

RESEARCH

Open Access



The computation of the viability kernel for switched systems

Jianfeng Lv^{1,2} and Yan Gao^{1*} 

*Correspondence:

gaoyan@usst.edu.cn

¹School of Management, University of Shanghai for Science and Technology, Shanghai, China
Full list of author information is available at the end of the article

Abstract

The computation of the viability kernel provides the guarantee for the security evolution of the systems. In this paper, we focus on the computation of the viability kernel for discrete-time and continuous-time switched systems. A connection between the backward reachable set and the viability kernel for switched systems is established. The methods of computing the viability kernel for switched systems are constructed by using this connection. First, a method of computing the viability kernel for discrete-time switched systems is proposed. Then, taking into account the special structure of switched linear systems, a simple algorithm that is easy to implement is developed. Moreover, the methods of dealing with the discrete systems are extended to the continuous systems, and the algorithms of computing the viability kernel for continuous-time switched systems and switched linear systems are proposed. Finally, examples are listed to illustrate the effectiveness of the main results.

Keywords: Switched systems; Control; Viability kernel; Reachable set

1 Introduction

The problem of viability [1] concerns the dynamic evolutions governed by complex systems under uncertainty that are found in many domains involving living beings, from biological evolution to economics [2], from environmental sciences to financial markets [3, 4], from control theory to cognitive sciences [5–7]. It aims at controlling dynamical systems with the goal to maintain the state of the systems inside a given set which we called the constraint set or safe region. Determining the viability on a constraint set has been researched in the references [8–12]. When faced with the constraint set that is not viable, one would like to establish a subset of the constraint set that is viable. This subset is said to be the viability kernel (or the maximal invariant set).

Computation of the viability kernel for dynamical systems is a fundamental problem in the viability theory. It has traditionally been computed using the viability kernel algorithm [13] and level set approach [14]. However, these methods require gridding the state space, and hence their time and memory complexity grow exponentially with the state dimension. Thus, these methods are feasible only for dynamical systems with low dimension. To overcome this limitation, researchers considered the same problem by using dynamic programming [15] and support vector machines [16]. Recently, Maidens in [17] proposed an algorithm of computing the viability kernel by using backward reachable set for dynamical

systems. Since there are some efficient techniques for computing the maximal reachable set in [18, 19], these methods can be used to compute the viability kernel [17].

Switched systems, which consist of two or more subsystems and a switching signal orchestrating switching between these subsystems, have attracted a growing interest in recent years. The viability kernel for switched systems has been largely studied, and several results have been reported in [20–28]. Aubin and Lygeros introduced a notion of hybrid strategy and proved convergence of the iterative algorithm by using non-smooth analysis tools in [23]. Margellos and Lygeros proposed a method of computing the viable set for hybrid systems [24] in the context of optimal control, and a complete characterization of the computation of the viable set was provided based on dynamic programming. Haimovich et al. in [25] also developed a method of computing the invariant set for continuous-time switched linear systems with disturbances and arbitrary switching. Abate and Prandini in [26] studied probabilistic reachability over a finite horizon for a class of discrete time stochastic hybrid systems. Lygeros et al. in [27] presented some simple properties of the invariant sets for hybrid automata, and the invariance principle was extended to hybrid automata. By establishing a link between reachability, viability, and invariance problems and viscosity solutions of a special form of the Hamilton–Jacobi equation, the numerical algorithm developed for the approximation of viscosity solutions to partial differential equations was extended to viability and invariance computation in [28].

It should be noted that all the methods in [23–25, 28] have certain limitations when they are executed. The first is that many conditions must be satisfied in the implementation of the algorithms. In order to guarantee the convergence of these algorithms, we need to compute the maximal fixed point of a monotone operator on a complete lattice of closed sets in [24] and determine the transformation matrix satisfying certain properties in [25]. The given set must be open and the partial differential equation must have a special form (standard Hamilton–Jacobi form, continuity of the Hamiltonian, and simple boundary conditions) in [28]. Thus, the convergence conditions are too strong. Moreover, because of the large amount of computation and considerable time consumption, the implementation of the algorithms is difficult. To overcome these limitations, we restrict our attention to switched linear systems. Since they have a special structure, the limitations of [23–25, 28] can be solved effectively. Thus, we try to design a simple method of computing the viability kernel for switched linear systems.

The organization of the paper is as follows. In Sect. 2 we introduce some necessary preliminaries. Section 3 discusses the computation of the viability kernel for discrete-time switched systems and develops a simple algorithm for discrete-time switched linear systems. Furthermore, we compute the viability kernel for continuous-time switched systems and propose an algorithm for switched linear systems in Sect. 4. We provide several examples in Sect. 5.

2 Preliminaries

Consider the following switched system:

$$\begin{cases} \mathcal{L}(x(t)) = f_{\sigma(t)}(x(t), u(t)), \\ u(t) \in U, \end{cases} \quad (2.1)$$

where the state $x \in \mathbb{R}^n$, the control $u \in U$, $U \subset \mathbb{R}^m$, the function $f_{\sigma} : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$, and the time t ranges over a time domain T , which can be either continuous ($T = [0, \tau] \cap \mathbb{R}_+$)

or discrete ($T = [0, \tau] \cap \mathbf{Z}_+$). If $0 < \tau < \infty$, problem (2.1) is said to have a finite horizon; if $\tau = \infty$, it is said to have an infinite horizon. $\mathcal{L}(\cdot)$ denotes the derivative operator for continuous-time switched systems or the difference operator for the case of discrete-time. $\sigma(t) : [t_0, +\infty) \rightarrow \Lambda$ is a switching signal that is a piecewise constant function of time t and takes values at the sampling times in a finite set $\Lambda = \{1, \dots, N\}$, where $N > 1$ is the number of subsystems. $\sigma(t) = i$ ($i \in \Lambda$) denotes that the subsystem f_i is activated. We assume that the functions f_i ($i \in \Lambda$) are sufficiently smooth to guarantee the existence and uniqueness of solutions to the corresponding initial value problems. We refer to the sequence t_0, t_1, \dots, t_s as a switching time sequence, and the sequence $\sigma(t_0), \sigma(t_1), \dots, \sigma(t_s)$ as a switching index sequence. It is clear that these two sequences can uniquely determine the switching path, and vice versa.

In order to discuss the method of computing the viability kernel conveniently, we present some important features of switched systems. Switched systems are dynamical systems that consist of a finite number of subsystems and a logical rule that orchestrates switching between these subsystems. Therefore, one of the features of switched systems is multi-subsystem. In addition, switched systems could not be guaranteed to be stable even if all subsystems are stable. Switching rule may be designed to stabilize switched systems even if all subsystems are unstable. According to the discussion above, the computation of the viability kernel is affected by the switching rules and subsystems.

We start with some notions, which will be used later on.

Definition 1 Let $K \subset \mathbf{R}^n$. The viability kernel of system (2.1) on the set K over the time horizon T is defined as follows:

$$\text{Viab}_T(K) = \{x_0 \in K \mid \exists u_0 : T \rightarrow U, \forall t \in T, x(t) \in K\}. \quad (2.2)$$

Obviously, $\text{Viab}_T(K)$ contains all initial states in K for which there exists an input such that the trajectories starting from those states remain within K for all time $t \in T$.

Definition 2 The backward reachable set from K at time t is defined as follows:

$$\text{Reach}_t(K) = \{x_0 \in \mathbf{R}^n \mid \exists u_0 \in U, x(t) \in K\}. \quad (2.3)$$

In fact, the backward reachable set contains all initial states for which there exists an input such that the trajectories emanating from those states reach K exactly at time t .

The viability theory is concerned with ensuring that the state of a system remains within the given set for the duration of a known (possibly infinite) time horizon. Any trajectory of a system that leaves the given set at some point in time is considered to be no longer viable. The reachability analysis provides us a method of simulating all possible trajectories of a dynamical system under all possible inputs. In fact, the reachability is concerned with determining if any trajectories of a system that begin in a set of initial conditions can reach a set of terminal states. Both of them study the evolution of dynamical systems under the inputs and the state constraints. Thus, there is a close connection between the viability theory and the constrained reachability. The connection was used to compute the viability kernel for dynamical systems in [17]. In the rest of the paper, we extend the method to switched systems for approximating the viability kernel.

3 The viability kernel of discrete-time switched systems

In this section, we establish a connection between the viability kernel and the reachable set for discrete-time switched systems. Then, we discuss algorithms of computing the viability kernel in terms of the reachable set.

3.1 Discrete-time switched systems

We consider the discrete-time switched systems as follows:

$$x(t + 1) = f_{\sigma(t)}(x(t), u(t)), \tag{3.1}$$

where the time $t \in T = [0, \tau] \cap \mathbf{Z}_+$ and switching signal $\sigma \in \Lambda = \{1, \dots, N\}$. We next discuss how to compute the viability kernel on the given region K by using the maximal reachable set. As we know, the viability kernel for discrete-time dynamical systems can be computed using Saint-Pierre’s viability kernel algorithm (see [13]). In the light of Saint-Pierre’s viability kernel algorithm, we give an iterative formula of the viability kernel $K_{n+1} := \text{Viab}_T(K)$ for system (3.1) on the finite horizon $T = [0, n + 1] \cap \mathbf{Z}_+$:

$$\begin{cases} K_0 = K, \\ K_{n+1} = \{x \in K_n \mid K_n \cap F(x) \neq \emptyset\}, \end{cases} \tag{3.2}$$

where $F(x) = \{f_{\sigma}(x, u) \mid u \in U, \sigma \in \Lambda\}$. We replace the finite horizon viability kernel K_n in (3.2) by the backward reachable set. Then we obtain the following theorem.

Theorem 1 *The viability kernel over a finite horizon $T = [0, n + 1] \cap \mathbf{Z}_+$ for system (3.1) on the set K can be computed by the following recursive formula:*

$$\begin{cases} K_0 = K, \\ K_{n+1} = K_0 \cap \text{Reach}_1(K_n). \end{cases} \tag{3.3}$$

Proof System (3.1) can be written as the difference inclusion

$$x(t + 1) \in F(x(t)), \quad t = 0, 1, \dots, n,$$

where $F(x) = \{f_{\sigma}(x, u) \mid u \in U, \sigma \in \Lambda\}$. First, we prove the following formula:

$$\text{Viab}_T(K) = K_n \cap \text{Reach}_1(K_n). \tag{3.4}$$

By the definition of $\text{Viab}_T(K)$ and (3.2), we have $x \in K_n$ and $K_n \cap F(x) \neq \emptyset$ when $x \in \text{Viab}_T(K)$. It means that there exists y such that $y \in K_n$ and $y \in F(x)$. Then there exist $u \in U$ and $\sigma \in \Lambda$ such that $y = f_{\sigma}(x, u)$. Thus, we have $u \in U$ and $\sigma \in \Lambda$ such that $f_{\sigma}(x, u) \in K_n$. According to (2.3) and $x \in K_n$, we have $x \in \text{Reach}_1(K_n)$. Moreover, we obtain that $x \in K_n \cap \text{Reach}_1(K_n)$. Similarly, we concluded that $x \in \text{Viab}_T(K)$ form $x \in K_n \cap \text{Reach}_1(K_n)$. This means that

$$\text{Viab}_T(K) = K_n \cap \text{Reach}_1(K_n).$$

Therefore, we only need to prove that

$$K_n \cap \text{Reach}_1(K_n) = K_0 \cap \text{Reach}_1(K_n).$$

According to (3.4), we have

$$K_{n+1} = K_n \cap \text{Reach}_1(K_n),$$

which implies that

$$K_{n+1} \subseteq K_n \subseteq \dots \subseteq K_1 \subseteq K_0$$

and

$$\text{Reach}_1(K_n) \subseteq \text{Reach}_1(K_{n-1}) \subseteq \dots \subseteq \text{Reach}_1(K_1) \subseteq \text{Reach}_1(K_0).$$

Thus, we get

$$\begin{aligned} K_{n+1} &= K_0 \cap \text{Reach}_1(K_0) \cap \text{Reach}_1(K_1) \cap \dots \cap \text{Reach}_1(K_n) \\ &= K_0 \cap \text{Reach}_1(K_n). \end{aligned}$$

This completes the proof of the theorem. □

Theorem 1 gives us a method of constructing a finite horizon viability kernel for discrete-time switched systems (3.1). In (3.3), we compute the viability kernel by $K_0 \cap \text{Reach}_1(K_n)$ instead of $K_n \cap \text{Reach}_1(K_n)$, though they are equivalent when the reachable set and the operation of intersections can be computed exactly. In fact, K_0 is simpler and more precise in implementation than K_n . Thus, we construct the viability kernel by recursive formula (3.3). In what follows, the key problem we need to solve is the computation of the reachable set for switched systems.

The computation of the reachable set for dynamical systems is given in [17]. The reachable states of dynamical systems are affected by the system and the control input. Since switched systems have many subsystems, the trajectories of switched systems are restricted by multiple subsystems and the reachability is also affected by the switching rules. Different switching rules lead to different reachable states. Thus, the computation of the reachable set for switched systems is more complex than for dynamical systems. We assume that the switching rules are arbitrary switches. In this case, we need to compute the possible reachable region for every subsystem. Then the reachable set for switched systems is obtained by taking the union of all the reachable regions.

We next illustrate the computation process of the reachable set $\text{Reach}_1(K)$ for system (3.1) on the set K over a unit time step in detail. First, we need to compute the reachable set of K for every subsystem $f_\sigma(x, u)$, $\sigma \in \Lambda$, and the obtained set is denoted by $\text{Reach}_1^\sigma(K)$, $\sigma \in \Lambda$. So we get N reachable sets. Then the set $\text{Reach}_1(K)$ can be got by intersecting these N reachable sets

$$\text{Reach}_1(K) = \bigcup_{i=1}^N \text{Reach}_1^i(K).$$

Algorithm 1 The viability kernel of discrete-time switched systems

```

Let  $K_0 \leftarrow K$ 
 $l \leftarrow 0$ 
while  $l \leq n$  do
  if  $K_l = \emptyset$  then
     $K_n \leftarrow \emptyset$ 
    break
  endif
  if  $K_l = K_{l-1}$  then
     $K_n \leftarrow K_l$ 
    break
  endif
   $L \leftarrow \bigcup_{i=1}^N \text{Reach}_1^i(K_l)$ 
   $K_{l+1} \leftarrow K_0 \cap L$ 
   $l \leftarrow l + 1$ 
end while
return  $K_n$ 

```

The amount of the computation is proportional to the number of the subsystems. When the number of the subsystems is higher, the amount of the computation is greater.

According to Theorem 1, an algorithm of computing the viability kernel for system (3.1) over a finite horizon $T = [0, n] \cap \mathbf{Z}_+$ is proposed (see Algorithm 1).

Algorithm 1 transforms the computation of the viability kernel on a finite horizon into the computation of the reachable set for finite times. The main advantages of Algorithm 1 are as follows: The first is that the key problem in the iteration is the computation of the reachable set. Many researchers are devoted to the computation of the reachable set for dynamical systems, and many methods of computing the reachable set are proposed. Since the methods of computing the reachable set for dynamical systems are extended to switched systems, Algorithm 1 is available in practice. The second is that the set obtained from Algorithm 1 is the exact viability kernel of system (3.1) on the given set.

3.2 Discrete-time switched linear systems

Algorithm 1 can be applied to switched nonlinear systems in theory. However, the computation process of nonlinear systems is rather complex and troublesome. The complexity of computing the reachable set lies in the part of nonlinear continuous evolution. Thus, we consider the viability kernel for switched linear systems. Due to the special structure of linear systems, which have some special properties, we discuss a simple algorithm of computing the viability kernel for switched linear systems. In this subsection, we consider the following discrete-time switched linear systems:

$$x(t+1) = A_\sigma x(t) + B_\sigma u(t), \quad (3.5)$$

where the state $x \in \mathbf{R}^n$, the time $t \in T = [0, n] \cap \mathbf{Z}_+$, the control input $u \in U$, U is a compact set, and switching signal σ takes values in a finite set $\Lambda = \{1, \dots, N\}$. $A_\sigma \in \mathbf{R}^{n \times n}$ is an invertible matrix and $B_\sigma \in \mathbf{R}^{n \times m}$. As we know, any region can be approximated by a polytope. Thus, we suppose that both the sets K and U are polytopes. In the following, we

develop an efficient algorithm of computing the backward reachable set for system (3.5) on the constraint set K .

Due to the special structure of linear systems, (3.5) can be rewritten as

$$x(t + 1) = A_\sigma x(t) - v(t),$$

where $v(t) \in V_\sigma = \{-B_\sigma u | u \in U\}$. In the case of discrete-time, the backward reachable set of subsystem A_1 over a unit time step is computed as follows:

$$\text{Reach}(K) = A_1^{-1}(K \oplus V_1). \tag{3.6}$$

The operator \oplus means the Minkowski sum of two sets. $A_1^{-1}(\cdot)$ denotes the preimage of a set under the map A_1 . Similarly, the backward reachable set of subsystem $A_l (l = 2, \dots, N)$ over a unit time step is obtained by

$$\text{Reach}(K) = A_l^{-1}(K \oplus V_l). \tag{3.7}$$

According to the discussion above, we get the reachable set of system (3.5) on the set K as follows:

$$\begin{aligned} \text{Reach}(K) &= A_1^{-1}(K \oplus V_1) \cup \dots \cup A_N^{-1}(K \oplus V_N) \\ &= \bigcup_{i=1}^N A_i^{-1}(K \oplus V_i), \end{aligned} \tag{3.8}$$

where $V_i = \{-B_i u | u \in U\}, i \in \Lambda$.

Formula (3.8) gives us a method of computing the reachable set in the case of linear systems. In this method, the computation of the reachable set is transformed into several operations on the sets. As long as the matrices of all the subsystems are invertible, the reachable set can be expressed by (3.8). It is true for linear systems, but not for nonlinear systems. From (3.8), the computation of the reachable set involves the operations performed on sets including Minkowski sum, linear transformation, and union. In addition, it is easy to see that the operation of intersection on sets is required in the computation of the viability kernel. In the following, we illustrate the processes of these operations performed on sets of polytopes.

Let

$$\begin{aligned} K &= \{x \in \mathbb{R}^n | H_1 x \leq b_1\}, \\ V &= \{x \in \mathbb{R}^n | H_2 x \leq b_2\}. \end{aligned} \tag{3.9}$$

We can compute the preimage of K under the linear transformation A_i as follows:

$$A_i^{-1}K = \{x \in \mathbb{R}^n | H_1 A_i^{-1} x \leq b_1\}.$$

The intersection and union of K and V can also be easily computed, respectively.

$$K \cap V = \left\{ x \in \mathbb{R}^n \mid \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} x \leq \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right\},$$

$$K \cup V = \{ x \in \mathbb{R}^n \mid H_1 x \leq b_1 \text{ or } H_2 x \leq b_2 \}.$$

However, the Minkowski sum is difficult to compute using the representations as (3.9). We need to convert them to another form. In fact, a polytope can also be expressed by its vertices (see [29])

$$P = \text{co}\{v_1, \dots, v_s\} = \left\{ \sum_{i=1}^s \lambda_i v_i \mid \lambda_i \geq 0, \sum_{i=1}^s \lambda_i = 1 \right\},$$

where v_1, v_2, \dots, v_s are vertices of P . Similarly, K and V have the following representations:

$$\begin{aligned} K &= \text{co}\{w_1, \dots, w_m\}, \\ V &= \text{co}\{v_1, \dots, v_n\}, \end{aligned} \tag{3.10}$$

where w_1, \dots, w_m and v_1, \dots, v_n are vertices of K and V , respectively.

In fact, (3.10) is equivalent to (3.9). This is a well-studied problem, and many algorithms have been proposed to solve it. Then the Minkowski sum of K and V is given as follows [29]:

$$K \oplus V = \text{co}\{w_i + v_j \mid i = 1, \dots, m; j = 1, \dots, n\}.$$

Moreover, the set $K \oplus V$ is also a polytope.

According to the discussion above, we can give a method of computing the reachable set on the polyhedral constraint set and the polyhedral input set. Applying (3.8) to Algorithm 1, we obtain an algorithm of computing the viability kernel for system (3.5) on the finite horizon T (see Algorithm 2).

It should be noted that the formula of computing the reachable set can be expressed by the operations on the constraint set and the control input set. Moreover, we get the viability kernel by implementing the operations on sets including intersection, union, linear transformation, and Minkowski sum. These operations are easy to implement when both the constraint set and the control input set are polytopes. Therefore, the algorithm has a small amount of computation.

3.3 Analysis of the algorithms

We next give some basic properties of Algorithms 1, 2.

Theorem 2 *If Algorithm 1 terminates with $l < n$ and $K_l = \emptyset$, then $\text{Viab}_{[0,n] \cap \mathbb{Z}_+}(K) = \emptyset$ and $\text{Viab}_{[0,\infty) \cap \mathbb{Z}_+}(K) = \emptyset$. If Algorithm 1 terminates with $l < n$ and $K_l = K_{l-1}$, then $\text{Viab}_{[0,n] \cap \mathbb{Z}_+}(K) = K_l$ and $\text{Viab}_{[0,\infty) \cap \mathbb{Z}_+}(K) = K_l$.*

Proof The viability kernel over a finite horizon $T = [0, n] \cap \mathbb{Z}_+$ is empty when Algorithm 1 terminates with $l < n$ and $K_l = \emptyset$. It means that all the trajectories of system (3.1) starting at

Algorithm 2 The viability kernel of discrete-time switched linear systems

```

Let  $K_0 \leftarrow K$ 
 $l \leftarrow 0$ 
while  $l \leq n$  do
  if  $K_l = \emptyset$  then
     $K_n \leftarrow \emptyset$ 
    break
  endif
  if  $K_l = K_{l-1}$  then
     $K_n \leftarrow K_l$ 
    break
  endif
   $L \leftarrow \bigcup_{i=1}^N A_i^{-1}(K_l \oplus V_i)$ 
   $K_{l+1} \leftarrow K_0 \cap L$ 
   $l \leftarrow l + 1$ 
end while
return  $K_n$ 

```

the given region K will leave K at the k th ($k \geq l$) step of the algorithm. Thus, the viability kernel over an infinite horizon is also empty, that is, $\text{Viab}_{[0,\infty) \cap \mathbb{Z}_+}(K) = \emptyset$.

If Algorithm 1 terminates with $l < n$ and $K_l = K_{l-1}$, according to Theorem 1, the viability kernel over a finite horizon T is $\text{Viab}_{[0,m] \cap \mathbb{Z}_+}(K) = K_l$. In fact, at the k th ($k > l$) step of the algorithm, we have

$$K_k = K_l. \tag{3.11}$$

Equation (3.11) still holds when k tends to infinity. This tells us that $\text{Viab}_{[0,\infty) \cap \mathbb{Z}_+}(K) = K_l$. This completes the proof of the theorem. \square

Algorithm 2 is a special case of Algorithm 1, thus it has the same conclusion.

4 The viability kernel of continuous-time switched systems

In this section, we discuss the computation of the viability kernel for continuous-time switched systems. By discretizing continuous-time systems, two efficient algorithms of approximating the viability kernel are developed by using the result of Sect. 3.

4.1 Continuous-time switched systems

Consider the following system:

$$\dot{x}(t) = f_{\sigma(t)}(x(t), u(t)), \tag{4.1}$$

where the function f_{σ} is bounded by $M > 0$ on the constraint set $K \subseteq \mathbb{R}^n$. It means that for all $x \in K$ and $u \in U$, we have $\|f_{\sigma}(x(t), u(t))\| \leq M, \sigma \in \Lambda$. We define a distance of a point $x \in \mathbb{R}^n$ from a nonempty set $S \subset \mathbb{R}^n$ as follows:

$$d_S(x) = \inf_{s \in S} \|x - s\|. \tag{4.2}$$

In the following, we compute an under-approximation of the viability kernel for system (4.1) on the finite horizon $T = [0, \tau]$. Given a discretization time step δ , define an under-approximation of the viability constraint set

$$K_\delta = \{x \in K \mid d_{K^c}(x) \geq \delta M\}. \tag{4.3}$$

K_δ approximates to K by the distance δM because we only consider the state at discrete times $t_k = k\delta$. For any interval $[t_k, t_{k+1}]$, the solution of (4.1) does not leave K at any time $t \in [t_k, t_{k+1}]$. In fact, a solution $x(t)$ of (4.1) can travel a distance of at most δM from its initial state $x(t_k)$.

$$\|x(t_k) - x(t)\| \leq \int_{t_k}^t \|\dot{x}(\tau)\| d\tau \leq M(t - t_k) \leq \delta M.$$

Thus, we define the recursive formula as follows:

$$\begin{cases} K_0(\delta) = K_\delta, \\ K_{n+1}(\delta) = K_0(\delta) \cap \text{Reach}_\delta(K_n(\delta)). \end{cases} \tag{4.4}$$

At each step, we compute the set of states from which $K_n(\delta)$ is reachable, and then intersect this set with $K_0(\delta)$. Each computed set $K_n(\delta)$ is an approximation of the finite horizon viability kernel $\text{Viab}_{[0,\tau]}(K)$ for $\tau = n\delta$.

Theorem 3 *Assume that f_σ is bounded by $M > 0$ on the set K for all $\sigma \in \Lambda$. For any time step δ , the set $\{K_n(\delta)\}$ produced by (4.4) satisfies*

$$K_n(\delta) \subseteq \text{Viab}_{[0,n\delta]}(K). \tag{4.5}$$

Proof In order to obtain the reachable set of $K_n(\delta)$ for system (4.1), we first compute the reachable set of $K_n(\delta)$ for every subsystem $f_\sigma(x, u)$, $\sigma \in \{1, \dots, N\}$. Thus, we get N reachable sets. So the reachable set $\text{Reach}_\delta(K_n(\delta))$ can be obtained by intersecting these N reachable sets. The rest of the proof is similar to that of Theorem 2 in [17]. This completes the proof of the theorem. □

Theorem 4 *Suppose that f_σ is bounded by $M > 0$ on the set K for all $\sigma \in \Lambda$. For any time step δ , the set $\{K_n(\delta)\}$ produced by (4.4) satisfies*

$$\text{Viab}_{[0,\tau]}(\text{int}(K)) \subseteq \bigcup_{n \in \mathbb{N}} K_n(\delta) \subseteq \text{Viab}_{[0,\tau]}(K). \tag{4.6}$$

Proof According to (4.5) and Theorem 3 in [17], the set $\{K_n(\delta)\}$ produced by (4.4) satisfies (4.6). This completes the proof of the theorem. □

According to (4.5) and (4.6), the set $K_n(\delta)$ produced by (4.4) is an under-approximation of the finite horizon viability kernel $\text{Viab}_{[0,\tau]}(K)$ for $\tau = n\delta$. Then we get an algorithm of computing a finite horizon viability kernel for system (4.1) on the constraint set K (see Algorithm 3).

Algorithm 3 The viability kernel of continuous-time switched systems

```

Choose  $\delta > 0$ 
 $n \leftarrow \frac{\tau}{\delta}$ 
 $K_0 \leftarrow K_\delta$ 
 $l \leftarrow 0$ 
while  $l \leq n$  do
  if  $K_l = \emptyset$  then
     $K_n \leftarrow \emptyset$ 
    break
  endif
  if  $K_l = K_{l-1}$  then
     $K_n \leftarrow K_l$ 
    break
  endif
   $L \leftarrow \text{Reach}_{[0, \delta n]}(K_l)$ 
   $K_{l+1} \leftarrow K_0 \cap L$ 
   $l \leftarrow l + 1$ 
end while
return  $K_n$ 

```

Algorithm 3 gives us a method of computing the viability kernel over a finite horizon for continuous-time switched systems. The main iterative process of the algorithm is as follows. First, we define an under-approximation $K_0(\delta)$ of the given set and compute the set of backward reachable states from $K_0(\delta)$. Next, we intersect the backward reachable set with $K_0(\delta)$ to get $K_1(\delta)$ and compute the reachable set from $K_1(\delta)$. Then, we intersect the reachable set with $K_0(\delta)$ to get a new set $K_2(\delta)$. By repeating this process, we eventually reach an under-approximation of the viability kernel.

4.2 Continuous-time switched linear systems

Since linear systems have special structure and some special properties, it is more convenient to compute the reachable set. Thus, we consider a simple algorithm of approximating the viability kernel for switched linear systems. Let us restrict our attention to the following system:

$$\dot{x}(t) = A_\sigma x(t) + B_\sigma u(t), \quad (4.7)$$

where the state $x \in \mathbb{R}^n$, the time $t \in T = [0, \tau] \cap \mathbb{R}_+$, the control input $u \in U$, U is a compact and convex set, and switching signal σ takes values in a finite set $\Lambda = \{1, \dots, N\}$. $A_\sigma \in \mathbb{R}^{n \times n}$ is an invertible matrix and bounded by M and $B_\sigma \in \mathbb{R}^{n \times m}$. Before computing the backward reachable set of system (4.7) on the constraint set K , we introduce a notion of Hausdorff distance.

Definition 3 Let X, Y be nonempty compact convex subsets of \mathbb{R}^n . The Hausdorff distance of X and Y is defined as follows:

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} \|x - y\|, \sup_{y \in Y} \inf_{x \in X} \|x - y\| \right\}. \quad (4.8)$$

We next compute an approximation of the reachable set by discretizing system (4.7). By choosing a time step δ , the number of time steps is $n = \frac{\tau}{\delta}$. Suppose that the control $u(t) = \tilde{u}(t_i)$ is constant over the time interval $[t_i, t_{i+1}]$, $i = 0, 1, \dots, n-1$. We discretize system (4.7) by using Eulerian method, then we have

$$\begin{cases} \tilde{x}(t_{i+1}) = \tilde{x}(t_i) + \delta(A_\sigma \tilde{x}(t_i) + B_\sigma \tilde{u}(t_i)), \\ \tilde{u}(t_i) \in U, \quad i = 0, 1, \dots, n-1. \end{cases} \tag{4.9}$$

Applying Theorem 1 of [30] to system (4.7) and the corresponding discretization system (4.9), we obtain the following result.

Theorem 5 *Suppose that $\text{Reach}_{[0,\delta n]}(K_l)$ and $\tilde{\text{Reach}}_{[0,\delta n]}(K_l)$ are the backward reachable sets of systems (4.7) and (4.9) on the set K_l over the time interval $[t_i, t_{i+1}]$, respectively. Then there exists a constant C such that*

$$d_H(\text{Reach}_{[0,\delta n]}(K_l), \tilde{\text{Reach}}_{[0,\delta n]}(K_l)) \leq C\delta, \tag{4.10}$$

where δ is a discretization time step.

Proof The conclusion can be established by following Theorem 1 in [30]. This completes the proof of the theorem. \square

Theorem 5 shows us that the Hausdorff distance between the reachable sets $\text{Reach}_{[0,\delta n]}(K_l)$ and $\tilde{\text{Reach}}_{[0,\delta n]}(K_l)$ has an upper bound, and the distance is proportional to the size of δ . Moreover, when δ is smaller, the upper bound of the associated Hausdorff distance is smaller. Theorem 5 provides us a theoretical basis for using $\tilde{\text{Reach}}_{[0,\delta n]}(K_l)$ approximating to $\text{Reach}_{[0,\delta n]}(K_l)$, that is, we can obtain an approximation of the reachable set for a continuous-time system by computing the reachable set for the corresponding discretization system. Thus, we only need to compute the reachable set for system (4.9), which is equivalent to the following system:

$$\tilde{x}(t+1) = (I + \delta A_\sigma)\tilde{x}(t) + \delta B_\sigma \tilde{u}(t). \tag{4.11}$$

I is an identity matrix. Setting $V_\sigma = -\delta B_\sigma U = \{-\delta B_\sigma \tilde{u}(t) | \tilde{u}(t) \in U\}$, the reachable set over a single time step can be computed as follows:

$$\tilde{\text{Reach}}_{[0,\delta n]}(K_l) = (I + \delta A_\sigma)^{-1}(K_l \oplus V_\sigma). \tag{4.12}$$

Since A_σ is non-singular, we assume that $\lambda_1, \dots, \lambda_{nN}$ are the eigenvalues of the matrices A_1, \dots, A_N , respectively. When the time step δ satisfies the following condition

$$\delta < \min \left\{ \frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_{nN}} \right\}, \tag{4.13}$$

then the matrix $I + \delta A_\sigma$ is non-singular for any $\sigma \in \Lambda$. Thus, (4.12) is meaningful.

Algorithm 4 The viability kernel of continuous-time switched linear systems

```

Choose  $\delta > 0$ 
 $n \leftarrow \frac{\tau}{\delta}$ 
 $K_0 \leftarrow K_\delta$ 
 $l \leftarrow 0$ 
while  $l \leq n$  do
  if  $K_l = \emptyset$  then
     $K_n \leftarrow \emptyset$ 
    break
  endif
  if  $K_l = K_{l-1}$  then
     $K_n \leftarrow K_l$ 
    break
  endif
   $L \leftarrow \bigcup_{\sigma=1}^N (I + \delta A_\sigma)^{-1} (K_l \oplus V_\sigma)$ 
   $K_{l+1} \leftarrow K_0 \cap L$ 
   $l \leftarrow l + 1$ 
end while
return  $K_n$ 

```

Based on Algorithm 3 and (4.12), we propose an algorithm of computing an approximation of the viability kernel for system (4.7) (see Algorithm 4).

Algorithm 4 is the development of Algorithm 3 in the case of linear systems. The computation of the reachable set has been simplified. Moreover, an approximation of the viability kernel can be obtained by computing the viability kernel of the corresponding discretization system. The approximation error can be made arbitrarily small by choosing δ small enough.

Algorithms 3 and 4 have the same properties as listed in Theorem 2. The proof is similar to that of Theorem 2, and we do not illustrate it in detail.

5 Illustrative examples

5.1 Example 1

Consider the following discrete-time switched linear system:

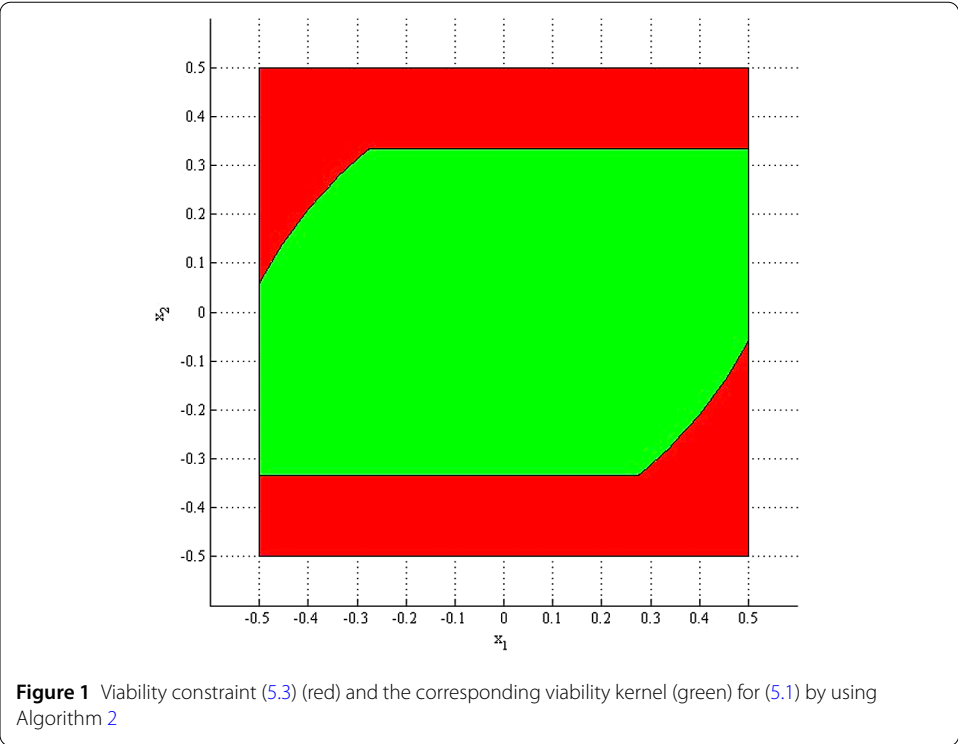
$$A_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (5.1)$$

where the state $x \in \mathbb{R}^2$, the time $t \in T = [0, n] \cap \mathbb{R}_+$, the control input $u \in U \subset \mathbb{R}^2$. We assume that the control input set

$$U = \{u = (u_1, u_2)^T \mid -0.3 \leq u_i \leq 0.3, i = 1, 2\}, \quad (5.2)$$

and the state constraint set

$$K = \{x = (x_1, x_2)^T \mid -0.5 \leq x_i \leq 0.5, i = 1, 2\}. \quad (5.3)$$



We next compute the viability kernel for system (5.1) on the state constraint set (5.3) with a horizon of 40 steps. The viability kernel can be obtained by using Algorithm 2, the result shows in Fig. 1.

Figure 1 shows us that the red region is the viability constraint set, and the green region is the viability kernel for system (5.1).

5.2 Example 2

Consider the following continuous-time switched linear system:

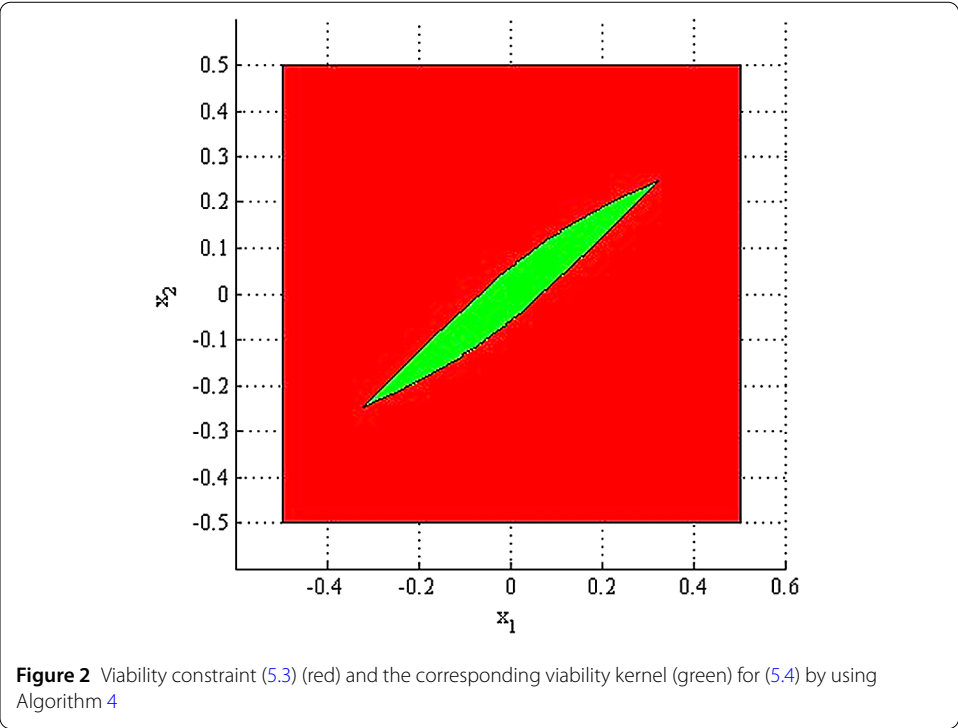
$$\begin{aligned}
 A_1 &= \begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}, & A_2 &= \begin{pmatrix} -1 & -10 \\ 0.1 & -1 \end{pmatrix}, \\
 B_1 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & B_2 &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.
 \end{aligned}
 \tag{5.4}$$

The control input set and the state constraint set are given by (5.2) and (5.3), respectively. We compute the viability kernel of system (5.4) on the region (5.3) with the time horizon [0, 4]. Choosing the time step $\delta = 0.1$, we discretize system (5.4). Then we obtain an approximation of the viability kernel by using Algorithm 4. The result shows in Fig. 2.

5.3 Example 3

Consider the following discrete-time switched linear system:

$$A_1 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B_1 = B_2 = I,
 \tag{5.5}$$



where I is an identity matrix, the state $x \in \mathbb{R}^3$, the time $t \in T = [0, n] \cap \mathbb{R}_+$, the control input $u \in U \subset \mathbb{R}^3$. We assume that the control input set

$$U = \{u = (u_1, u_2, u_3)^T \mid -0.03 \leq u_i \leq 0.03, i = 1, 2, 3\}, \tag{5.6}$$

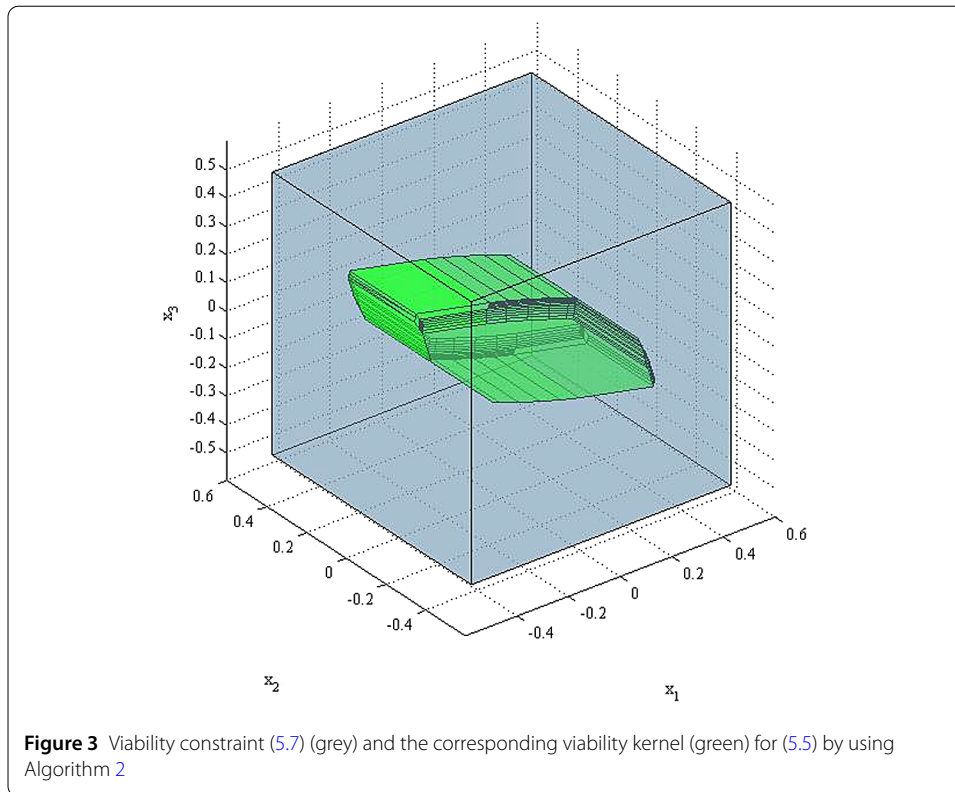
and the state constraint set

$$K = \{x = (x_1, x_2, x_3)^T \mid -0.5 \leq x_i \leq 0.5, i = 1, 2, 3\}. \tag{5.7}$$

We next compute the viability kernel for system (5.5) on the state constraint set (5.7) with a horizon of 10 steps. The viability kernel can be obtained by using Algorithm 2, the result shows in Fig. 3.

6 Conclusions

We established a connection between the viability and the reachability, which enables us to compute the viability kernel for switched systems in terms of the backward reachable sets. For the discrete-time switched systems, the viability kernel over a finite horizon can be computed exactly by using the algorithm we proposed. According to the special structure of switched linear systems, we obtained a simple algorithm that is easy to implement. For the continuous-time switched systems, we got an approximation of the true viability kernel by computing the viability kernel of the corresponding discretization system. The approximation error can be made arbitrarily small by choosing the time step small enough. The method is simple when the systems are linear. The effectiveness of the obtained results has been verified through numerical examples.



Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.11171221, Doctoral Program Foundation of Institutions of Higher Education of China under Grant No.20123120110004.

Funding

This work is supported by the National Natural Science Foundation of China under Grant No.11171221, Doctoral Program Foundation of Institutions of Higher Education of China under Grant No.20123120110004.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

Author details

¹School of Management, University of Shanghai for Science and Technology, Shanghai, China. ²School of Science, Inner Mongolia University of Science and Technology, Baotou, China.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 30 March 2018 Accepted: 13 August 2018 Published online: 29 August 2018

References

1. Aubin, J.P.: *Viability Theory*. Springer, Berlin (2011)
2. Béné, C., Doyen, L., Gabay, D.: A viability analysis for a bio-economic model. *Ecol. Econ.* **36**(3), 385–396 (2001)
3. Mullon, C., Cury, P., Shannon, L.: Viability model of trophic interactions in marine ecosystems. *Nat. Resour. Model.* **17**(1), 71–102 (2004)
4. Sanogo, C., Raïssi, N., Miled, S.B., et al.: A viability analysis of fishery controlled by investment rate. *Acta Biotheor.* **61**(3), 341–352 (2013)
5. Liu, L., Gao, Y., Wu, Y.P.: Speed optimization control for wheeled robot navigation with obstacle avoidance based on viability theory. *Automatika* **57**(2), 428–440 (2017)
6. Spiteri, R.J., Pai, D.K., Ascher, U.M.: Programming and control of robots by means of differential algebraic inequalities. *IEEE Trans. Robot. Autom.* **16**(2), 135–145 (2000)
7. Jaulin, L., Bars, F.L.: An interval approach for stability analysis: application to sailboat robotics. *IEEE Trans. Robot. Autom.* **29**(1), 282–287 (2013)

8. Chen, Z., Gao, Y.: Determining the viable unbounded polyhedron under linear control systems. *Asian J. Control* **16**(5), 1561–1567 (2014)
9. Lou, Z.E., Gao, Y.: The exponential stability for a class of hybrid systems. *Asian J. Control* **15**(2), 624–629 (2013)
10. Dong, X.X., Zhao, J.: Incremental passivity and output tracking of switched nonlinear systems. *Int. J. Control* **85**(10), 1477–1485 (2012)
11. Gao, Y.: Viability criteria for differential inclusions. *J. Syst. Sci. Complex.* **24**(5), 825–834 (2011)
12. Lv, J.F., Gao, Y., Zhao, N.: Viability criteria for a switched system on bounded polyhedron. *Asian J. Control* (2017, in press). <https://doi.org/10.1002/asjc.1719>
13. Saint-Pierre, P.: Approximation of viability kernel. *Appl. Math. Optim.* **29**(2), 187–209 (1994)
14. Mitchell, I.M., Bayen, A.M., Tomlin, C.J.: A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Trans. Autom. Control* **50**(7), 947–957 (2005)
15. Coquelin, P.A., Martin, S., Munos, R.: A dynamic programming approach to viability problems. In: *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 178–184 (2007)
16. Deffuant, G., Chapel, L., Martin, S.: Approximating viability kernels with support vector machines. *IEEE Trans. Autom. Control* **52**(5), 933–937 (2007)
17. Maidens, J.N., Kaynama, S., Mitchell, I.M., Oishi, M.M.K.: Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica* **49**, 2017–2029 (2013)
18. Kaynama, S., Mitchell, I.M., Oishi, M., et al.: Scalable safety-preserving robust control synthesis for continuous-time linear systems. *IEEE Trans. Autom. Control* **60**(11), 3065–3070 (2015)
19. Maidens, J., Arcak, M.: Reachability analysis of nonlinear systems using matrix measures. *IEEE Trans. Autom. Control* **60**(1), 265–270 (2015)
20. Fribourg, L., Goubault, E., Putot, S., et al.: A topological method for finding invariant sets of switched systems. In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pp. 61–70. ACM, New York (2016)
21. Riedinger, P., Sigalotti, M., Daafouz, J.: On the algebraic characterization of invariant sets of switched linear systems. *Automatica* **46**(6), 1047–1052 (2010)
22. Han, Y., Gao, Y.: The calculation of discriminating kernel based on viability kernel and reachability. *Adv. Differ. Equ.* **2017**, 370 (2017)
23. Aubin, J.P., Lygeros, J., Quincampoix, M., et al.: Impulse differential inclusions: a viability approach to hybrid systems. *IEEE Trans. Autom. Control* **47**(1), 2–20 (2002)
24. Margellos, K., Lygeros, J.: Viable set computation for hybrid systems. *Nonlinear Anal. Hybrid Syst.* **10**, 45–62 (2013)
25. Haimovich, H., Seron, M.M.: Componentwise ultimate bound and invariant set computation for switched linear systems. *Automatica* **46**(11), 1897–1901 (2010)
26. Abate, A., Prandini, M., Lygeros, J., et al.: Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica* **44**(11), 2724–2734 (2008)
27. Lygeros, J., Johansson, K.H., Simic, S.N., et al.: Dynamical properties of hybrid automata. *IEEE Trans. Autom. Control* **48**(1), 2–17 (2003)
28. Lygeros, J.: On reachability and minimum cost optimal control. *Automatica* **40**(6), 917–927 (2004)
29. Kvasnica, M., Grieder, P., Baotić, M., et al.: Multi-parametric toolbox (MPT). In: *Hybrid Systems: Computation and Control*, pp. 448–462 (2004)
30. Dontchev, A.L., Farkhi, E.M.: Error estimates for discretized differential inclusions. *Computing* **41**(4), 349–358 (1989)

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
