leq 1\} \) and \( D = \{d \mid d^2 - 0.01 \leq 0\} \), that is, no switching occurs in this system. The inner-approximations of the maximal robust invariant set \( R_0 \) when \( d_a = 10 \) and \( d_a = 12 \) are illustrated in Fig. 4. Fig. 2 also presents the estimated maximal robust invariant set, which is computed by solving (17) based on the value iteration method. The level sets of the corresponding computed solution are illustrated in Fig. 4.

\(^2\)It can be downloaded from https://yalmip.github.io/

\(^3\)For academic use, the software Mosek can be obtained free from https://www.mosek.com/
Table 1: Parameters and performance of our implementations on the examples presented in this section. \( \alpha \): the parameter value in (4); \( d_u, d_s, d_s' \): the degree of the polynomials \( u, s_i, s^X, s^D \) in (24), respectively, \( i = 1, \ldots, k \); \( l_1 = 1, \ldots, n_i \); \( l_2 = 1, \ldots, n_{k+1} \); \( j = 1, \ldots, n_0 \); \( T_{SDP} \): computation times (seconds) in solving (24); \( \epsilon \): the stopping criterion in the value iteration method; \( N, M \): numbers of elements in \( \Lambda \) and \( \Delta \) respectively in the value iteration method; \( T_{VI} \): computation times (seconds) in solving (17) using value iteration methods.

![Figure 1: Level sets of \( V \) obtained via value iteration methods for Example 1.](image1)

**Example 2.** In this example we consider SD with \( f_1(x, y) = (x; (0.5+d)x - 0.1y) \), \( f_2(x, y) = (y; 0.2x - (0.1+d)y + y^2) \), \( X_0 = \{(x, y) \mid x^2 + y^2 - 0.8 \leq 0\} \), \( X_1 = \{(x, y) \mid 1 - (x-1)^2 - y^2 \leq 0\} \), \( X_2 = \{(x, y) \mid -1 + (x-1)^2 + y^2 < 0\} \) and \( D = \{d \mid d^2 - 0.01 \leq 0\} \). The inner-approximations computed by solving (24) when \( d_u = 8, 10 \) and 12 respectively are illustrated in Fig. 4. Fig. 4 also presents the maximal robust invariant set, which is computed by solving (17) via the value iteration method. The level sets of the corresponding computed solution is visualized in Fig. 3.

The level sets displayed in Fig. 1 and Fig. 3 further confirm that the solution to (17) with \( \alpha \in (0, 1) \) is non-negative, as stated in Theorem 1. We apply the semi-definite program (24) with \( \alpha = 0.5 \) to Examples 1 and 2 as well. However, we did not obtain non-empty robust invariant sets for both examples based on the parameters in Table 1. This justifies Remark 2.

The semi-definite programming based method with polynomials of appropriate degree, i.e. (24), can increase the computational efficiency in computing robust invariant sets, compared with the value iteration method. On the other side, the semi-definite programming based method brings conservativeness in estimating the maximal robust invariant sets. This effect can be observed from the visualized results in Fig. 2 and 4. Although the value iteration method can solve the Bellman equation (17) with \( \alpha \in (0, 1) \) and produce an approximation of the maximal robust invariant set, they require partitioning the state and perturbation spaces, thereby exhibiting exponential growth in
computational complexity with the number of state and perturbation variables and preventing their application for higher dimensional systems. As opposed to grid-based numerical methods, the semi-definite programming based method (24) falls within the convex programming framework and can be applied to systems with moderately high dimensionality. We illustrate this issue through an example with seven state variables.

**Example 3.** In this example, we consider SD with seven dimensional variables \( \mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \) and illustrate the scalability of our approach. In PS, \( f_1(\mathbf{x}) = ((0.5 + d)x_1; 0.8x_2; 0.6x_3 + 0.1x_6; x_4; 0.8x_5; 0.1x_2 + x_6; 0.2x_2 + 0.6x_7); \) \( f_2(\mathbf{x}) = (0.5x_1 + 0.1x_6; (0.5 + d)x_2; x_3; 0.1x_1 + 0.4x_4; 0.2x_1 + x_5; x_6; 0.1x_1 + x_7) \), \( X = \{ \mathbf{x} \mid \sum_{i=1}^{7} x_i^2 - 1 \geq 0 \} \), \( X_1 = \{ \mathbf{x} \mid x_1 + x_2 + x_3 - x_4 - x_5 - x_6 - x_7 \geq 0 \} \), \( X_2 = \{ (x,y) \mid x_1 + x_2 + x_3 - x_4 - x_5 - x_6 - x_7 < 0 \} \) and \( D = \{ d \mid d^2 - 0.01 \leq 0 \} \).

Plots of the computed robust invariant sets are illustrated in Fig. 6. Unlike for the low-dimensional Examples 1 and 2, the value iteration method for solving (17) here runs out of memory and thus does not return an estimate. The semi-definite programming based method (24), however, is still able to compute robust invariant sets, which are illustrated in Fig. 6. In order to shed light on the accuracy of the computed robust invariant sets, we synthesize

Figure 2: Computed robust invariant set for Example 1. (Blue and Green curves – boundaries of computed robust invariant sets when \( d_u = 10 \) and \( d_u = 12 \), respectively; Blue points – the maximal robust invariant set via numerical simulation techniques; Red curve – boundary of the computed maximal robust invariant set via value iteration methods ; Black curve – the boundary of \( X_0 \).)

Figure 3: Level sets of \( V \) obtained via value iteration methods for Example 2.
Figure 4: Computed robust invariant sets for Example 2. Black, Blue, Purple and Green curves: boundaries of $X_0$ and computed robust invariant sets when $d_u = 8, 10, 12$, respectively; Red curve: boundary of the computed maximal robust invariant set via value iteration methods; Blue points – the approximated maximal robust invariant set via numerical simulation techniques.

Coarse estimates of the maximal robust invariant sets on planes $x_5 - x_6$ with $x_1 = x_2 = x_3 = x_4 = x_7 = 0$ and $x_6 - x_7$ with $x_1 = x_2 = x_3 = x_4 = x_5 = 0$ respectively. These estimates are also depicted in Fig. 6.

Besides, we compare the methods in the present paper with the ones in [33]. We firstly compare the value iteration method. Based on the same parameter inputs listed in Table 1 the value iteration method to solve the Bellman equation in [33] does not terminate after one hour for Examples 1 and 2. The underlying reason is that the value iteration method for solving Bellman equation in [33] does not converge. Consequently, it may not be used to compute an approximation of the maximal robust invariant set. The value iteration method for solving the Bellman equation in [33] still runs out of memory and thus does not return an estimate. Next, we compare the semi-definite programming based method (24) with the one in [33] based on the same parameters in Table 1. The results obtained by these two methods for Examples 1, 2 and 3 are respectively visualized in Fig. 6, 7 and 8. We from Fig. 6 observe that almost the same inner-approximations are produced for Example 1 by these two methods. Fig. 7 indicates that (24) produces a more conservative estimation of the maximal robust invariant set than the one in [33] in the case of $d_u = 8$ for Example 2 however, it gives less conservative estimations when $d_u = 10$ and $d_u = 12$. However, Fig. 8 shows that (24) produces a more conservative estimation of the maximal robust invariant set than the one in [33].

5 Conclusion and Future Work

In this paper we systematically studied the problem of computing robust invariant sets for switched discrete-time polynomial systems subject to state constraints. We for the first characterize the maximal robust invariant set as the zero level set of the unique bounded solution to a modified Bellman functional equation. Existing well-developed numerical methods can be used to solve such equation with appropriate number of state variables. Furthermore, based on the derived Bellman equation we construct a semi-definite program to synthesize robust invariant sets. Three examples demonstrated the performance of our methods.

We will extend our method to the computation of robust invariant sets for the state-constrained perturbed hybrid systems subject to control inputs in our future work.

References

[1] C. O. Aguilar and A. J. Krener. Numerical solutions to the bellman equation of optimal control. Journal of optimization theory and applications, 160(2):527–552, 2014.
Figure 5: Computed robust invariant sets for Example 3. Black curve: the boundary of the computed robust invariant set by solving (24) when $d_u = 4$. Gray region is the estimate of the maximal robust invariant set via simulation techniques.

[2] N. Athanasopoulos and G. Bitsoris. Invariant set computation for constrained uncertain discrete-time linear systems. In Decision and Control (CDC), 2010 49th IEEE Conference on, pages 5227–5232. IEEE, 2010.

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

[4] M. Bardi and I. Capuzzo-Dolcetta. Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Springer Science & Business Media, 1997.

[5] F. Blanchini and S. Miani. Set-theoretic methods in control. Springer, 2008.

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

[7] Y.-F. Chen, C.-D. Hong, B.-Y. Wang, and L. Zhang. Counterexample-guided polynomial loop invariant generation by lagrange interpolation. In International Conference on Computer Aided Verification, pages 658–674. Springer, 2015.

[8] G. Chesi. Domain of attraction: analysis and control via SOS programming, volume 415. Springer Science & Business Media, 2011.
Figure 6: Robust invariant sets for Example 1 computed when $d_u = 10, 12$ (from left to right). Blue and Red curves: boundaries of robust invariant sets computed via (24) and the semi-definite programming based method in [33].

Figure 7: Robust invariant sets for Example 2 computed when $d_u = 8, 10, 12$ (from left to right). Blue and Red curves: boundaries of robust invariant sets computed via (24) and the semi-definite programming based method in [33].

[9] M. Falcone and R. Ferretti. Numerical methods for Hamilton-Jacobi type equations. In Handbook of Numerical Analysis, volume 17, pages 603–626. Elsevier, 2016.

[10] M. Fiacchini, T. Alamo, and E. F. Camacho. On the computation of convex robust control invariant sets for nonlinear systems. Automatica, 46(8):1334–1338, 2010.

[11] P. Giesl and S. Hafstein. Review on computational methods for lyapunov functions. Discrete and Continuous Dynamical Systems-Series B, 20(8):2291–2331, 2015.

[12] P. Grieder, S. Raković, M. Morari, and D. Mayne. Invariant sets for switched discrete time systems subject to bounded disturbances. IFAC Proceedings Volumes, 38(1):115–120, 2005.

[13] Z. W. Jarvis-Wloszek. Lyapunov based analysis and controller synthesis for polynomial systems using sum-of-squares optimization. PhD thesis, University of California, Berkeley, 2003.
Figure 8: Robust invariant sets for Example 3 computed when $d_u = 4$. Blue and Red curves: boundaries of robust invariant sets computed via (24) and the semi-definite programming based method in [33].

[14] J. Kapinski and J. Deshmukh. Discovering forward invariant sets for nonlinear dynamical systems. In Interdisciplinary Topics in Applied Mathematics, Modeling and Computational Science, pages 259–264. Springer, 2015.

[15] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. Mathematical problems in engineering, 4(4):317–367, 1998.

[16] J. B. Lasserre. Tractable approximations of sets defined with quantifiers. Mathematical Programming, 151(2):507–527, 2015.

[17] Y. Li and J. Liu. Invariance control synthesis for switched nonlinear systems: An interval analysis approach. IEEE Transactions on Automatic Control, 63(7):2206–2211, 2018.

[18] Y. Li and J. Liu. Rocs: A robustly complete control synthesis tool for nonlinear dynamical systems. In Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week), pages 130–135. ACM, 2018.

[19] J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In Computer Aided Control Systems Design, 2004 IEEE International Symposium on, pages 284–289. IEEE, 2004.

[20] C. K. Luk and G. Chesi. On the estimation of the domain of attraction for discrete-time switched and hybrid nonlinear systems. International Journal of Systems Science, 46(15):2781–2787, 2015.
Appendix

The proof of Corollary 1.

Proof. The statement that \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} \) is a robust invariant set when \( \alpha \in (0, 1) \) can be justified by following the proof of Corollary 1 in [33].

In the sequel we prove that \( u(x) \geq 0 \) over \( x \in \mathbb{R}^n \) when \( \alpha \in (0, 1) \). Assume that there exists \( y \in \mathbb{R}^n \) such that \( u(y) < 0 \). Due to the fact that

\[
u(f(y, d)) \leq \frac{1}{\alpha} u(y), \forall d \in D,
\]

we obtain that

\[
u(f^i(y, d)) \leq \frac{1}{\alpha^i} u(y), \forall d \in D,
\]

where \( f^i(\cdot) = f \circ f \circ \cdots \circ f(\cdot) \) is the iterated function. Therefore, we have that

\[
\lim_{i \to \infty} u(f^i(y, d)) = -\infty,
\]

6 Appendix

The proof of Corollary 1.

Proof. The statement that \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} \) is a robust invariant set when \( \alpha \in (0, 1) \) can be justified by following the proof of Corollary 1 in [33].

In the sequel we prove that \( u(x) \geq 0 \) over \( x \in \mathbb{R}^n \) when \( \alpha \in (0, 1) \). Assume that there exists \( y \in \mathbb{R}^n \) such that \( u(y) < 0 \). Due to the fact that

\[
u(f(y, d)) \leq \frac{1}{\alpha} u(y), \forall d \in D,
\]

we obtain that

\[
u(f^i(y, d)) \leq \frac{1}{\alpha^i} u(y), \forall d \in D,
\]

where \( f^i(\cdot) = f \circ f \circ \cdots \circ f(\cdot) \) is the iterated function. Therefore, we have that

\[
\lim_{i \to \infty} u(f^i(y, d)) = -\infty,
\]
Consequently, \( u(x) \geq 0 \) for \( x \in \mathbb{R}^n \) when \( \alpha \in (0, 1) \). Therefore, \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} = \{ x \in \mathbb{R}^n \mid u(x) = 0 \} \) when \( \alpha \in (0, 1) \). An immediate result is \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} = \{ x \in \mathbb{R}^n \mid u(x) = 0 \} \).

The proof of Theorem 4:

**Proof.** Since \( u(x) \) satisfies the constraint in (24) and \( f(x, d) \) satisfies (2), we obtain that \( u(x) \) satisfies according to \( S^- \) procedure presented in [6]:

\[
\begin{align*}
u(x) - \alpha u(f(x, d)) &\geq 0, \forall d \in D, \forall x \in B \quad \text{(25)} \\
(1 + h^2_{0, j}(x))u(x) - h_{0, j}(x) &\geq 0, \forall j \in \{1, \ldots, n_0\}.
\end{align*}
\]

Assume that there exist an initial state \( y \in \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} \) and an input policy \( \pi' \) such that \( \phi_{\pi'}(l) \notin X_0 \) for every \( l \in \mathbb{N} \). Since (26) holds, we have the conclusion that \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} \subset X_0 \) and thus \( y \in X_0 \). Let \( l_0 \in \mathbb{N} \) be the first time making \( \phi_{\pi'}(l) \) outside the set \( X_0 \), i.e.

\[
\phi_{\pi'}(l_0) \notin X_0 \text{ and } \phi_{\pi'}(l) \in X_0
\]

for \( l = 0, \ldots, l_0 - 1 \). That is, \( u(\phi_{\pi'}(l_0)) > 0 \). However, since \( \Omega(X_0) \subset B \), (25) and (26), we derive that

\[
u(\phi_{\pi'}(l_0)) \leq 0.
\]

This is a contradiction. Thus, every possible trajectory to \( S^- \) initialized in \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} \) will live in \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} \) forever. Therefore, \( \{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} \) is a robust invariant set.

When \( \alpha \in (0, 1) \), using analogous arguments as in the proof of Corollary 4, we obtain \( u(x) \geq 0 \) over \( x \in B \) and as a result we have

\[
\{ x \in \mathbb{R}^n \mid u(x) \leq 0 \} = \{ x \in \mathbb{R}^n \mid u(x) = 0 \}.
\]