Improving Polygonal Hybrid Systems Reachability Analysis through the use of the Phase Portrait

Gordon J. Pace
Department of Computer Science and AI
University of Malta, Malta
gordon.pace@um.edu.mt

Gerardo Schneider
Department of Informatics
University of Oslo, Norway
gerardo@ifi.uio.no

ABSTRACT

Polygonal hybrid systems (SPDI) are a subclass of planar hybrid automata which can be represented by piecewise constant differential inclusions. The computation of certain objects of the phase portrait of an SPDI, namely the viability, controllability, invariance kernels and semi-separatrix curves have been shown to be efficiently decidable. On the other hand, although the reachability problem for SPDIs is known to be decidable, its complexity makes it unfeasible on large systems. We summarise our recent results on the use of the SPDI phase portraits for improving reachability analysis by (i) state-space reduction and (ii) decomposition techniques of the state space, enabling compositional parallelisation of the analysis. Both techniques contribute to increasing the feasability of reachability analysis on large SPDI systems.

1. INTRODUCTION

Hybrid systems combining discrete and continuous dynamics arise as mathematical models of various artificial and natural systems, and as approximations to complex continuous systems. They have been used in various domains, including avionics, robotics and bioinformatics. Reachability analysis has been the principal research question in the verification of hybrid systems, even if it is a well-known result that for most subclasses of hybrid systems most verification questions are undecidable. Various decidable subclasses have, subsequently, been identified, including timed [1] and rectangular automata [10], hybrid automata with linear vector fields [11], piecewise constant derivative systems (PCDs) [12] and polygonal hybrid systems (SPDIs) [4].

Compared to reachability verification, qualitative analysis of hybrid systems is a relatively neglected area [8, 9, 13, 19, 22]. Typical qualitative questions include: 'Are there 'sink' regions where a trajectory can never leave once it enters the region?' and 'Are there regions in which every point in the region is reachable from every other?'. The collection of objects in a system satisfying these and similar properties is called the *phase portrait* of the system.

Defining and constructing phase portraits of hybrid systems has been directly addressed for PCDs in [13], and for SPDIs in [5]. Given a cycle on a SPDI, the *viability* kernel is the largest set of points in the cycle which may loop forever within the cycle. The *controllability* kernel is the largest set of strongly connected points in the cycle (such that any point in the set may be reached from any other). An *invariant set* is a set of points such that each point must keep rotating within the set forever, and the *invariance kernel* is the largest such set. Algorithms for computing these kernels have been presented in [5, 21] and implemented in the tool set SPeeDI$^+$ [14].

Given the non-compositional nature of hybrid systems, decomposing SPDIs to reduce the state space and to distribute the reachability algorithms is a challenging task. A qualitative analysis of hybrid systems does, however, provide useful information for partitioning the state-space in independent subspaces, thus helping in achieving compositional analysis.

In this paper we summarise and combine some recent results [17, 16] we have obtained, showing how kernels can be used to improve the reachability analysis of SPDIs. We use kernel information to (i) reduce the number of states of the SPDI graph, based on topological properties of the plane (and in particular, those of SPDIs); and (ii) partition the reachability questions in a compositional manner, dividing the problem into independent smaller ones and combining the partial results to answer the original question, hence enabling parallelization with minimal communication costs.

2. THEORETICAL BACKGROUND

We summarize here the main definitions and results about SPDIs; for a more detailed description refer to [20]. A (positive) *affine* function $f : \mathbb{R} \to \mathbb{R}$ is such that $f(x) = ax + b$ with $a > 0$. An *affine multivalued* function $F : \mathbb{R} \to 2^{\mathbb{R}}$, denoted $F = \langle f_l, f_u \rangle$, is defined by $F(x) = \langle f_l(x), f_u(x) \rangle$ where f_l and f_u are affine and $\langle \cdot, \cdot \rangle$ denotes an interval. For notational convenience, we do not make explicit whether intervals are open, closed, left-open or right-open, unless required for comprehension. For an interval $I = \langle l, u \rangle$ we have that $F(\langle l, u \rangle) = \langle f_l(l), f_u(u) \rangle$. The *inverse* of F is defined by $F^{-1}(x) = \{y \mid x \in F(y)\}$. The *universal inverse* of F is defined by $\tilde{F}^{-1}(I) = I'$ where I' is the greatest non-empty interval satisfying $\forall x \in I' \cdot F(x) \subseteq I$.

Clearly, $F^{-1} = \langle f_u^{-1}, f_l^{-1} \rangle$ and $\tilde{F}^{-1} = \langle f_l^{-1}, f_u^{-1} \rangle$, provided that $\langle f_l^{-1}, f_u^{-1} \rangle \neq \emptyset$.

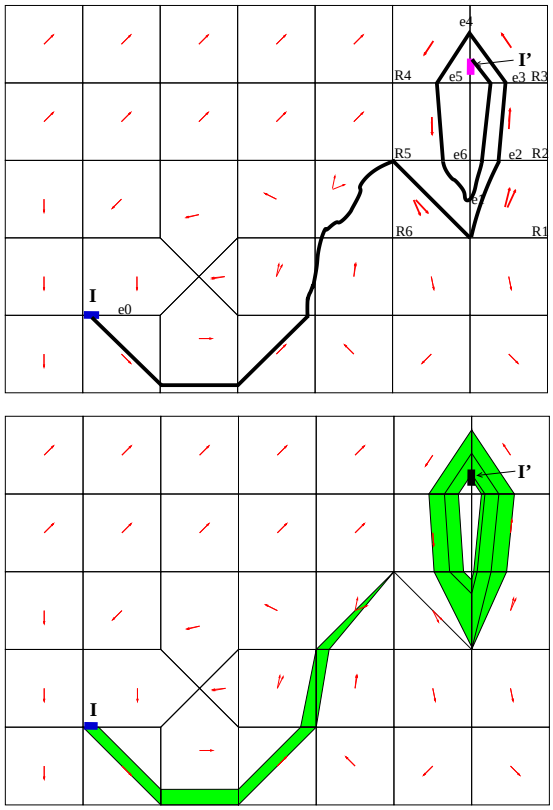

Figure 1: (a) An SPDI and its trajectory segment; (b) Reachability analysis

A *truncated affine multivalued* function (TAMF) $\mathcal{F} : \mathbb{R} \to 2^{\mathbb{R}}$ is defined by an affine multivalued function F and intervals $S \subseteq \mathbb{R}^+$ and $J \subseteq \mathbb{R}^+$ as follows: $\mathcal{F}(x) = F(x) \cap J$ if $x \in S$, otherwise $\mathcal{F}(x) = \emptyset$. For convenience we write $\mathcal{F}(x) = F(\{x\} \cap S) \cap J$. For an interval I, $\mathcal{F}(I) = F(I \cap S) \cap J$ and $\mathcal{F}^{-1}(I) = F^{-1}(I \cap J) \cap S$. The *universal inverse* of $\mathcal{F}$ is defined by $\tilde{\mathcal{F}}^{-1}(I) = I'$ if and only if I' is the greatest non-empty interval such that for all $x \in I'$, $F(x) \subseteq I$ and $F(x) = \mathcal{F}(x)$. We say that $\mathcal{F}$ is *normalized* if $S = \mathsf{Dom}(\mathcal{F}) = \{x \mid F(x) \cap J \neq \emptyset\}$ (thus, $S \subseteq F^{-1}(J)$) and $J = \mathsf{Im}(\mathcal{F}) = \mathcal{F}(S)$.

It can be proved [4], that TAMFs are closed under composition.

Theorem 1 *The composition of TAMFs $\mathcal{F}_1(I) = F_1(I \cap S_1) \cap J_1$ and $\mathcal{F}_2(I) = F_2(I \cap S_2) \cap J_2$, is the TAMF $(\mathcal{F}_2 \circ \mathcal{F}_1)(I) = \mathcal{F}(I) = F(I \cap S) \cap J$, where $F = F_2 \circ F_1$, $S = S_1 \cap F_1^{-1}(J_1 \cap S_2)$ and $J = J_2 \cap F_2(J_1 \cap S_2)$.* □

2.1 SPDIs

An *angle* $\angle_{\mathbf{a}}^{\mathbf{b}}$ on the plane, defined by two non-zero vectors $\mathbf{a}, \mathbf{b}$, is the set of all positive linear combinations $\mathbf{x} = \alpha\, \mathbf{a} + \beta\, \mathbf{b}$, with $\alpha, \beta \geq 0$, and $\alpha + \beta > 0$. We will assume that $\mathbf{b}$ is situated in the counter-clockwise direction from $\mathbf{a}$.

A *polygonal hybrid system*[1] (SPDI) is a finite partition $\mathbb{P}$ of the plane into convex polygonal sets, such that for each $P \in \mathbb{P}$ we have two vectors $\mathbf{a}_P$ and $\mathbf{b}_P$. Let $\phi(P) = \angle_{\mathbf{a}_P}^{\mathbf{b}_P}$. The SPDI is determined by $\dot{\mathbf{x}} \in \phi(P)$ for $\mathbf{x} \in P$.

Let $E(P)$ be the set of edges of P. We say that e is an *entry* of P if for all $\mathbf{x} \in e$ and for all $\mathbf{c} \in \phi(P)$, $\mathbf{x} + \mathbf{c}\epsilon \in P$ for some $\epsilon > 0$. We say that e is an *exit* of P if the same condition holds for some $\epsilon < 0$. We denote by $in(P) \subseteq E(P)$ the set of all entries of P and by $out(P) \subseteq E(P)$ the set of all exits of P.

Assumption 1 *All the edges in $E(P)$ are either entries or exits, that is, $E(P) = in(P) \cup out(P)$.*

Reachability for SPDIs is decidable provided the above assumption holds [4]; without such assumption it is not know whether reachability is decidable.

A *trajectory segment* of an SPDI is a continuous function $\xi : [0, T] \to \mathbb{R}^2$ which is smooth everywhere except in a discrete set of points, and such that for all $t \in [0, T]$, if $\xi(t) \in P$ and $\dot{\xi}(t)$ is defined then $\dot{\xi}(t) \in \phi(P)$. The *signature*, denoted $\mathsf{Sig}(\xi)$, is the ordered sequence of edges traversed by the trajectory segment, that is, $e_1, e_2, \ldots$, where $\xi(t_i) \in e_i$ and $t_i < t_{i+1}$. If $T = \infty$, a trajectory segment is called a *trajectory*.

Example 1 *Consider the SPDI illustrated in Fig. 1-(a). For sake of simplicity we will only show the dynamics associated to regions R_1 to R_6 in the picture. For each region R_i, $1 \leq i \leq 6$, there is a pair of vectors $(\mathbf{a}_i, \mathbf{b}_i)$, where: $\mathbf{a}_1 = (45, 100), \mathbf{b}_1 = (1, 4)$, $\mathbf{a}_2 = \mathbf{b}_2 = (1, 10)$, $\mathbf{a}_3 = \mathbf{b}_3 = (-2, 3)$, $\mathbf{a}_4 = \mathbf{b}_4 = (-2, -3)$, $\mathbf{a}_5 = \mathbf{b}_5 = (1, -15)$, $\mathbf{a}_6 = (1, -2), \mathbf{b}_6 = (1, -1)$. A trajectory segment starting on interval $I \subset e_0$ and finishing in interval $I' \subseteq e_4$ is depicted.* ■

We say that a signature σ is *feasible* if and only if there exists a trajectory segment ξ with signature σ, i.e., $\mathsf{Sig}(\xi) = \sigma$. From this definition, it immediately follows that extending an unfeasible signature can never make it feasible:

Proposition 1 *If a signature σ is not feasible, then neither is any extension of the signature — for any signatures σ' and σ'', the signature $\sigma'\sigma\sigma''$ is not feasible.* □

Given an SPDI $\mathcal{S}$, let $\mathcal{E}$ be the set of edges of $\mathcal{S}$, then we can define a graph $\mathcal{G}_{\mathcal{S}}$ where nodes correspond to edges of $\mathcal{S}$ and such that there exists an arc from one node to another if there exists a trajectory segment from the first edge to the second one without traversing any other edge. More formally: Given an SPDI $\mathcal{S}$, the *underlying graph of $\mathcal{S}$* (or simply the *graph of $\mathcal{S}$*), is a graph $\mathcal{G}_{\mathcal{S}} = (N_{\mathcal{G}}, A_{\mathcal{G}})$, with $N_{\mathcal{G}} = \mathcal{E}$ and $A_{\mathcal{G}} = \{(e, e') \mid \exists \xi, t\ .\ \xi(0) \in e \wedge \xi(t) \in e' \wedge \mathsf{Sig}(\xi) = ee'\}$. We say that a sequence $e_0 e_1 \ldots e_k$ of nodes in $\mathcal{G}_{\mathcal{S}}$ is a *path* whenever $(e_i, e_{i+1}) \in A_{\mathcal{G}}$ for $0 \leq i \leq k - 1$.

The following lemma shows the relation between edge signatures in an SPDI and paths in its corresponding graph.

[1] In the literature the names *polygonal differential inclusion* and *simple planar differential inclusion* have been used to describe the same systems.

Lemma 1 *If ξ is a trajectory segment of $\mathcal{S}$ with edge signature $\mathsf{Sig}(\xi) = \sigma = e_0 \dots e_p$, it follows that σ is a path in $\mathcal{G}_\mathcal{S}$.* □

Note that the converse of the above lemma is not true in general. It is possible to find a counter-example where there exists a path from node e to e', but no trajectory from edge e to edge e' in the SPDI.

2.2 Successors and Predecessors

Given an SPDI, we fix a one-dimensional coordinate system on each edge to represent points laying on edges [4]. For notational convenience, we indistinctly use letter e to denote the edge or its one-dimensional representation. Accordingly, we write $\mathbf{x} \in e$ or $x \in e$, to mean "point $\mathbf{x}$ in edge e with coordinate x in the one-dimensional coordinate system of e". The same convention is applied to sets of points of e represented as intervals (e.g., $\mathbf{x} \in I$ or $x \in I$, where $I \subseteq e$) and to trajectories (e.g., "ξ starting in x" or "ξ starting in $\mathbf{x}$").

Now, let $P \in \mathbb{P}$, $e \in in(P)$ and $e' \in out(P)$. For $I \subseteq e$, $\mathsf{Succ}_{e,e'}(I)$ is the set of all points in e' reachable from some point in I by a trajectory segment $\xi : [0,t] \to \mathbb{R}^2$ in P (i.e., $\xi(0) \in I \wedge \xi(t) \in e' \wedge \mathsf{Sig}(\xi) = ee'$). $\mathsf{Succ}_{e,e'}$ is a TAMF [4].

Example 2 *Let $e_1, \dots, e_6$ be as in Fig. 1-(a), where all the edges have local coordinates over $[0, 10]$, and $I = [l, u]$. We assume a one-dimensional coordinate system. We show only the first and last edge-to-edge TAMF of the cycle:*

$$F_{e_1e_2}(I) = \left[\tfrac{l}{4}, \tfrac{9}{20}u\right], \quad S_1 = [0,10], \quad J_1 = \left[0, \tfrac{9}{2}\right]$$
$$F_{e_6e_1}(I) = [l, 2u], \quad S_6 = [0,10], \quad J_6 = [0,10]$$

with $\mathsf{Succ}_{e_ie_{i+1}}(I) = F_{e_ie_{i+1}}(I \cap S_i) \cap J_i$, for $1 \le i \le 6$; S_i and J_i are computed as shown in Theorem 1. ■

Given a sequence $w = e_1, e_2, \dots, e_n$, since TAMFs are closed under composition, the successor of I along w, defined as $\mathsf{Succ}_w(I) = \mathsf{Succ}_{e_{n-1},e_n} \circ \dots \circ \mathsf{Succ}_{e_1,e_2}(I)$, is a TAMF.

Example 3 *Let $\sigma = e_1 \cdots e_6 e_1$. We have that $\mathsf{Succ}_\sigma(I) = F(I \cap S_\sigma) \cap J_\sigma$, where: $F(I) = [\frac{l}{4} + \frac{1}{3}, \frac{9}{10}u + \frac{2}{3}]$, with $S_\sigma = [0, 10]$ and $J_\sigma = [\frac{1}{3}, \frac{29}{3}]$.* ■

For $I \subseteq e'$, $\mathsf{Pre}_{e,e'}(I)$ is the set of points in e that can reach a point in I by a trajectory segment in P. The *$\forall$-predecessor* $\widetilde{\mathsf{Pre}}(I)$ is defined in a similar way to $\mathsf{Pre}(I)$ using the universal inverse instead of just the inverse: For $I \subseteq e'$, $\widetilde{\mathsf{Pre}}_{ee'}(I)$ is the set of points in e such that *any* successor of such points are in I by a trajectory segment in P. Both definitions can be extended straightforwardly to signatures $\sigma = e_1 \cdots e_n$: $\mathsf{Pre}_\sigma(I)$ and $\widetilde{\mathsf{Pre}}_\sigma(I)$. The successor operator thus has two "inverse" operators.

2.3 Qualitative Analysis of Simple Edge-Cycles

Let $\sigma = e_1 \cdots e_k e_1$ be a simple edge-cycle, i.e., $e_i \neq e_j$ for all $1 \le i \neq j \le k$. Let $\mathsf{Succ}_\sigma(I) = F(I \cap S_\sigma) \cap J_\sigma$ with $F = \langle f_l, f_u \rangle$ (we suppose that this representation is normalized). We denote by $\mathcal{D}_\sigma$ the one-dimensional discrete-time dynamical system defined by Succ_σ, that is $x_{n+1} \in \mathsf{Succ}_\sigma(x_n)$.

Assumption 2 *None of the two functions f_l, f_u is the identity.*

Without the above assumption the results are still valid but need a special treatment making the presentation more complicated.

Let l^* and u^* be the fixpoints[2] of f_l and f_u, respectively, and $S_\sigma \cap J_\sigma = \langle L, U \rangle$. A simple cycle is of one of the following types [4]: STAY, the cycle is not abandoned neither by the leftmost nor the rightmost trajectory, that is, $L \le l^* \le u^* \le U$; DIE, the rightmost trajectory exits the cycle through the left (consequently the leftmost one also exits) or the leftmost trajectory exits the cycle through the right (consequently the rightmost one also exits), that is, $u^* < L \vee l^* > U$; EXIT-BOTH, the leftmost trajectory exits the cycle through the left and the rightmost one through the right, that is, $l^* < L \wedge u^* > U$; EXIT-LEFT, the leftmost trajectory exits the cycle (through the left) but the rightmost one stays inside, that is, $l^* < L \le u^* \le U$; EXIT-RIGHT, the rightmost trajectory exits the cycle (through the right) but the leftmost one stays inside, that is, $L \le l^* \le U < u^*$.

Example 4 *Let $\sigma = e_1 \cdots e_6 e_1$. Then, $S_\sigma \cap J_\sigma = \langle L, U \rangle = [\frac{1}{3}, \frac{29}{3}]$. The fixpoints from Example 3 are $\frac{1}{3} < l^* = \frac{11}{25} < u^* = \frac{20}{3} < \frac{29}{3}$. Thus, σ is a STAY.* ■

Any trajectory that enters a cycle of type DIE will eventually quit it after a finite number of turns. If the cycle is of type STAY, all trajectories that happen to enter it will keep turning inside it forever. In all other cases, some trajectories will turn for a while and then exit, and others will continue turning forever. This information is crucial for proving decidability of the reachability problem.

Example 5 *Consider the SPDI of Fig. 1-(a). Fig. 1-(b) shows part of the reach set of the interval $[8, 10] \subset e_0$, answering positively to the reachability question: Is $[1, 2] \subset e_4$ reachable from $[8, 10] \subset e_0$? Fig. 1-(b) has been automatically generated by the SPeeDI toolbox we have developed for reachability analysis of SPDIs [2, 14].* ■

2.4 Reachability Analysis

It has been shown that reachability is decidable for SPDIs. Proof of the decidability result is constructive, giving an algorithmic procedure $Reach(\mathcal{S}, e, e')$ based on a depth-first search algorithm. An alternative breadth-first search algorithm which can deal with multiple edges has been presented in [15].

Theorem 2 ([4]) *The reachability problem for SPDIs is decidable.* □

[2]The fixpoint x^* is the solution of $f(x^*) = x^*$, where $f(\cdot)$ is positive affine.

An edgelist is a set of intervals of edges. Given edgelists I and I', we denote the reachability of (some part of) I' from (some part of) I as $I \xrightarrow{\mathcal{S}} I'$. Clearly, using the decidability result on edge intervals, reachability between edgelists is decidable. Although decidability may be point-to-point, edge-to-edge, edgelist-to-edgelist and region-to-region, in the rest of this paper, we will only talk about edgelist reachability.

Example 6 *Consider the SPDI of Fig. 1-(a). Fig. 1-(b) shows part of the reach set of the interval* $[8, 10] \subset e_0$*, answering positively to the reachability question: Is* $[1, 2] \subset e_4$ *reachable from* $[8, 10] \subset e_0$*? Fig. 1-(b) has been automatically generated by the SPeeDI toolbox [14] we have developed for reachability analysis of SPDIs based on the results of [4].* ■

2.5 Kernels

We present now how to compute the invariance, controllability and viability kernels of an SPDI. Proofs are omitted but for further details, refer to [5] and [21]. In the following, for σ a cyclic signature, we define $K_\sigma \subseteq \mathbb{R}^2$ as follows: $K_\sigma = \bigcup_{i=1}^{k}(int(P_i) \cup e_i)$ where P_i is such that $e_{i-1} \in in(P_i)$, $e_i \in out(P_i)$ and $int(P_i)$ is P_i's interior.

2.5.1 Viability Kernel

We now recall the definition of *viability kernel* [6]. A trajectory ξ is *viable* in K if $\xi(t) \in K$ for all $t \geq 0$. K is a *viability domain* if for every $\mathbf{x} \in K$, there exists at least one trajectory ξ, with $\xi(0) = \mathbf{x}$, which is viable in K. The *viability kernel* of K, denoted $\mathsf{Viab}(K)$, is the largest viability domain contained in K.

For $I \subseteq e_1$ we define $\overline{\mathsf{Pre}}_\sigma(I)$ to be the set of all $\mathbf{x} \in \mathbb{R}^2$ for which there exists a trajectory segment ξ starting in $\mathbf{x}$, that reaches some point in I, such that $\mathsf{Sig}(\xi)$ is a suffix of $e_2 \dots e_k e_1$. It is easy to see that $\overline{\mathsf{Pre}}_\sigma(I)$ is a polygonal subset of the plane which can be calculated using the following procedure. We start by defining $\overline{\mathsf{Pre}}_e(I) = \{\mathbf{x} \mid \exists \xi : [0, t] \to \mathbb{R}^2, t > 0 \,.\, \xi(0) = \mathbf{x} \wedge \xi(t) \in I \wedge \mathsf{Sig}(\xi) = e\}$ and apply this operation k times: $\overline{\mathsf{Pre}}_\sigma(I) = \bigcup_{i=1}^{k} \overline{\mathsf{Pre}}_{e_i}(I_i)$ with $I_1 = I$, $I_k = \mathsf{Pre}_{e_k,e_1}(I_1)$ and $I_i = \mathsf{Pre}_{e_i,e_{i+1}}(I_{i+1})$, for $2 \leq i \leq k-1$.

The following result provides a non-iterative algorithmic procedure for computing the viability kernel of K_σ on an SPDI:

Theorem 3 *If* σ *is a DIE cycle, then* $\mathsf{Viab}(K_\sigma) = \emptyset$*, otherwise* $\mathsf{Viab}(K_\sigma) = \overline{\mathsf{Pre}}_\sigma(S_\sigma)$. □

Example 7 *Fig. 2-(a) shows all the viability kernels of the SPDI given in Example 1. There are 4 cycles with viability kernels — in the picture two of the kernels are overlapping.* ■

2.5.2 Controllability Kernel

We say K is *controllable* if for any two points $\mathbf{x}$ and $\mathbf{y}$ in K there exists a trajectory segment ξ starting in $\mathbf{x}$ that reaches an arbitrarily small neighborhood of $\mathbf{y}$ without leaving K. More formally: A set K is controllable if $\forall \mathbf{x}, \mathbf{y} \in K, \forall \delta > 0, \exists \xi : [0, t] \to \mathbb{R}^2, t > 0 \,.\, (\xi(0) = \mathbf{x} \wedge |\xi(t) - \mathbf{y}| < \delta \wedge \forall t' \in [0, t] \,.\, \xi(t') \in K)$. The *controllability kernel* of K, denoted $\mathsf{Cntr}(K)$, is the largest controllable subset of K.

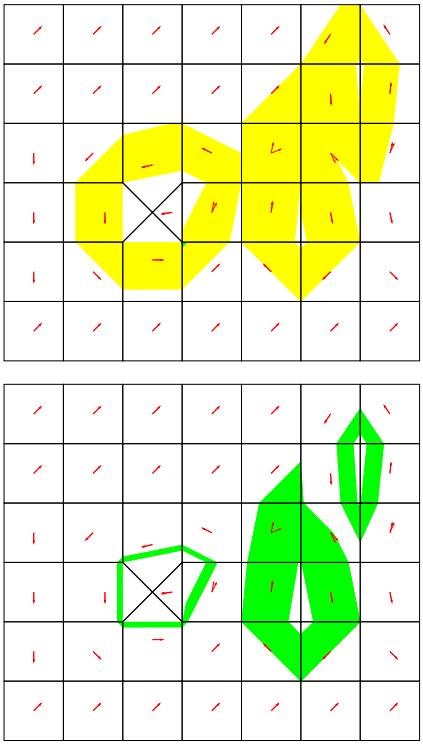

Figure 2: (a) Viability kernels; (b) Controllability kernels

For a given cyclic signature σ, we define $\mathcal{C}_\mathcal{D}(\sigma)$ as follows:

$$\mathcal{C}_\mathcal{D}(\sigma) = \begin{cases} \langle L, U \rangle & \text{if } \sigma \text{ is EXIT-BOTH} \\ \langle L, u^* \rangle & \text{if } \sigma \text{ is EXIT-LEFT} \\ \langle l^*, U \rangle & \text{if } \sigma \text{ is EXIT-RIGHT} \\ \langle l^*, u^* \rangle & \text{if } \sigma \text{ is STAY} \\ \emptyset & \text{if } \sigma \text{ is DIE} \end{cases} \quad (1)$$

For $I \subseteq e_1$ let us define $\overline{\mathsf{Succ}}_\sigma(I)$ as the set of all points $\mathbf{y} \in \mathbb{R}^2$ for which there exists a trajectory segment ξ starting in some point $x \in I$, that reaches $\mathbf{y}$, such that $\mathsf{Sig}(\xi)$ is a prefix of $e_1 \dots e_k$. The successor $\overline{\mathsf{Succ}}_\sigma(I)$ is a polygonal subset of the plane which can be computed similarly to $\overline{\mathsf{Pre}}_\sigma(I)$. Define $\mathcal{C}(\sigma) = (\overline{\mathsf{Succ}}_\sigma \cap \overline{\mathsf{Pre}}_\sigma)(\mathcal{C}_\mathcal{D}(\sigma))$. We compute the controllability kernel of K_σ as follows:

Theorem 4 $\mathsf{Cntr}(K_\sigma) = \mathcal{C}(\sigma)$. □

Example 8 *Fig. 2-(b) shows all the controllability kernels of the SPDI given in Example 1. There are 4 cycles with controllability kernels — in the picture two of the kernels are overlapping.* ■

The following result which relates controllability and viability kernels, states that the viability kernel of a given cycle

is the local basin of attraction of the corresponding controllability kernel.

Proposition 2 *Any viable trajectory in K_σ converges to* $\mathsf{Cntr}(K_\sigma)$. □

Let $\mathsf{Cntr}^l(K_\sigma)$ be the closed curve obtained by taking the leftmost trajectory and $\mathsf{Cntr}^u(K_\sigma)$ be the closed curve obtained by taking the rightmost trajectory which can remain inside the controllability kernel. In other words, $\mathsf{Cntr}^l(K_\sigma)$ and $\mathsf{Cntr}^u(K_\sigma)$ are the two polygons defining the controllability kernel.

A non-empty controllability kernel $\mathsf{Cntr}(K_\sigma)$ of a given cyclic signature σ partitions the plane into three disjoint subsets: (1) the controllability kernel itself, (2) the set of points limited by $\mathsf{Cntr}^l(K_\sigma)$ (and not including $\mathsf{Cntr}^l(K_\sigma)$) and (3) the set of points limited by $\mathsf{Cntr}^u(K_\sigma)$ (and not including $\mathsf{Cntr}^u(K_\sigma)$). We define the *inner* of $\mathsf{Cntr}(K_\sigma)$ (denoted by $\mathsf{Cntr}_{in}(K_\sigma)$) to be the subset defined by (2) above if the cycle is counter-clockwise or to be the subset defined by (3) if it is clockwise. The *outer* of $\mathsf{Cntr}(K_\sigma)$ (denoted by $\mathsf{Cntr}_{out}(K_\sigma)$) is defined to be the subset which is not the inner nor the controllability itself. Note that an edge in the SPDI may intersect a controllability kernel. In such cases, we can generate a different SPDI, with the same dynamics but with the edge split into parts, such that each part is completely inside, on or outside the kernel. Although the signatures will obviously change, it is easy to prove that the behaviour of the SPDI remains identical to the original. In the rest of the paper, we will assume that all edges are either completely inside, on or completely outside the kernels. We note that in practice splitting is not necessary since we can just consider parts of edges.

Proposition 3 *Given two edges e and e', one lying completely inside a controllability kernel, and the other outside or on the same controllability kernel, such that ee' is feasible, then there exists a point on the controllability kernel, which is reachable from e and from which e' is reachable.* □

2.5.3 Invariance Kernel

In general, an *invariant set* is a set of points such that for any point in the set, every trajectory starting in such point remains in the set forever and the *invariance kernel* is the largest of such sets. In particular, for an SPDI, given a cyclic signature, an *invariant set* is a set of points which keep rotating in the cycle forever and the *invariance kernel* is the largest of such sets. More formally: A set K is said to be *invariant* if for any $x \in K$ there exists at least one trajectory starting in it and every trajectory starting in x is viable in K. Given a set K, its largest invariant subset is called the *invariance kernel* of K and is denoted by $\mathsf{Inv}(K)$. We need some preliminary definitions before showing how to compute the kernel. The *extended* $\forall$*-predecessor* of an output edge e of a region R is the set of points in R such that every trajectory segment starting in such point reaches e without traversing any other edge. More formally, let R be a region and e be an edge in $out(R)$, then the *e-extended* $\forall$*-predecessor* of I, $\widetilde{\overline{\mathsf{Pre}}}_e(I)$ is defined as: $\widetilde{\overline{\mathsf{Pre}}}_e(I) = \{\mathbf{x} \mid \forall \xi \,.\, (\xi(0) = \mathbf{x} \Rightarrow \exists t \geq 0 \,.\, (\xi(t) \in I \wedge \mathsf{Sig}(\xi[0,t]) = e))\}$.

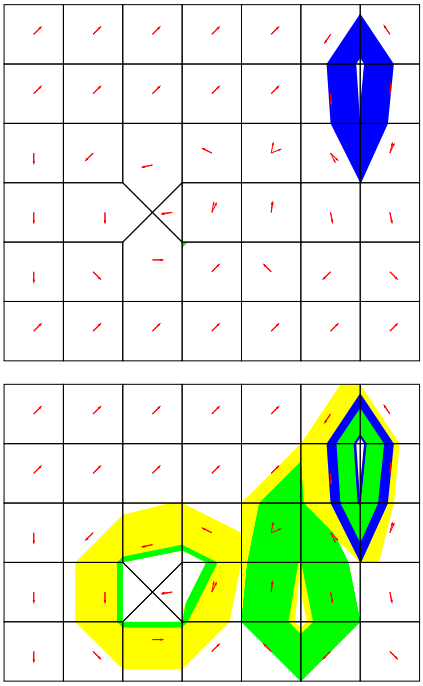

Figure 3: (a) Invariance kernel; (b) All the kernels

It is easy to see that $\widetilde{\overline{\mathsf{Pre}}}_\sigma(I)$ is a polygonal subset of the plane which can be calculated using a similar procedure as for $\overline{\mathsf{Pre}}_\sigma(I)$. We compute the invariance kernel of K_σ as follows:

Theorem 5 *If σ is STAY then* $\mathsf{Inv}(K_\sigma) = \widetilde{\overline{\mathsf{Pre}}}_\sigma(\widetilde{\overline{\mathsf{Pre}}}_\sigma(J_\sigma))$, *otherwise it is* $\emptyset$. □

Example 9 *Fig. 3-(a) shows the unique invariance kernel of the SPDI given in Example 1.* ■

An interesting property of invariance kernels is that the limits are included in the invariance kernel, i.e. $[l^*, u^*] \subseteq \mathsf{Inv}(K_\sigma)$. In other words:

Proposition 4 *The set delimited by the polygons defined by the interval $[l^*, u^*]$ is an invariance set of STAY cycles.* □

The following result relates controllability and invariance kernels.

Proposition 5 *If σ is STAY then* $\mathsf{Cntr}(K_\sigma) \subseteq \mathsf{Inv}(K_\sigma)$. □

Example 10 *Fig. 3-(b) shows the viability, controllability and invariance kernels of the SPDI given in Example 1. For any point in the viability kernel of a cycle there exists a trajectory which will converge to its controllability kernel (proposition 2). It is possible to see in the picture that* $\mathsf{Cntr}(\cdot) \subset \mathsf{Inv}(.)$ *(proposition 5). All the above pictures has been obtained with the toolbox SPeeDI$^+$ [14].* ■

In a similar way as for the controllability kernel, we define $\mathsf{Inv}^l(K_\sigma)$ and $\mathsf{Inv}^u(K_\sigma)$.

2.5.4 Kernel Properties

Controllability and viability kernels can be related together in the following manner.

Definition 1 *Given a controllability kernel C (of a loop σ — $C = \mathsf{Cntr}(K_\sigma)$), then let C^+ be the related viability kernel ($C^+ = \mathsf{Viab}(K_\sigma)$), C_{in} be the inside of the kernel, and C_{out} be the outside.*

Proposition 3 in [18] gives conditions for feasible trajectories traversing controllability kernels. The following is a generalization of such result:

Proposition 6 *Given two edges e and e', one lying completely inside a kernel, and the other outside or on the same kernel, such that ee' is feasible, then there exists a point on the kernel, which is reachable from e and from which e' is reachable.* □

The following corollary follows from [18, Proposition 2], asserting that the controllability kernel is the local basin of attraction of the viability kernel:

Corollary 1 *Given an controllability kernel C, and related viability kernel C^+, then for any $e \subseteq C^+$, $e' \subseteq C$, there exists a feasible path $e\sigma e'$.* □

2.6 Semi-Separatrix Curves

In this section we define the notion of *separatrix curves*, which are curves dissecting the plane into two mutually non-reachable subsets, and *semi-separatrix curves* which can only be crossed in one direction. All the proofs of this and forthcoming sections may be found in [17]. We start by defining these notions independently of SPDIs.

Definition 2 *Let $K \subseteq \mathbb{R}^2$. A* separatrix *in K is a closed curve γ partitioning K into three sets K_A, K_B and γ itself, such that K_A, K_B and γ are pairwise disjoint, $K = K_A \cup K_B \cup \gamma$ and the following conditions hold: (1) For any point $\mathbf{x}_0 \in K_A$ and trajectory ξ, with $\xi(0) = \mathbf{x}_0$, there is no t such that $\xi(t) \in K_B$; and (2) For any point $\mathbf{x}_0 \in K_B$ and trajectory ξ, with $\xi(0) = \mathbf{x}_0$, there is no t such that $\xi(t) \in K_A$. If only one of the above conditions holds then we say that the curve is a* semi-separatrix*. If only condition 1 holds, then we say that K_A is the* inner *of γ (written γ_{in}) and K_B is the* outer *of γ (written γ_{out}). If only condition 2 holds, K_B is the* inner *and K_A is the* outer *of γ.*

Notice that, as in the case of the controllability kernel, an edge of the SPDI may be split into two by a semi-separatrix — part inside, and part outside. As before, we can split the edge into parts, such that each part is completely inside, or completely outside the semi-separatrix.

The above notions are extended to SPDIs straightforwardly. The set of all the separatrices of an SPDI $\mathcal{S}$ is denoted by $\mathsf{Sep}(\mathcal{S})$, or simply Sep.

Now, let $\sigma = e_1 \dots e_n e_1$ be a simple cycle, $\angle^{\mathbf{b}_i}_{\mathbf{a}_i}$ $(1 \leq i \leq n)$ be the dynamics of the regions for which e_i is an entry edge and $I = [l, u]$ an interval on edge e_1. Remember that $\mathsf{Succ}_{e_1e_2}(I) = F(I \cap S_1) \cap J_1$, where $F(x) = [a_1x + b_1, a_2x + b_2]$. Let $\mathbf{l}$ be the vector corresponding to the point on e_1 with local coordinates l and $\mathbf{l}'$ be the vector corresponding to the point on e_2 with local coordinates $F(l)$ (similarly, we define $\mathbf{u}$ and $\mathbf{u}'$ for $F(u)$). We define first $\overline{\mathsf{Succ}}^{\mathbf{b}_1}_{e_1}(I) = \{\mathbf{l} + \alpha(\mathbf{l}' - \mathbf{l}) \mid 0 < \alpha < 1\}$ and $\overline{\mathsf{Succ}}^{\mathbf{a}_1}_{e_1}(I) = \{\mathbf{u} + \alpha(\mathbf{u}' - \mathbf{u}) \mid 0 < \alpha < 1\}$. We extend these definitions in a straight way to any (cyclic) signature $\sigma = e_1 \dots e_n e_1$, denoting them by $\overline{\mathsf{Succ}}^{\mathbf{b}}_{\sigma}(I)$ and $\overline{\mathsf{Succ}}^{\mathbf{a}}_{\sigma}(I)$, respectively; we can compute them similarly as for $\overline{\mathsf{Pre}}$. Whenever applied to the fixpoint $I^* = [l^*, u^*]$, we denote $\overline{\mathsf{Succ}}^{\mathbf{b}}_{\sigma}(I^*)$ and $\overline{\mathsf{Succ}}^{\mathbf{a}}_{\sigma}(I^*)$ by ξ^l_σ and ξ^u_σ respectively. Intuitively, ξ^l_σ (ξ^u_σ) denotes the piece-wise affine closed curve defined by the leftmost (rightmost) fixpoint l^* (u^*).

We show now how to identify semi-separatrices for simple cycles.

Theorem 6 *Given an SPDI, let σ be a simple cycle, then the following hold:*

1. *If σ is EXIT-RIGHT then ξ^l_σ is a semi-separatrix curve (filtering trajectories from "left" to "right");*
2. *If σ is EXIT-LEFT then ξ^u_σ is a semi-separatrix curve (filtering trajectories from "right" to "left");*
3. *If σ is STAY, then the two polygons defining the invariance kernel ($\mathsf{Inv}^l(K_\sigma)$ and $\mathsf{Inv}^u(K_\sigma)$), are semi-separatrices.* □

In the case of STAY cycles, ξ^l_σ and ξ^u_σ are both also semi-separatrices. Notice that in the above result, computing a semi-separatrix depends only on one simple cycle, and the corresponding algorithm is then reduced to find simple cycles in the SPDI and checking whether it is STAY, EXIT-RIGHT or EXIT-LEFT. DIE cycles induce an infinite number of semi-separatrices and are not treated in this setting.

Example 11 *Fig. 4 shows all the semi-separatrices of the SPDI given in Example 1, obtained as shown in Theorem 6. The small arrows traversing the semi-separatrices show the inner and outer of each semi-separatrix: a trajectory may traverse the semi-separatrix following the direction of the arrow, but not vice-versa.* ■

The following two results relate feasible signatures and semi-separatrices.

Proposition 7 *If, for some semi-separatrix γ, $e \in \gamma_{in}$ and $e' \in \gamma_{out}$, then the signature ee' is not feasible.* □

Proposition 8 *If, for some semi-separatrix γ, and signature σ (of at least length 2), then, if $head(\sigma) \in \gamma_{in}$ and $last(\sigma) \in \gamma_{out}$, σ is not feasible.* □

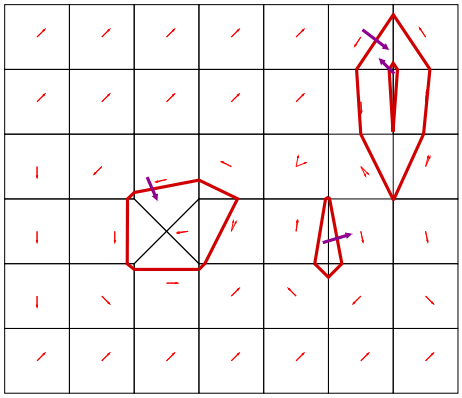

Figure 4: Semi-separatrices

3. STATE-SPACE REDUCTION

3.1 Reduction using Semi-Separatrices

Semi-separatrices partition the state space into two parts[3] – once one crosses such a border, all states outside the region can be ignored. We present a technique, which, given an SPDI and a reachability question, enables us to discard portions of the state space based on this information. The approach is based on identifying *inert* states (edges in the SPDI) not playing a role in the reachability analysis.

Definition 3 *Given an SPDI $\mathcal{S}$, a semi-separatrix $\gamma \in \mathsf{Sep}$, a source edge $e0$ and a destination edge $e1$, an edge e is said to be* inert *if it lies outside the semi-separatrix while $e0$ lies inside, or it lies inside, while $e1$ lies outside:*

$$inert^{\gamma}_{e0\to e1} = \{e : \mathcal{E} \mid e0 \in \gamma_{in} \wedge e \in \gamma_{out}\} \cup \{e : \mathcal{E} \mid e1 \in \gamma_{out} \wedge e \in \gamma_{in}\}.$$

We can prove that these inert edges can never appear in a feasible signature:

Lemma 2 *Given an SPDI $\mathcal{S}$, a semi-separatrix γ, a source edge $e0$ and a destination edge $e1$, and a feasible signature $e0\sigma e1$ in $\mathcal{S}$. No inert edge from $inert^{\gamma}_{e0\to e1}$ may appear in $e0\sigma e1$.* □

Given an SPDI, we can reduce the state space by discarding inert edges.

Definition 4 *Given an SPDI $\mathcal{S}$, a semi-separatrix γ, and two edges, a source edge $e0$ and a destination edge $e1$, we define the reduced SPDI $\mathcal{S}^{\gamma}_{e0\to e1}$ to be the same as $\mathcal{S}$ but without the inert edges.*

Clearly, the resulting SPDI is not bigger than the original one. Finally, we prove that checking reachability on the reduced SPDI is equivalent to checking reachability on the original SPDI:

[3]Here, we do not consider the semi-separatrix itself.

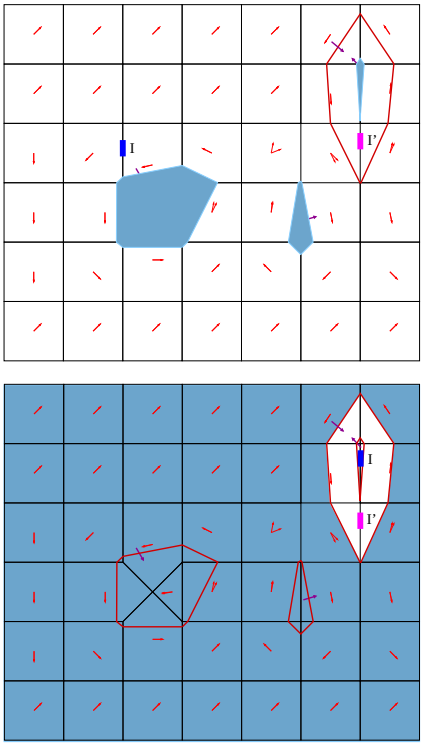

Figure 5: Reduction using semi-separatrices

Theorem 7 *Given an SPDI $\mathcal{S}$, a semi-separatrix γ, and edges $e0$ and $e1$, then, $e1$ is reachable from $e0$ in $\mathcal{S}$ if and only if $e1$ is reachable from $e0$ in $\mathcal{S}^{\gamma}_{e0\to e1}$.* □

We have shown, that once semi-separatrices are identified, given a reachability question, we can reduce the size of the SPDI to be verified by removing inert edges of all the known semi-separatrices.

Example 12 *The shaded areas of Fig. 5 (a) and (b) are examples of subsets of the SPDI edges of the reachability graph, eliminated by the reduction presented in this section applied to all semi-separatrices, when answering reachability questions (in this case to the question: Is I' reachable from I?).* ■

This result enables us to verify SPDIs much more efficiently. It is important to note that model-checking an SPDI requires identification of simple loops, which means that the calculation of the semi-separatrices is not more expensive than the initial pass of the model-checking algorithm. Furthermore, we can perform this analysis only once for an SPDI and store the information to be used in any reachability analysis on that SPDI. Reduction, however, can only be applied once we know the source and destination states.

3.2 State-space Reduction using Kernels

We have already shown that any invariant set is essentially a pair of semi-separatices, and since the invariance kernel is an invariant set, we can use the results from section 2.6

to abstract an SPDI using invariance kernels. We now turn our attention to state space reduction using controllability kernels:

Definition 5 *Given an SPDI $\mathcal{S}$, a loop σ, a source edge $e0$ and a destination edge $e1$, an edge e is said to be* redundant *if it lies on the opposite side of a controllability kernel as both $e0$ and $e1$:*

$$redundant^{\sigma}_{e0\rightarrow e1}$$
$$\{e : \mathcal{E} \mid \{e0,\ e1\} \subseteq \mathsf{Cntr}_{in}(\sigma) \cup \mathsf{Cntr}(\sigma) \wedge e \in \mathsf{Cntr}_{out}(\sigma)\} \cup$$
$$\{e : \mathcal{E} \mid \{e0,\ e1\} \subseteq \mathsf{Cntr}_{out}(\sigma) \cup \mathsf{Cntr}(\sigma) \wedge e \in \mathsf{Cntr}_{in}(\sigma)\}$$

We can prove that we can do without these edges to check feasibility:

Lemma 3 *Given an SPDI $\mathcal{S}$, a loop σ, a source edge $e0$, a destination edge $e1$, and a feasible signature $e0\sigma e1$ then there exists a feasible signature $e0\sigma' e1$ such that σ' contains no redundant edge from $redundant^{\sigma}_{e0\rightarrow e1}$.* □

Given an SPDI, we can reduce the state space by discarding redundant edges.

Definition 6 *Given an SPDI $\mathcal{S}$, a loop σ, a source edge $e0$ and a destination edge $e1$, we define the reduced SPDI $\mathcal{S}^{\sigma}_{e0\rightarrow e1}$ to be the same as $\mathcal{S}$ but without redundant edges.*

Clearly, the resulting SPDI is smaller than the original one. Finally, based on proposition 3, we prove that reachability on the reduced SPDI is equivalent to reachability on the original one:

Theorem 8 *Given an SPDI $\mathcal{S}$, a loop σ, a source edge $e0$ and a destination edge $e1$, then, $e1$ is reachable from $e0$ in $\mathcal{S}$ if and only if $e1$ is reachable from $e0$ in $\mathcal{S}^{\sigma}_{e0\rightarrow e1}$.* □

Given a loop which has a controllability kernel, we can thus reduce the state space to explore. In practice, we apply this state space reduction for each controllability kernel in the SPDI. Once a loop in the SPDI is identified, it is straightforward to apply the reduction algorithm.

3.3 Immediate Answers

By definition of the controllability kernel, any two points inside it are mutually reachable. This can be used to answer reachability questions in which both the source and destination edge lie (possibly partially) within the same controllability kernel. Using proposition 2, we know that any point in the viability kernel of a loop can eventually reach the controllability kernel of the same loop, which allows us to relax the condition about the source edge to just check whether it (partially) lies within the viability kernel. Finally, we note that the union of non-disjoint controllability sets is itself a controllability set which allows us to extend the result to work for a collection of loops whose controllability kernels form a strongly connected set.

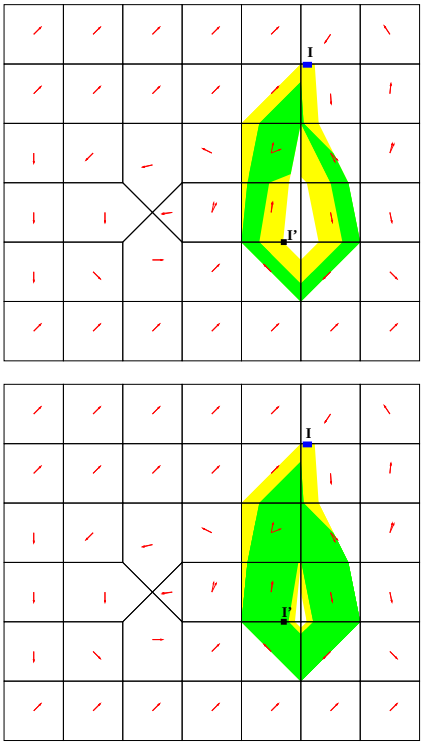

Figure 6: Answering reachability using kernels

Definition 7 *We extend viability and controllability kernels for a set of loops Σ by taking the union of the kernels of the individual loops, with $\mathsf{Viab}(K_\Sigma)$ being the union of all viability kernels of loops in Σ, and similarly $\mathsf{Cntr}(K_\Sigma)$.*

Definition 8 *Two loops σ and σ' are said to be compatible ($\sigma \leftrightsquigarrow \sigma'$) if their controllability kernels overlap: $\mathsf{Cntr}(K_\sigma) \cap \mathsf{Cntr}(K_{\sigma'}) \neq \emptyset$.*

We extend the notion of compatibility to a set of loops Σ to mean that all loops in the set are transitively compatible: $\forall \sigma,\ \sigma' \in \Sigma \cdot \sigma \leftrightsquigarrow^* \sigma'$.

Based on proposition 2, we can prove the following:

Theorem 9 *Given a source edge e_{src} and a destination edge e_{dst}, if for some compatible set of loops Σ, we know that $e_{src} \cap \mathsf{Viab}(K_\Sigma) \neq \emptyset$ and $e_{dst} \cap \mathsf{Cntr}(K_\Sigma) \neq \emptyset$, then e_{dst} is reachable from e_{src}.* □

Example 13 *Fig. 6-(a) shows a viability and a controllability kernel of a cycle and two intervals I and I'. Whether I' is reachable from I cannot be answered immediately in this case, but Fig. 6-(b) shows the overlapping of the viability and controllability kernels depicted in Fig. 6-(a) with the kernels of an inner cycle. I' thus lies in a compatible controllability kernel, and we can immediately conclude (by theorem 9) that I' is reachable from I.* ■

In practice, we propose to use these theorems to enable answering certain reachability questions without having to ex-

plore the complete state space. It can also be used to reduce reachability questions to (possibly) simpler ones by trying to reach a viability kernel rather than a particular edge. As in the case of semi-separatrices, a preliminary analysis of an SPDI's kernels be used in all subsequent reachability queries. SPeeDI [14] starts by calculating and caching all loops in the given SPDI, and can thus easily identify maximal compatible sets of loops. Combining this technique with the semi-separatrix reduction technique we envisage substantial gains.

4. COMPOSITIONAL ANALYSIS

4.1 SPDI Decomposition

In this section, we propose a number of theorems which, given an SPDI and a reachability question, for each controllability kernel, allow us to either (i) answer the reachability question without any further analysis; or (ii) reduce the state space necessary for reachability analysis; or (iii) decompose the reachability question into two smaller, and independent reachability questions.

The following theorem enables us to answer certain reachability questions without any analysis, other than the identification of controllability and viability kernels. This result is based on two properties, that within the controllability kernel of a loop, any two points are mutually reachable, and that any point on the viability kernel of the same loop can eventually reach the controllability kernel. Therefore if the source edgelist lies (possibly partially) within the viability kernel of a loop, and the destination edgelist lies (possibly partially) within the controllability kernel of the same loop, then, there must exist a trajectory from the source to the destination edgelist. The full proof of this result can be found in [18].

Theorem 10 *Given an SPDI $\mathcal{S}$, two edgelists I and I' and a controllability kernel C, then if $I \subseteq C^+$ and $I' \subseteq C$, then $I \xrightarrow{\mathcal{S}} I'$.* $\square$

The following theorem allows us to reduce the state space based on controllability kernels. If both the source and destination edgelists lie on the same side of a controllability kernel, then we can disregard all edges on the other side of the kernel. The full proof of this result can be found in [18].

Theorem 11 *Given an SPDI $\mathcal{S}$, two edgelists I and I' and a controllability kernel C, then if $I \subseteq C_{in}$ and $I' \subseteq C_{in}$, then $I \xrightarrow{\mathcal{S}} I'$ if and only if $I \xrightarrow{\mathcal{S}\setminus C_{out}} I'$. Similarly, if $I \subseteq C_{out}$ and $I' \subseteq C_{out}$, then $I \xrightarrow{\mathcal{S}} I'$ if and only if $I \xrightarrow{\mathcal{S}\setminus C_{in}} I'$.* $\square$

Finally, the following new result allows us to decompose a reachability question into two smaller questions independent of each other. The theorem states that if the source and destination edgelists lie on opposite sides of a controllability kernel, then we can try (i) to reach the related viability kernel from the source edgelist, and (ii) to reach the destination from the controllability kernel. The original reachability question can be answered affirmatively if and only if both these questions are answered affirmatively.

Theorem 12 *Given an SPDI $\mathcal{S}$, two edgelists I and I' and a controllability kernel C, then if $I \subseteq C_{in}$ and $I' \subseteq C_{out}$, then $I \xrightarrow{\mathcal{S}} I'$ if and only if $I \xrightarrow{\mathcal{S}\setminus C_{out}} C^+ \wedge C \xrightarrow{\mathcal{S}\setminus C_{in}} I'$. Similarly, if $I \subseteq C_{out}$ and $I' \subseteq C_{in}$, then $I \xrightarrow{\mathcal{S}} I'$ if and only if $I \xrightarrow{\mathcal{S}\setminus C_{in}} C^+ \wedge C \xrightarrow{\mathcal{S}\setminus C_{out}} I'$.*

4.2 Unavoidable Kernels

Unavoidable kernels are defined geometrically to be kernels which a straight line from the source interval to the destination interval 'intersects' an odd number of times. We call the kernel unavoidable since it can be proved that any path from the source to the destination will have to pass through the kernel.

Definition 9 *Given an SPDI $\mathcal{S}$ and two edgelists I and I', we say that a controllability kernel $\mathsf{Cntr}(K_\sigma)$ is* unavoidable *if any segment of line with extremes on points lying on I and I' intersects with both the edges of $\mathsf{Cntr}^l(K_\sigma)$ and those of $\mathsf{Cntr}^u(K_\sigma)$ an odd number of times (disregarding tangential intersections with vertices).*

The following theorem enables us to discover separating controllability kernels using a simple geometric test.

Theorem 13 *Given an SPDI $\mathcal{S}$, two edgelists I and I', and a controllability kernel $C = \mathsf{Cntr}(K_\sigma)$, then C is an unavoidable kernel if and only if one of the following conditions holds (i) $I \subseteq C_{in}$ and $I' \subseteq C_{out}$; or (ii) $I \subseteq C_{out}$ and $I' \subseteq C_{in}$.*

Corollary 2 *Given an SPDI $\mathcal{S}$, two edgelists I and I', and an unavoidable controllability kernel $C = \mathsf{Cntr}(K_\sigma)$, then $I \xrightarrow{\mathcal{S}} I'$ if and only if $I \xrightarrow{\mathcal{S}} C$ and $C \xrightarrow{\mathcal{S}} I'$.*

The following result relates unavoidable kernels:

Proposition 9 *Given two disjoint controllability kernels C and C', both unavoidable from I to I', then either C' is unavoidable from I to C or C' is unavoidable from C to I', but not both.*

4.3 Parallel Reachability Algorithm

In Fig. 8 we give an algorithm for parallel reachability analysis of SPDIs using parallel recursive calls corresponding to independent reachability questions.

The function `ReachParKernels` is called with the SPDI to consider, a list of kernels still to be used for reduction, and the source and destination edgelists. With no kernels to consider, the algorithm simply calls the standard sequential algorithm (`Reach`). Otherwise, one of the kernels is analyzed, with three possible cases:

1. If the source lies (possibly partially) on the extended kernel, and the destination lies (possibly partially) on the kernel, then we can give an immediate answer (using theorem 10).

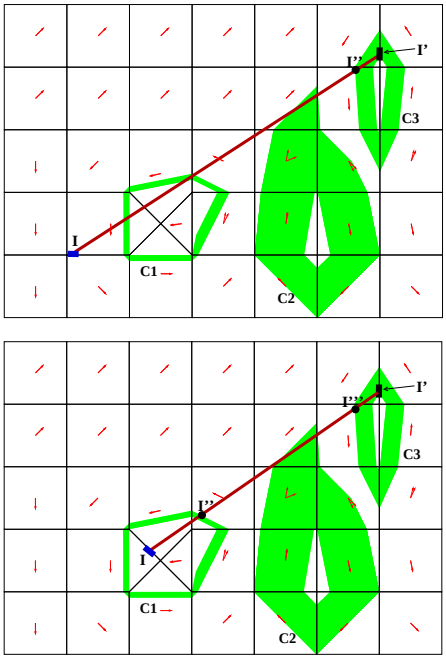

Figure 7: Unavoidable kernels and independent reachability questions

2. If both the edgelists lie on the same side of the kernel, then we simply eliminate redundant parts of the SPDI — anything on the other side of the kernel (theorem 11).

3. Otherwise, if the kernels both lie on opposite sides of the kernel, we can split the problem into two independent questions (reaching the kernel from the source, and the destination from the kernel) which can be run in parallel (theorem 12). An affirmative answer from both these subquestions is equivalent to an affirmative answer to the original question.

Note that the function `ReachParKernels` is compositional in the sense that each recursive call launch a process which operates in (most cases in) disjoint state spaces which are smaller than the original one ($\mathcal{S}$). The final answer is the composition of the partial reachability questions.

Given two edgelists I and I', we define the following predicate $I \xrightarrow{\mathcal{S}}_{\|} I' \equiv \texttt{ReachPar}(\mathcal{S}, I, I')$. The following theorem states that the (compositional) parallel algorithm exactly answers the reachability question, also giving a soundness and completeness proof of the algorithm:

Theorem 14 *Given an SPDI $\mathcal{S}$ and two intervals $I \subseteq e$ and $I' \subseteq e'$, $I \xrightarrow{\mathcal{S}} I'$ if and only if $I \xrightarrow{\mathcal{S}}_{\|} I'$.*

5. CONCLUDING REMARKS

We have given an overview of our recent results on the optimisation of SPDI reachability analysis. Using semi-separatrices and kernels, we presented techniques to improve reachability analysis on SPDIs. In all cases, the techniques require the identification and analysis of loops in the SPDI. We note that most of this information is still required in reachability analysis, and thus no extra work is required to perform the optimization presented in this paper. We have also shown how answering reachability on an SPDI can be reduced to a number of smaller reachability questions. These two techniques can be combined together applying the reduction techniques globally (on the whole SPDI) or locally on the decomposed SPDIs).

```
function ReachPar(S, I, I') =
  ReachParKernels (S, ControllabilityKernels(S), I, I')

function ReachParKernels(S, [], I, I') =
  Reach(S, I, I');

function ReachParKernels(S, k:ks, I, I') =
  if (ImmediateAnswer(S, I, I')) then
     True;
  elsif (SameSideOfKernel(S, k, I, I')) then
     S_I := S \ EdgesOnOtherSideOf(S, k, I');
     ReachParKernels(S_I, ks, I, I');
  else
     S_I := S \ EdgesOnOtherSideOf(S, k, I);
     S_I' := S \ EdgesOnOtherSideOf(S k, I');
     parbegin
        r1 := ReachParKernels(S_I, ks, I, viability(k));
        r2 := ReachParKernels(S_I', ks, k, I');
     parend;
     return (r1 and r2);
```

Figure 8: Parallel algorithm for reachability of SPDIs.

Our work is obviously restricted to planar systems, which enables us to compute these kernels exactly. In higher dimensions and hybrid systems with higher complexity, calculation of kernels is not computable. Other work is thus based on calculations of approximations of these kernels (e.g., [8, 7, 19]). We are not aware of any work using kernels and semi-separatrices to reduce the state-space of the reachability graph as presented in this paper.

We are currently exploring the implementation of the optimizations presented in this paper to improve the efficiency of SPeeDI$^+$ [14]. We are also investigating other applications of these kernels in the model-checking of SPDIs.

One current research direction is to apply our results to semi-decide the reachability question for SPDIs defined on 2-dimensional manifolds, for which the decidability of reachability remains unresolved [3]. Maybe the most prominent application of SPDIs is for approximating complex non-linear differential equations on the plane, for which an exact solution is not known. The decidability of SPDIs reachability and of its phase portrait construction would be of invaluable help for the qualitative analysis of such equations. The challenge would be to find an "intelligent" partition of the plane in order to get an optimal approximation of the equations. Since such partition might produce a high number of regions, our parallel algorithm might be extremely useful here.

6. REFERENCES

[1] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[2] E. Asarin, G. Pace, G. Schneider, and S. Yovine. SPeeDI: a verification tool for polygonal hybrid systems. In *CAV'2002*, volume 2404 of *LNCS*, 2002.

[3] E. Asarin and G. Schneider. Widening the boundary between decidable and undecidable hybrid systems. In *CONCUR'2002*, volume 2421 of *LNCS*, 2002.

[4] E. Asarin, G. Schneider, and S. Yovine. On the decidability of the reachability problem for planar differential inclusions. In *HSCC'2001*, number 2034 in LNCS, pages 89–104, 2001.

[5] E. Asarin, G. Schneider, and S. Yovine. Towards computing phase portraits of polygonal differential inclusions. In *HSCC'02*, volume LNCS 2289, 2002.

[6] J.-P. Aubin. The substratum of impulse and hybrid control systems. In *HSCC'01*, volume 2034 of *LNCS*, pages 105–118. Springer, 2001.

[7] J.-P. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, and N. Seube. Towards a viability theory for hybrid systems. In *European Control Conference*, 2001.

[8] J.-P. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, and N. Seube. Viability and invariance kernels of impulse differential inclusions. In *Conference on Decision and Control*, volume 40 of *IEEE*, pages 340–345, December 2001.

[9] A. Deshpande and P. Varaiya. Viable control of hybrid systems. In *Hybrid Systems II*, number 999 in LNCS, pages 128–147, 1995.

[10] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *STOC'95*, pages 373–382. ACM Press, 1995.

[11] G. Lafferriere, G. Pappas, and S. Yovine. Symbolic reachability computation of families of linear vector fields. *Journal of Symbolic Computation*, 32(3):231–253, Sept. 2001.

[12] O. Maler and A. Pnueli. Reachability analysis of planar multi-linear systems. In *CAV'93*, pages 194–209. LNCS 697, Springer Verlag, July 1993.

[13] A. Matveev and A. Savkin. *Qualitative theory of hybrid dynamical systems*. Birkhäuser Boston, 2000.

[14] G. Pace and G. Schneider. SPeeDI$^+$. `http:\\www.cs.um.edu.mt\speedi`.

[15] G. Pace and G. Schneider. Model checking polygonal differential inclusions using invariance kernels. In *VMCAI'04*, number 2937 in LNCS, pages 110–121. Springer Verlag, December 2003.

[16] G. Pace and G. Schneider. Compositional algorithm for parallel model checking of polygonal hybrid systems. In *3rd International Colloquium on Theoretical Aspects of Computing (ICTAC'06)*, volume 4281 of *LNCS*. Springer-Verlag, 2006.

[17] G. Pace and G. Schneider. Static analysis for state-space reduction of polygonal hybrid systems. In *Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *LNCS*. Springer-Verlag, 2006.

[18] G. Pace and G. Schneider. Static analysis of SPDIs for state-space reduction. Technical Report 336, Department of Informatics, University of Oslo, PO Box 1080 Blindern, N-0316 Oslo, Norway, April 2006.

[19] P. Saint-Pierre. Hybrid kernels and capture basins for impulse constrained systems. In *HSCC'02*, volume 2289 of *LNCS*. Springer-Verlag, 2002.

[20] G. Schneider. *Algorithmic Analysis of Polygonal Hybrid Systems*. PhD thesis, VERIMAG – UJF, Grenoble, France, July 2002.

[21] G. Schneider. Computing invariance kernels of polygonal hybrid systems. *Nordic Journal of Computing*, 11(2):194–210, 2004.

[22] S. Simić, K. Johansson, S. Sastry, and J. Lygeros. Towards a geometric theory of hybrid systems. In *HSCC'00*, number 1790 in LNCS, 2000.