The Controlled Composition Analysis of Hybrid Automata

Ying Shang, Michael D. Lemmon
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
USA

Abstract

A controlled hybrid automaton is a hybrid automaton whose continuous-state dynamics are described by inhomogeneous differential equations. This paper presents a sufficient condition for the existence of global non-terminating solutions in controlled hybrid automata. The condition is based on a recursive algorithm that can always terminate after a finite number of iterations to a limit set of states called the *inner viability kernel*. If the inner viability kernel is non-empty, then there exists a measurable control under which the hybrid automaton generates a global non-terminating solution. The inner viability kernel is easily computed for controlled hybrid automata whose underlying continuous dynamics have controllability manifolds of dimension $n-1$ or higher (where n is the dimension of the continuous state space). The more important is that this result can also be used to infer the existence of global solutions to compositions of controlled hybrid automata, thereby providing a foundation for the analysis of large scale hybrid systems.

1 Introduction

Hybrid systems are dynamical systems whose states consist of discrete and continuous-state variables. Hybrid automata [1] are commonly used mathematical models for the analysis and design of hybrid dynamical systems. A hybrid automaton models the coupled interaction of discrete event and continuous dynamical systems. The continuous state evolves according to a differential equation called the modal equation. The discrete transition happens when the continuous states satisfy a guard predicate on the hybrid automaton arc.

The original hybrid model[1] assumed that modal equations were homogeneous differential equations. This paper studies hybrid automata whose modal equations are inhomogeneous differential equations driven by an exogenous control signal. This paper calls this system as a *controlled hybrid automaton* [9]. In particular, this paper investigates the existence of measurable control signals under which the controlled hybrid automaton generates a global non-terminating solution. Roughly speaking, a global non-terminating solution is a system trajectory which has an infinite number of switches between the discrete states.

Prior work has addressed the existence of global solutions within the framework of algorithmic verification [1, 2, 3]. In algorithmic verification, one first identifies a cycle of arcs in the automaton's graph and then one recursively computes the set of all states that can satisfy the guard predicates on each arc of the cycle. If this recursive algorithm terminates in a non-empty set, then we know there exists a global solution generating the specified cycle. If the recursion terminates in an empty set, then we know that no global solution exists. This paper refers to the limiting set generated by the recursion as the cycle's *outer viability kernel*. Results on outer viability kernels have appeared for hybrid automata [7] and impulsive hybrid systems [8].

There are numerous problems with the recursion used to compute the outer viability kernel. In the first place, the computation of the kernel is expensive. The recursion actually computes the set of all states from which a guard condition can be reached in finite time. Computing this preset requires precise knowledge of the system flow. In practice, it is only feasible to compute approximations to these flows and numerous verification tools have been proposed which rely on flow-pipe or ellipsoidal approximations of the flow [4, 5, 6]. Another problem with the existing recursion is that it rarely terminates after a finite number of iterations. For all but the simplest class of hybrid automata, it has been shown that the outer viability recursion is undecidable [3]. Finally, the outer viability kernels of two different cycles cannot generally be used to compute the outer viability kernel of a sequential composition of these cycles. In other words, the existing recursion does not lend itself to a compositional analysis of hybrid automata.

This paper presents an alternative recursion that computes the so-called *inner viability kernel* of a cycle. The inner viability kernel is the set of all states such that any point in the set can be controlled to reach any point in this set. The inner viability kernel is always a subset of the outer viability kernel and hence it provides a more restrictive sufficient condition on the existence of global non-terminating solutions. In return for these restrictions, however, we obtain some important benefits. In particular, the inner viability recursion always terminates after a finite number of iterations so it provides a semi-decidable way to verify the existence of global solutions. In addition to this, the inner viability kernel is easily computed for controlled hybrid automata whose underlying continuous subsystems have controllability manifolds of dimension $n - 1$ or greater. Finally, it can be shown that the inner viability kernel supports a compositional analysis of hybrid automata. Composition in this paper means that the inner viability kernels of two different cycles can be used to determine the inner viability kernel of a concatenation of these two cycles.

The remainder of this paper is organized as follows. Section 2 presents the controlled hybrid automaton. Section 3 uses a recursive algorithm to compute the outer and inner viability kernels for a cycle accepted by the hybrid automaton. This section states that the inner viability recursion is a semi-decidable algorithm for the existence of non-terminating solution in controlled hybrid automata. Section 4 uses the inner viability kernel to demonstrate the compositional analysis of hybrid automata. Section 5 presents a simple example illustrating many of the points raised in this paper. Final conclusions and future research plans will be found in section 6.

2 Controlled Hybrid Automaton

A *controlled hybrid automaton* [9] is a hybrid automaton whose underlying continuous dynamics are represented by inhomogeneous differential equations. A formal definition is given below:

Definition 2.1. (Controlled Hybrid Automaton) *A controlled hybrid automaton, $\mathcal{H}$, is characterized by the 7-tuple, $(X, \Sigma, U, A, G, F, Q_0)$, where*

- *X is an n-dimensional manifold and an element $x \in X$ is called the automaton's continuous state.*
- *Σ is a discrete set of integers and an element $i \in \Sigma$ is called the automaton's discrete state or mode.*
- *U is a subset of $\Re^m$ and a vector $u \in U$ is called the automaton's controlled input.*
- *$A \subset \Sigma \times \Sigma$ consists of ordered pairs of integers. An element $(i, j) \in A$ is called a discrete event of the automaton, which is generated when the system's discrete state changes.*
- *$G : A \to \mathcal{P}(X)$ is a map that takes each discrete event $(i, j) \in A$ onto a closed subset of X. The value that G takes at (i, j) is denoted as G_i^j, which is called the guard for event (i, j).*
- *$F : X \times U \times \Sigma \to \Re^n$ is a map that takes an ordered triple, $(x, u, i) \in X \times U \times \Sigma$ onto a vector $F(x, u, i) \in \Re^n$, $(x, i) \in X \times \Sigma$ is the current state and $u \in U$ is current control.*
- *$Q_0 \subset X \times \Sigma$ is a closed subset of the hybrid state space $X \times \Sigma$ that we refer to as the automaton's initial set.*

For a given $i \in \Sigma$, we let $F_i : X \times U \to \Re^n$ denote the vector field associated with mode i such that $F_i(x, u) = F(x, u, i)$. We refer to the differential equation

$$\dot{x}(\tau) = F_i(x(\tau), u(\tau))$$

as the ith modal equation.

A *trajectory* is a function $q : I \to X \times \Sigma$ where I is an interval in $\Re$. The value that q takes at time $\tau \in I$ is denoted as $q(\tau) = (x(\tau), \sigma(\tau))$. We refer to $x(\tau) \in X$ and $\sigma(\tau) \in \Sigma$ as the trajectory's continuous and discrete state, respectively, at time τ.

Given a trajectory q, we say that time $\tau \in I$ is *regular* if q is continuous at τ. Otherwise we say that τ is a *switching instant*. If a trajectory q has an infinite number of switching instants, we say that q is *non-blocking* or *non-terminating*. If q is non-blocking with an infinite number of switching instants occurring in a finite subinterval of I, then q is said to be *chattering*.

A trajectory $q : [0, T) \to X \times \Sigma$ is a *solution* of the controlled hybrid automaton $\mathcal{H}$ if and only if $q(0) \in Q_0$ and the following conditions hold. First, for all closed subintervals $[\tau_a, \tau_b] \subseteq [0, T)$, that contain no switching instants, there exists a mode $i \in \Sigma$, a function $x : [\tau_a, \tau_b] \to X$, and a measurable function $u : [\tau_a, \tau_b] \to U$ such that $\sigma(t) = i$ and

$$x(t) = \int_{\tau_a}^{t} F(x(\tau), u(\tau), i) d\tau$$

for all $t \in [\tau_a, \tau_b]$. Second at any switching instant $\tau_s \in [0, T)$ there exist discrete modes $i, j \in \Sigma$ such that $(i, j) \in A$, $x(\tau_s) \in G_i^j$ and $j = \lim_{t \downarrow \tau_s} \sigma(t)$.

Given a non-blocking trajectory q that is a solution to the hybrid automaton $\mathcal{H}$, there exists a sequence of switching instants, $\{\tau_j\}_{j=0}^{\infty}$ such that $\tau_j < \tau_{j+1}$. Associated with this sequence of switching instants, there is a sequence of discrete states $\{i_j\}_{j=0}^{\infty}$ such that $i_j = \sigma(\tau_j)$ for $j = 1, \ldots, \infty$. For $j = 0$ we let $i_0 = \sigma(0)$. Define the *trace* of trajectory q as the sequence of events arcs $\alpha = \{\alpha_j\}_{j=0}^{\infty}$ where the jth event is $\alpha_j = (i_j, i_{j+1}) \in A$ for $j = 0, \ldots, \infty$. We say that this trace is *logically accepted* by the hybrid automaton, $\mathcal{H} = (X, \Sigma, U, A, G, F, Q_0)$ if the trace is accepted by the associated state machine (Σ, A).

Consider a controlled hybrid automaton $\mathcal{H}$. A state $q_f \in X \times \Sigma$ is *reachable* from $q_0 \in Q_0$, if there exists a finite time $T \geq 0$ and a trajectory $q : [0, T] \to X \times \Sigma$ such that q is a solution to the hybrid automaton, $q(0) = q_0$, and $q(T) = q_f$.

3 Global Non-blocking Solutions

This section derives conditions characterizing the existence of global non-terminating solutions in controlled hybrid automata. We present a sufficient condition based on the use of the *inner* viability recursion.

3.1 Outer and Inner Viability Recursion

Given a state $x_f \in X$, we define the *preset* of x_f under event (i, j) as the set of all states $x_0 \in X$ such that (x_f, j) is reachable from (x_0, i). We denote the preset of x_f under arc (i, j) as $\mathrm{Pre}_i^j(x_f)$. We will often denote this preset as $\mathrm{Pre}(x_f)$ when the arc is understood from the problem's context. With respect to this preset we define the *outer* and *inner* precondition of $K \subseteq X$, respectively, as

$$\begin{aligned} \overline{\mathrm{Pre}}(K) &= \bigcup_{x_f \in K} \mathrm{Pre}(x_f) \\ \underline{\mathrm{Pre}}(K) &= \bigcap_{x_f \in K} \mathrm{Pre}(x_f) \end{aligned}$$

Consider the trace of N events

$$\alpha = (i_0, i_1), (i_1, i_2), \cdots, (i_{N-2}, i_{N-1}), (i_{N-1}, i_0)$$

and assume α is logically accepted by the hybrid automaton $\mathcal{H} = (X, U, \Sigma, A, G, F, Q_0)$ shown in figure 1. Let G_j^k denote the guard set $G_{i_j}^{i_k}$. Let $\mathrm{mod}(k)$ denote the integer k modulo N. Let α^* denote the trace obtained by concatenating an infinite number of α traces.

Let $\{\overline{\Gamma}_j\}_{j=0}^{\infty}$ denote the infinite sequence of sets in X generated by the recursion,

$$\begin{aligned} \overline{\Gamma}_0 &= G_0^1 \\ \overline{\Gamma}_{j+1} &= G_{\mathrm{mod}(N-j-1)}^{\mathrm{mod}(N-j)} \cap \overline{\mathrm{Pre}}(\overline{\Gamma}_j) \end{aligned} \qquad (3.1)$$

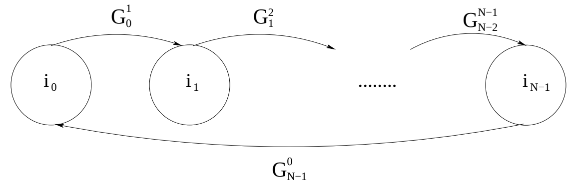

Figure 1: Controlled Hybrid Automaton $\mathcal{H}$

for $j = 0, \ldots, \infty$. The recursion in equation 3.1 is called the *outer* preset recursion. We define the *inner* preset recursion by the equations,

$$\begin{aligned} \underline{\Gamma}_0 &= G_0^1 \\ \underline{\Gamma}_{j+1} &= G_{\text{mod}(N-j-1)}^{\text{mod}(N-j)} \cap \underline{\text{Pre}}(\underline{\Gamma}_j) \end{aligned} \tag{3.2}$$

for $j = 0, \ldots, \infty$.

We denote the set, $\overline{\Gamma}_N$, that is obtained after N iterations of the *outer* preset recursion; the set $\underline{\Gamma}_N$, that is obtained after N iterations of the *inner* preset recursion. This paper will study the use of the inner recursion (Eq. 3.2) to verify the existence of global non-terminating solutions to a controlled hybrid automaton. The outer recursion method in controlled hybrid automata was introduced in [13] is a minor extension of the prior work in [7] and [8]. The limit set of the outer recursion is called "outer viability kernel" in this paper, denoted by $\overline{\Gamma}^*$. Roughly speaking, a set of states, Γ is called viable if for all initial conditions in it there exists a solution of the dynamical system that still remains in Γ. The largest subset of Γ which is viable is called the *viability kernel*. The term "outer viability kernel" is not standard in the viability literature. We use it in order to distinguish $\overline{\Gamma}^*$ from a smaller viable set, $\underline{\Gamma}^*$, which is the limit point of the inner recursion of equation 3.2.

3.2 Inner Viability Kernel

This section will present the important properties with respect to the existence of global non-blocking solutions in controlled hybrid automata. The first two lemmas are "inner"-version of lemmas of the outer recursion [13]. To keep the flow of this paper, the proofs of this subsection will be found in Appendix.

Lemma 3.1. *If $\underline{\Gamma}_N$ is non-empty, then for any $x_f \in G_0^1$, there exists a $T_f \geq 0$ and a trajectory $q : [0, T_f] \to X \times \Sigma$ such that for any $q_0 = (x_0, i_0) \in Q_0$, where $x_0 \in \underline{\Gamma}_N$,*

- *q solves the hybrid automaton $\mathcal{H}$,*
- *q has the trace α,*
- *and $x(T_f) = x_f$*

The lemma 3.1 states any points of G_0^1 are reachable from $\underline{\Gamma}_N$. The next lemma shows $\underline{\Gamma}_N$ is the largest subset of G_0^1 such that any point of $\underline{\Gamma}_N$ can reach any point of G_0^1.

Lemma 3.2. *If $x_0 \notin \underline{\Gamma}_N$, then there exists at least one point x_f in G_0^1 that is not reachable from x_0 under a trajectory q with trace α.*

From the lemmas 3.2 we get the following lemma, which states that the inner recursion terminates after a finite number of iterations.

Lemma 3.3. $\underline{\Gamma}_{2N} = \underline{\Gamma}_N$

Lemma 3.3 shows the inner recursion algorithm terminates after N steps. We can let the limit of the inner recursion be denoted as $\lim_{n\to\infty} \underline{\Gamma}_{jN} = \underline{\Gamma}^*$. From lemma 3.3, we know that this limiting set is $\underline{\Gamma}^* = \underline{\Gamma}_N$. This lemma tells us the computation of the inner viability recursion is much easier than the outer viability recursion, because it can always terminate in a finite number of iterations.

Proposition 3.1. *If $\underline{\Gamma}^*$ is non-empty, then there exists a trajectory $q : [0,\infty) \to X \times \Sigma$ such that for any $q_0 = (x_0, i_0) \in Q_0$ where $x_0 \in \underline{\Gamma}^*$,*

- *q solves the hybrid automaton $\mathcal{H}$,*
- *q has the trace α,*
- *and there exists a sequence of switching instants $\{\tau_j\}_{j=0}^{\infty}$ such that $x(\tau_j) \in \underline{\Gamma}^*$.*

Proposition 3.1 states that the inner preset recursion is a semi-decidable algorithm. To formally characterize the relationship between the above results and prior work in [7, 8], let's consider a map $Q_\alpha : \mathcal{P}(G_0^1) \to \mathcal{P}(G_0^1)$ that is associated with the hybrid automata $\mathcal{H}$ and discrete trace $\alpha \in \Sigma^*$. We define this map as follows. If $x_0 \in G_0^1$, then $Q_\alpha(x_0)$ is the set of points $x \in G_0^1$ such that *there exists* a trajectory q with trace α that solves $\mathcal{H}$ and there exist times T_0, T_1, and T_2 such that

- for all $0 < T_0 < \tau < T_1$, we know that $x(\tau) \notin G_0^1$ and
- for all $T_1 \leq \tau \leq T_2$, we know that $x(\tau) \in G_0^1$.

We refer to Q_α as a first-return map. In that if K is a subset of the domain of Q_α, then there exists a measurable control that generates a trajectory q with trace α that returns to $Q_\alpha(K) \subset G_0^1$. With regard to the preceding definition, we say that a set K is *viable* with respect to Q_α if $[Q_\alpha]^n(K) \in K$ for all $n \geq 0$. We say a set K is the *viability kernel* of Q_α if it is the largest viable set. From the construction of $\underline{\Gamma}^*$, it should be clear that it is a viable set under map Q_α and moreover, it should be apparent that it is a subset of the outer viability kernel $\overline{\Gamma}^*$. Consider a set $K \subset G_0^1$. We will say that this set is *inner viable* if for all $x_0 \in K$ there exists a measurable control that can return to any point in G_0^1. Clearly $\underline{\Gamma}^*$ is inner viable by lemma 3.1. But by lemma 3.2, we can clearly see that this is the largest such inner viable set associated with trace α. We can therefore refer to $\underline{\Gamma}^*$ as the *inner viability kernel* associated with α. The inner viability kernel is smaller than the outer viability kernel, $\overline{\Gamma}^*$, and this is why the condition in proposition 3.1 is only sufficient whereas the corresponding result in proposition 1 [13] is necessary and sufficient. Nevertheless, the inner viability kernel (when it exists) has some important properties with respect to the composition of event traces. These properties are studied in the next section.

4 Compositional Analysis

This section presents the compositional analysis of the inner viability kernel. Composition in this paper means the inner viability kernels of two different cycles can be used to determine the inner viability kernel of a sequential composition of these cycles. Consider a hybrid automaton $\mathcal{H}$. Let $\overline{V} : \Sigma^* \to \mathcal{P}(X)$ map a trace α, that is logically accepted by $\mathcal{H}$, onto its outer viability kernel. Let $\underline{V} : \Sigma^* \to \mathcal{P}(X)$ map the string α onto its inner viability kernel.

Proposition 4.1. *Consider a hybrid automaton $\mathcal{H}$ and let α and β be two traces*

$$\begin{aligned} \alpha &= (i_0, i_1), (i_1, i_2), \cdots, (i_{N-2}, i_{N-1}), (i_{N-1}, i_0) \\ \beta &= (j_0, j_1), (j_1, j_2), \cdots, (j_{M-2}, j_{M-1}), (j_{M-1}, j_0) \end{aligned}$$

with $i_0 = j_0$. Then the string $\alpha\beta$ is logically accepted by $\mathcal{H}$ and $\underline{V}(\alpha\beta) = \underline{V}(\alpha) \cap \underline{V}(\beta)$.

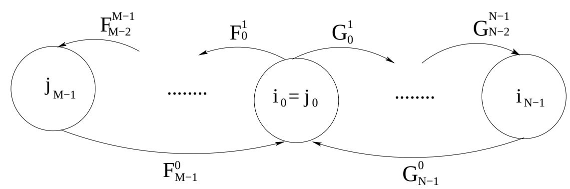

Figure 2: Hybrid Automaton $\mathcal{H}$

Proposition 4.1 is the main result of this paper. Proofs will be found in Appendix. It is important because it tells us that $\underline{V}$ is a homomorphism. On a practical level, this means that the inner viability kernel supports a compositional analysis. This means that if we know the inner viability kernels of α and β, we can use this information to immediately compute the inner viability kernel for the composed string $\alpha\beta$. Moreover, we can now use this result to establish conditions on the existence of non-blocking solutions to the hybrid automata that are formed by the sequential composition of viable traces α and β. The following corollary states this fact.

Corollary 4.1. *Assume $\underline{V}(\alpha)$ and $\underline{V}(\beta)$ are the inner viability kernels of controlled hybrid automata $\mathcal{H}$ with trace α and β*

$$\begin{aligned} \alpha &= (i_0, i_1), (i_1, i_2), \cdots, (i_{N-2}, i_{N-1}), (i_{N-1}, i_0) \\ \beta &= (j_0, j_1), (j_1, j_2), \cdots, (j_{M-2}, j_{M-1}), (j_{M-1}, j_0) \end{aligned}$$

with $i_0 = j_0$, if $\underline{V}(\alpha) \cap \underline{V}(\beta) \neq \emptyset$, then there exists a non-terminating solution for hybrid automata $\mathcal{H}$ with trace $\alpha\beta$.

Actually, Proposition 4.1 is a stronger statement for the existence of non-terminating solutions to controlled hybrid automata, because only $\underline{V}(\alpha\beta) \supseteq \underline{V}(\alpha) \cap \underline{V}(\beta)$ will be enough to answer whether or not there exists a non-terminating solution in a controlled hybrid automaton. If the intersection of $\underline{V}(\alpha)$ and $\underline{V}(\beta)$ is non-empty, then obviously $\underline{V}(\alpha\beta)$ is not an empty set. Therefore there exists a non-terminating solution to the controlled hybrid automaton with trace α and β. The compositional nature of the inner viability kernel is an important property that distinguishes it from the more commonly used outer viability kernel. In particular, we show that $\overline{V}$ is not necessarily a homomorphism.

Proposition 4.2. *Consider a hybrid automaton $\mathcal{H}$ and let α and β be two traces*

$$\begin{aligned} \alpha &= (i_0, i_1), (i_1, i_2), \cdots, (i_{N-2}, i_{N-1}), (i_{N-1}, i_0) \\ \beta &= (j_0, j_1), (j_1, j_2), \cdots, (j_{M-2}, j_{M-1}), (j_{M-1}, j_0) \end{aligned}$$

with $i_0 = j_0$. Then the string $\alpha\beta$ is logically accepted by $\mathcal{H}$ and $\overline{V}(\alpha\beta) \subseteq \overline{V}(\alpha) \cap \overline{V}(\beta)$.

Proposition 4.2 not only shows that $\overline{V}$ is not necessarily a homomorphism, but it also shows that the composition of two event traces does not hold for the outer viability kernel. This means even if the intersection of two outer viability kernels is not empty, we still can not determine whether the outer viability kernel of the concatenations of the two traces is empty or not, because it is only a subset of the intersection of two outer viability kernels. This means that given the information of the two outer viability kernels of two traces, we can not determine the existence of non-terminating solution to the controlled hybrid automaton with the concatenation of the two traces. This is the main difference between inner and outer viability kernels.

5 Example

The previous sections state the principal results concerning the inner viability kernel approach to address the conditions on the existence of the non-blocking solutions in controlled hybrid automata. We consider an example illustrating the use of the inner viability kernel in evaluating the composition of sequential event traces. Compositional analysis is the main advantage of the inner viability kernel approach. The following lemma formally identifies a situation in which the inner viability kernel, $\underline{\Gamma}^*$, is easily computed. Proofs will be found in the Appendix.

Lemma 5.1. *Let $\mathcal{H}$ be a controlled hybrid automaton. Assume trace $\alpha = (i_0, i_1), (i_1, i_2), \cdots,$ $(i_{N-2}, i_{N-1}), (i_{N-1}, i_0)$ is logically accepted by $\mathcal{H}$ whose underlying continuous-state dynamics are stabilizable and $\dot{x}(\tau) = A_{i_k}x + B_{i_k}u, k = 0, \cdots, N-1$ have controllable subspaces $\mathcal{R}(\mathcal{C}_{i_k})$ with dimension $n-1$, where $\mathcal{C}_{i_k}$is a controllability matrix*

$$\mathcal{C}_{i_k} = [B_{i_k} \; A_{i_k}B_{i_k} \; \cdots \; A_{i_k}^{n-1}B_{i_k}]$$

associated with mode i_k. Let e_{i_kj}, $j = 1, \cdots, n-1$, be a standard basis vector for the subspace $\mathcal{R}(\mathcal{C}_{i_k})$.

1. *$\forall k \in 1, \cdots, N-1$, assume $\underline{\Gamma}_k \cap \mathcal{R}(\mathcal{C}_{i_k}) = \emptyset$, i.e. $\underline{\Gamma}_k \subseteq H^p$ or H^n, where $H^p = \{x | c^T x \geq 0, c \cdot e_{i_kl} = 0\}$ and $H^n = \{x | c^T x \leq 0, c \cdot e_{i_kl} = 0\}$ are positive and negative halfspaces associated with the controllability subspace $\mathcal{R}(\mathcal{C}_{i_k})$. Assume $\underline{\Gamma}_k$ is a convex hull with the vertices v_l, $l = 1, \cdots, m$.*
   - *if $\underline{\Gamma}_k \subseteq H^p$, then $\underline{\mathrm{Pre}}(\underline{\Gamma}_k) = \{y | c^T y \geq D = max(d(v_l, \mathcal{R}(\mathcal{C}_{i_k})))\}$;*
   - *if $\underline{\Gamma}_k \subseteq H^n$, then $\underline{\mathrm{Pre}}(\underline{\Gamma}_k) = \{y | c^T y \leq D = max(d(v_l, \mathcal{R}(\mathcal{C}_{i_k})))\}$;*
2. *$\forall k \in 1, \cdots, N-1$, if $\underline{\Gamma}_k \cap H^p \neq \emptyset$ and $\underline{\Gamma}_k \cap H^n \neq \emptyset$, then $\underline{\mathrm{Pre}}(\underline{\Gamma}_k) = \emptyset$.*

This lemma shows that under certain assumptions, the inner presets of subsystems are halfspaces. We can easily determine the inner viability kernel by taking the intersections of $G_{\mathrm{mod}(N-j-1)}^{\mathrm{mod}(N-j)}$ and the halfspaces. If all underlying continuous-state dynamics of a hybrid automaton are controllable, we can easily prove $\underline{\mathrm{Pre}}(\underline{\Gamma}_j) = \overline{\mathrm{Pre}}(\underline{\Gamma}_j) = \mathcal{R}^n, j = 0, \cdots, \infty$. In this case, the inner and the outer viability kernels are the same, so the outer viability kernel supports composition of sequential event traces.

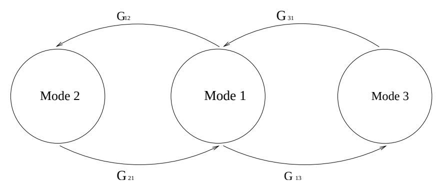

Figure 3: hybrid automaton model

Consider a controlled hybrid automaton $\mathcal{H}$ shown in figure 3 with subsystem 1:

$$\dot{x} = \begin{bmatrix} -1 & -2 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} -1 \\ 1 \end{bmatrix} u$$

and subsystem 2:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u.$$

and subsystem 3:

$$\dot{x} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u.$$

These three subsystems are uncontrollable but stabilizable, and their controllable subspaces have dimension 1. The *guard set* is defined to be the union of convex hulls of subsets L_k of $\mathcal{R}^2$, denoted $\cup_{k=1}^{K} conv(L_k)$. The subset L_k consists of a finite number of vectors $l_k^1, l_k^2, \cdots, l_k^m$. It is clear that the vectors in L_k are the extreme points(vertices) of its convex hull $conv(L_k)$. Let $G_{ij}, i, j = 1, 2, 3$ denote the guard sets from i to j and

$$G_{12} = conv(\left\{ \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{9}{4} \\ 1 \end{bmatrix} \right\})$$

and

$$G_{21} = conv(\left\{ \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix} \right\})$$

and

$$G_{13} = G_{13}^a \bigcup G_{13}^b$$

$$G_{13}^a = conv(\left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} \frac{5}{2} \\ \frac{1}{2} \end{bmatrix}, \begin{bmatrix} \frac{5}{2} \\ \frac{3}{2} \end{bmatrix} \right\})$$

$$G_{13}^b = conv(\left\{ \begin{bmatrix} 4 \\ \frac{1}{2} \end{bmatrix}, \begin{bmatrix} 4 \\ -1 \end{bmatrix}, \begin{bmatrix} \frac{7}{2} \\ -\frac{1}{2} \end{bmatrix} \right\})$$

and

$$G_{31} = conv(\left\{ \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 4 \\ 1 \end{bmatrix}. \right\})$$

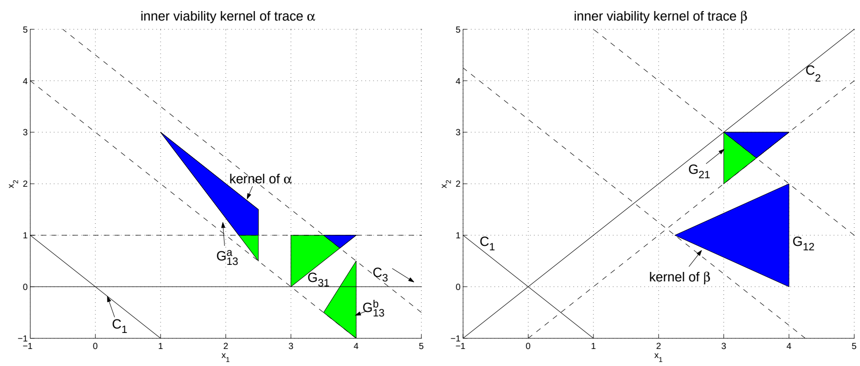

Figure 4: The inner viability kernels of trace α and β

We define the trace α is the cycle from mode 1 to mode 3 and then to mode 1, and the trace β is the cycle from mode 1 to model 2 and then to mode 1. We computed the inner viability kernels, $\underline{V}(\alpha)$, $\underline{V}(\beta)$, of trace α and β using the inner preset recursion and computation methods of Lemma 5.1:

$$\underline{V}(\alpha) = conv(\left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} \frac{11}{5} \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{5}{2} \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{5}{2} \\ \frac{3}{2} \end{bmatrix} \right\})$$

and

$$\underline{V}(\beta) = G_{12}.$$

Figure 4 shows the geometry of the inner viability kernel of trace α and β. The left figure shows the guard sets of the two arcs connecting modes 1 and 3. There are three sets in this figure, G^a_{13} and G^b_{13} are the disconnected sets whose union is the guard G_{13}. The set G_{31} is the other arc's guard. C_1 and C_3 are the controllable subspaces for modes 1 and 3, respectively. The dotted lines in the figure are the affine varieties of the controllable subspaces. The dark subsets shown in the figure are the inner viability kernels that we computed with trace α. In a similar way, the right figure illustrates the inner viability kernels we computed with trace β.

We used the inner recursion to compute the inner viability kernel for composed trace $\alpha\beta$. The resulting set is

$$\underline{V}(\alpha\beta) = conv(\left\{ \begin{bmatrix} \frac{5}{2} \\ \frac{6}{5} \end{bmatrix}, \begin{bmatrix} \frac{9}{4} \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{5}{2} \\ 1 \end{bmatrix} \right\}).$$

We computed $\underline{V}(\alpha) \bigcap \underline{V}(\beta)$ and found that $\underline{V}(\alpha\beta) = \underline{V}(\alpha) \bigcap \underline{V}(\beta)$. This illustrates the result of Proposition 4.1. Because $\underline{V}(\alpha\beta) \neq \emptyset$, then by proposition 3.1 there exists a non-blocking solution to the hybrid automaton $\mathcal{H}$.

Next, we computed the outer viability kernel for trace α and β using the outer preset recursion. In this case we found that $\overline{V}(\alpha) = G^a_{13} \bigcup V_1$, where

$$V_1 = conv(\left\{ \begin{bmatrix} \frac{15}{4} \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ \frac{1}{2} \end{bmatrix}, \begin{bmatrix} 4 \\ 0 \end{bmatrix} \right\})$$

and $\overline{V}(\beta) = G_{12}$. Taking the intersection of $\overline{V}(\alpha)$ and $\overline{V}(\beta)$ we found that $\overline{V}(\alpha) \bigcap \overline{V}(\beta) = V_2 \bigcup V_3$, where

$$V_2 = conv(\left\{ \begin{bmatrix} \frac{5}{2} \\ \frac{5}{6} \end{bmatrix}, \begin{bmatrix} \frac{9}{4} \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{5}{2} \\ \frac{7}{6} \end{bmatrix} \right\})$$

$$V_3 = conv(\left\{ \begin{bmatrix} \frac{137}{36} \\ \frac{1}{9} \end{bmatrix}, \begin{bmatrix} 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ \frac{1}{2} \end{bmatrix} \right\}).$$

Obviously, the point $(4, 0)$ is in $\overline{V}(\alpha) \bigcap \overline{V}(\beta)$, but $(4, 0)$ is not in $\overline{V}(\alpha\beta)$. This is because the only points that point $(4, 0)$ can reach in G_{13} are $(\frac{5}{2}, \frac{1}{2})$ and $y = -x + 3, x \in [3.5, 4]$. From the figure 4 it can be seen that $(4, 0)$ reaches $(\frac{5}{2}, \frac{1}{2})$ and $y = -x + 3, x \in [3.5, 4]$ under a two part control that first moves the state along the controllable subspace C_3 and then along the affine variety of C_1. Since $(\frac{5}{2}, \frac{1}{2})$ and $y = -x + 3, x \in [3.5, 4]$ are not in $\overline{V}(\alpha) \bigcap \overline{V}(\beta)$, the point $(4, 1)$ can not be in the outer viability kernel $\overline{V}(\alpha\beta)$. So we can conclude $\overline{V}(\alpha) \bigcap \overline{V}(\beta) \neq \overline{V}(\alpha\beta)$.

We can also directly compute the outer viability kernel of trace $\alpha\beta$ using the outer recursion. This computation shows that $\overline{V}(\alpha\beta) = V_2 \bigcup V_4$, where

$$V_4 = conv(\left\{ \begin{bmatrix} \frac{15}{4} \\ \frac{1}{4} \end{bmatrix}, \begin{bmatrix} 4 \\ \frac{1}{2} \end{bmatrix}, \begin{bmatrix} 4 \\ \frac{1}{4} \end{bmatrix} \right\}).$$

It can be seen that $\overline{V}(\alpha) \bigcap \overline{V}(\beta) \supset \overline{V}(\alpha\beta)$, which demonstrates the result of Proposition 4.2.

This simple example demonstrates all of the principal properties of the inner viability kernel. The inner preset recursion terminates after 2 steps; the inner viability kernel is easily computed for the controlled hybrid automaton $\mathcal{H}$ whose underlying dynamics have controllability manifold of dimension 1; the inner viability kernels of two different cycles, $\underline{V}(\alpha)$, $\underline{V}(\beta)$, can be used to obtain the inner viability kernel of a sequential composition of these cycles, $\underline{V}(\alpha\beta)$. This example also verifies that the outer viability kernel can not support the composition of event traces.

6 Conclusions

The principal result of this paper is a sufficient condition for the existence of global non-blocking solutions to controlled hybrid automata. This condition is derived from the inner recursion algorithm that terminates after a finite number of iterations to a limit set called the *inner viability kernel*. The inner viability approach presented in this paper has the following advantages. First, the *inner* preset recursion always terminates after a finite number of iterations. Second, the inner viability kernel is easily computed for controlled hybrid automata whose underlying continuous dynamics have controllability manifolds of dimension $n-1$ or higher. Third, in our opinion, the major benefit is the composition property of inner viability kernel. This property is useful to the analysis of large scale hybrid systems.

There exist, however, problems with the inner viability approach. All of preceding benefits of the inner viability approach rely on the fact that the inner viability kernel must be a non-empty set. If it is empty, we can say nothing about the existence of non-terminating solutions to controlled hybrid automata. Moreover, although the outer viability kernel does not necessarily ensure the composition of sequential event traces, we may find it supports cycle composition in some special cases. We have yet to answer under what conditions, the outer viability kernel supports cycle composition.

The previous observations suggest future research directions. The first direction is to find the conditions under which the inner viability kernels of a hybrid automaton are non-empty. The second direction is to find a class of hybrid systems whose outer viability kernels support cycle composition.

Acknowledgement: Research supported in part by National Science Foundation(ECS-99-86918).

References

[1] R. Alur, C. Courcoubetis, T.A. Henzinger, and P-H Ho, " Hybrid Automata: An Algorithmic Approach to The Specification and Verification of Hybrid Systems," *Hybrid Systems*, Grossman, Nerode and Ravn, Lecture Notes in Computer Science, Vol 736, Springer-verlag, 1993.

[2] T.A.Henzinger, "Hybrid Automata with Finite Bisimulation," *ICALP 95: Automata, Languages, and Programming*, Springer-Verlag, pp.324-335, 1995.

[3] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, Pravin Varaiya, "What's Decidable about Hybrid Automata," *Proceedings of the 27th Annual Symposium on Theory of Computing*, ACM Press, pp.373-382, 1995.

[4] Alongkrit Chutinan, Bruce H. Krogh, " Verification of Polyhedral-invariant Hybrid Automata Using Polygonal Flow Pipe Approximations," *Hybrid Systems: Computation and Control*, F.W. Vaandrager and J.H.van Schuppen, Eds. New York: Springer-Verlag, LNCS 1569, pp.76-90, 1999.

[5] Alexander B.Kurzhanski, Pravin Varaiya, "Ellipsoidal Techniques for Reachability Analysis, " *Hybrid Systems: Computation and Control*, N.Lynch and B.H.Krogh, Eds. New York: Springer-Verlag, LNCS 1790, pp.202-214, 2000.

[6] T. Dang and O. Maler, " Reachability Analysis via Face Lifting," *Hybrid Systems: Computation and Cotrol*, S. Sastry and T.A. Henzinger,Eds. New York: Springer-Verlag, LNCS 1386, pp.96-109, 1998.

[7] Akash Deshpande, Pravin Varaiya, " Viable Control of Hybrid Systems," *Hybrid Systems II*, P. Antsaklis, W. Kohn, A. Nerode, S. Sastry, Eds., Lecture Notes in Computer Science, LNCS 999, Springer-Verlag,pp. 128-147, 1995.

[8] J.P.Aubin, John Lygeros, Marc Qincampoix, Shankar Sastry, Fellow, IEEE and Nicolas Seube, " Impulse Differential Inclusions :A Viability Approach to Hybrid Systems," *IEEE Transactions On Automatic Control*, Vol.47, No1, 2002.

[9] Michael D. Lemmon, " On the Existence of Solutions to Controlled Hybrid Automaton, " *Hybrid Systems: Computation and Control*, LNCS 1790, pp.229-242, 2000.

[10] Michael D. Lemmon, " Deadlock-Free Piecewise-linear Controlled Hybrid Automata," *Proceedings of IEEE Conference on Decision and Control*, Vol 1, pp.478-483, 2000.

[11] J. P. Aubin, H. Frankowska, " Set Valued Analysis," Boston, MA: Birkhäuser, 1990.

[12] J.P.Aubin, "Viability Theory," Birkhäuser, 1991.

[13] Ying Shang, Michael D. Lemmon, "The Controlled Composition of Hybrid Automata Through the Use of Inner Viability Kernels," *subbmitted to IEEE Conference on Decision and Control*, 2002(`http://www.nd.edu/~lemmon`).

Appendix:(Proofs)

Proof of Lemma 3.1: If $\underline{\Gamma}_N$ is non-empty, then clearly $\underline{\Gamma}_j$ is also non-empty for $j = 0, \ldots, N-1$. So we'll assume $q_0 = (x_0, i_0) \in Q_0$ and assume that $x_0 \in \underline{\Gamma}_N = G_0^1 \cap \underline{\text{Pre}}(\underline{\Gamma}_{N-1})$.

Since $x_0 \in \underline{\text{Pre}}(\underline{\Gamma}_{N-1})$ we know that for any $x_1 \in \underline{\Gamma}_{N-1})$ there exists $T_1 \geq 0$ and trajectory $q_1 : [0, T_1] \to X \times \Sigma$ such that x_1 is reachable from x_0 under $\alpha_1 = (i_0, i_1)$.

The state x_1 is any point in $\overline{\Gamma}_{N-1} = G_1^2 \cap \underline{\text{Pre}}(\underline{\Gamma}_{N-2})$. So x_1 lies in $\underline{\text{Pre}}(\Gamma_{N-2})$ which means there exists $T_2 \geq T_1 \geq 0$, there exists trajectory $q_2 : [0, T_2] \to X \times \Sigma$ such that any $x_2 \in \underline{\Gamma}_{N-2}$ is reachable from x_0 under the trace $(i_0, i_1), (i_1, i_2)$.

We can repeat the above argument a finite number of times to conclude that for any $x_N \in \underline{\Gamma}_0$ there exists $T_N \geq 0$ and there exists trajectory $q_N : [0, T_N] \to X \times \Sigma$ with trace α such that x_N is reachable from x_0. The resulting q_N clearly solves the hybrid automaton and has trace α. $\diamondsuit$

Proof of Lemma 3.2: Let's assume $\forall x_f \in G_0^1$, x_f is reachable from x_0 with trace α, then $x_0 \in \underline{\Gamma}_N$ by the definition, contradiction. So there exists at least one t_f in G_0^1 that is not reachable from x_0 under a trajectory q with trace α. $\diamondsuit$

Proof of lemma 3.3: Assume that $\underline{\Gamma}_{2N}$ is nonempty, then clearly by the inner recursion we can conclude that $\underline{\Gamma}_N$ is non-empty also. So assume that $x \in \underline{\Gamma}_N$. Since $\underline{\Gamma}_N = G_0^1 \cap \underline{\text{Pre}}(\underline{\Gamma}_{N-1})$, we can conclude $x \in G_0^1$ and so $\underline{\Gamma}_N \subset G_0^1$. By similar reasoning we can conclude that $\underline{\Gamma}_{2N} \subset G_0^1$.

Assume that $x(0) \notin \underline{\Gamma}_{2N}$. This means by lemma 3.2 that there exists at least a point in G_0^1 that is not reachable from $x(0)$ under α. But if we also assume that $x(0) \in \underline{\Gamma}_N$, lemma 3.1 also lets us infer that any point in G_0^1 is reachable from $x(0)$. This contradicts our earlier assumption that $x(0) \notin \underline{\Gamma}_{2N}$. So we must conclude that $x(0)$ must also be in $\underline{\Gamma}_{2N}$ or rather that $\underline{\Gamma}_N \subseteq \underline{\Gamma}_{2N}$.

Note that

$$\underline{\Gamma}_j = \bigcap_{k=0}^{j} \underline{\text{Pre}}^{N-k}(G_{\text{mod}(N-k)}^{\text{mod}(N-k+1)})$$

where $\underline{\mathrm{Pre}}^k$ is the kth recursion of the $\underline{\mathrm{Pre}}$ operator where $\mathrm{Pre}^0(K) = K$. Note that this means that $\underline{\Gamma}_{2N} \subseteq \underline{\Gamma}_N$. Combining this fact with the previous result, we can conclude that $\underline{\Gamma}_N = \underline{\Gamma}_{2N}$. $\diamondsuit$

Proof of Proposition 3.1: From lemma 3.3 we know that $\underline{\Gamma}^* = \underline{\Gamma}_N$. Any $q_0 = (x_0, i_0) \in Q_0$ and $x_0 \in \underline{\Gamma}^*$ can reach any point x_f in G_0^1 by lemma 3.1. Therefore, any point in $\underline{\Gamma}^*$ can be reached along the trajectory q. So there exists a sequence of switching instants $\{\tau_j\}_{j=0}^{\infty}$ such that $x(\tau_j) \in \underline{\Gamma}^*$. The trajectory q solves the hybrid automaton $\mathcal{H}$ with trace α. $\diamondsuit$

Proof of Proposition 4.1: First, we show $\underline{V}(\alpha, \beta) \supseteq \underline{V}(\alpha) \cap \underline{V}(\beta)$. By the definition of the inner-recursion by the equations 3.2, we define the inner-recursion for trace α as follows,where G_i^k denotes the guard set with event (i, k),$(i, k) \in \alpha$.

$$\begin{aligned} \underline{\Gamma}_0^{\alpha} &= G_0^1 \\ \underline{\Gamma}_{n+1}^{\alpha} &= G_{\mathrm{mod}(N-n-1)}^{\mathrm{mod}(N-n)} \cap \underline{\mathrm{Pre}}(\underline{\Gamma}_n^{\alpha}) \end{aligned}$$

for $n = 0, \ldots, N-1$. By the lemma 3.3, we know $\underline{\Gamma}_N^{\alpha} = \underline{V}(\alpha)$, which is the inner viability kernel with trace α. We define the inner-recursion for trace β, where F_j^k denotes the guard set with event (j, k),$(j, k) \in \beta$.

$$\begin{aligned} \underline{\Gamma}_0^{\beta} &= \underline{\Gamma}_N^{\alpha} \cap F_0^1 \\ \underline{\Gamma}_{m+1}^{\beta} &= F_{\mathrm{mod}(M-m-1)}^{\mathrm{mod}(M-m)} \cap \underline{\mathrm{Pre}}(\underline{\Gamma}_m^{\beta}) \end{aligned}$$

for $m = 0, \ldots, M-1$. After the inner-recursion computation, we know $\underline{\Gamma}_N^{\alpha} \cap \underline{\Gamma}_M^{\beta} = \underline{V}(\alpha, \beta)$, since $\underline{\Gamma}_M^{\beta} = \underline{\mathrm{Pre}}^{\beta}(\underline{\Gamma}_N^{\alpha} \cap F_0^1)$ with trace β and $\underline{V}(\beta) = \underline{\mathrm{Pre}}^{\beta}(F_0^1)$ with trace β, $\underline{\mathrm{Pre}}(\underline{\Gamma}_N^{\alpha} \cap F_0^1) \supseteq \underline{\mathrm{Pre}}(F_0^1)$,because of $\underline{\Gamma}_N^{\alpha} \cap F_0^1 \subseteq F_0^1$, so $\underline{\Gamma}_M^{\beta} \supseteq \underline{V}(\beta)$, so $\underline{\Gamma}_N^{\alpha} \cap \underline{\Gamma}_M^{\beta} = \underline{V}(\alpha, \beta) \supseteq \underline{\Gamma}_N^{\alpha} \cap \underline{V}(\beta) = \underline{V}(\alpha) \cap \underline{V}_(\beta)$.

Or we can use an alternative proof, consider $\forall x_0 \in \underline{V}(\alpha) \cap \underline{V}(\beta)$, any $x_f \in \underline{V}(\alpha) \cap \underline{V}(\beta)$ are reachable from x_0 with trace α and for any x_f, any $x_f^* \in \underline{V}(\alpha) \cap \underline{V}(\beta)$ are reachable from x_f with trace β, so $x_0 \in \underline{V}(\alpha, \beta)$.

Then we show the other direction $\underline{V}(\alpha, \beta) \subseteq \underline{V}(\alpha) \cap \underline{V}(\beta)$. This holds directly from the definition of inner viability kernel. For all $x_0 \in \underline{V}(\alpha\beta)$, any $x_f \in \underline{V}(\alpha\beta)$ is reachable from x_0 under trace (α, β), obviously $x_0 \in \underline{V}(\beta)$ and $x_0 \in \underline{V}(\alpha)$. so $\underline{V}(\alpha\beta) \subseteq \underline{V}(\alpha) \cap \underline{V}(\beta)$. Therefore $\underline{V}(\alpha\beta) = \underline{V}(\alpha) \cap \underline{V}(\beta)$. $\diamondsuit$

Proof of Corollary 4.1: if $\underline{V}(\alpha) \cap \underline{V}(\beta) \neq \emptyset$, by proposition 4.1, we know $\underline{V}(\alpha\beta) \neq \emptyset$, then there exists a non-terminating solution for hybrid automata $\mathcal{H}$ with trace $\alpha\beta$ by Proposition 3.1. $\diamondsuit$

Proof of Proposition 4.2 : First of all let's prove $\overline{V}(\alpha, \beta) \subseteq \overline{V}(\alpha)$. if $x \in \overline{V}(\alpha, \beta)$ but $x \notin \overline{V}(\alpha)$, using lemma 2 [13], we know there exists no point $x_f \in G_0^1$ that is reachable from x with trace α, so $x \notin \overline{V}(\alpha, \beta)$, contradiction with our assumption. Therefore, $\overline{V}(\alpha, \beta) \subseteq \overline{V}(\alpha)$.

Secondly, we show $\overline{V}(\alpha, \beta) \subseteq \overline{V}(\beta)$. Similarly,if $x \in \overline{V}(\alpha, \beta)$, but $x \notin \overline{V}(\beta)$, using lemma 2 [13], we know there exists no point $x_f \in G_0^1$ that is reachable from x with trace β, so $x \notin \overline{V}(\alpha, \beta)$, contradiction with our assumption. Therefore, $\overline{V}(\alpha, \beta) \subseteq \overline{V}(\beta)$.

So in brief, $\overline{V}(\alpha\beta) \subseteq \overline{V}(\alpha) \cap \overline{V}(\beta)$. $\diamondsuit$

Proof of Lemma 5.1:

1. Let's first consider the standard controllable form of subsystems

$$\bar{A}_{i_k} = T^{-1}A_{i_k}T = \begin{bmatrix} A^1_{i_k} & A^{12}_{i_k} \\ 0 & A^2_{i_k} \end{bmatrix},$$

$$\bar{B}_{i_k} = T^{-1}B_{i_k} = \begin{bmatrix} B^1_{i_k} \\ 0 \end{bmatrix}$$

and initial condition is

$$\hat{x}(0) = T^{-1}x(0) = T^{-1}\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} \hat{x}_1(0) \\ \hat{x}_2(0) \end{bmatrix}.$$

So

$$\hat{x}_1(t) = e^{A^1_{i_k}t}\hat{x}_1(0) + \int_0^t e^{A^1_{i_k}(t-\tau)}B^1_{i_k}u(\tau)d\tau$$

$$\hat{x}(t) = e^{A^2_{i_k}t}\hat{x}_2(0)$$

Because the original subsystems are stabilizable, then $A^2_{i_k} \leq 0$, $|\hat{x}_2(t)| \leq |\hat{x}_2(0)|$. Any $\hat{x}_1(t)$ can be reached from $\hat{x}_1(0)$ under a control u, we know $\forall \hat{x}(t) \in \mathcal{R}^n$ can be reached from $\hat{x}(0)$ in time T if $0 \leq \hat{x}_n(t) \leq \hat{x}_n(0)$ or $\hat{x}_n(0) \leq \hat{x}_n(t) \leq 0$. The new controllable matrix is $\hat{\mathcal{C}}_{i_k} = T^{-1}\mathcal{C}_{i_k}$, and $\hat{e}_{i_k j}$ for $j = 1, \cdots, n-1$ be a standard basis vector for the subspace $\mathcal{R}(\hat{\mathcal{C}}_{i_k})$. Let $\hat{H}^p$ and $\hat{H}^n$ denote the positive and negative halfspaces associated with the hyperplane $\mathcal{R}(\hat{\mathcal{C}}_{i_k})$, where $\hat{H}^p = \{x|\hat{c}^T x \geq 0, \hat{c}\cdot\hat{e}_{i_k l} = 0\}$ and $\hat{H}^n = \{x|\hat{c}^T x \leq 0, \hat{c}\cdot\hat{e}_{i_k l} = 0\}$. If the new guard set $\underline{\hat{\Gamma}}_k \subseteq \hat{H}^p$, then $\underline{\mathrm{Pre}}(\underline{\hat{\Gamma}}_k) = \{y|\hat{c}^T y \geq \hat{D} = max(d(\hat{v}_l, \mathcal{R}(\hat{\mathcal{C}}_{i_k}))) \text{ where } \hat{v}_l \text{ are vertices of } \hat{\Gamma}_k\}$; If $\underline{\hat{\Gamma}}_k \subseteq \hat{H}^n$, then $\underline{\mathrm{Pre}}(\underline{\hat{\Gamma}}_k) = \{y|\hat{c}^T y \leq \hat{D} = max(d(\hat{v}_l, \mathcal{R}(\hat{\mathcal{C}}_{i_k}))) \text{ where } \hat{v}_l \text{ are vertices of } \hat{\Gamma}_k\}$;

We can extend this special case to a general result, because coordinate transformation only changed the coordinate positions. So we know that if $\underline{\Gamma}_k \subseteq H^p$, then $\underline{\mathrm{Pre}}(\underline{\Gamma}_k) = \{y|c^T y \geq D = max(d(v_l, \mathcal{R}(\mathcal{C}_{i_k}))) \text{ where } v_l \text{ are vertices of } \underline{\Gamma}_k\}$; if $\underline{\Gamma}_k \subseteq H^n$, then $\underline{\mathrm{Pre}}(\underline{\Gamma}_k) = \{y|c^T y \leq D = max(d(v_l, \mathcal{R}(\mathcal{C}_{i_k}))) \text{ where } v_l \text{ are vertices of } \underline{\Gamma}_k\}$.

2. if $\underline{\Gamma}_k \cap H^p \neq \emptyset$ and $\underline{\Gamma}_k \cap H^n \neq \emptyset$, then $\underline{\Gamma}_k = \underline{\Gamma}^p_k \cup \underline{\Gamma}^n_k$, which are two parts divided by $\mathcal{R}(\mathcal{C}_{i_k})$ and $\underline{\Gamma}^p_k \subseteq H^p$ and $\underline{\Gamma}^n_k \subseteq H^n$. $\underline{\mathrm{Pre}}(\underline{\Gamma}_k) = \underline{\mathrm{Pre}}(\underline{\Gamma}^p_k) \cap \underline{\mathrm{Pre}}(\underline{\Gamma}^n_k) = \emptyset$, where $\underline{\mathrm{Pre}}(\underline{\Gamma}^p_k) = \{y|c^T y \geq D^p = max(d(v_l, \mathcal{R}(\mathcal{C}_{i_k}))) \text{ where } v_l \text{ are vertices of } \underline{\Gamma}^p_k\}$; and $\underline{\mathrm{Pre}}(\underline{\Gamma}^n_k) = \{y|c^T y \leq D^n = max(d(v_l, \mathcal{R}(\mathcal{C}_{i_k}))) \text{ where } v_l \text{ are vertices of } \underline{\Gamma}^n_k\}$; $\diamondsuit$