Compositional Abstraction and Safety Synthesis using Overlapping Symbolic Models

Pierre-Jean Meyer, Antoine Girard, Emmanuel Witrant

Compositional abstraction and safety synthesis using overlapping symbolic models

Pierre-Jean Meyer, Antoine Girard, and Emmanuel Witrant

***Abstract*—In this paper, we develop a compositional approach to abstraction and safety synthesis for a general class of discrete time nonlinear systems. Our approach makes it possible to define a symbolic abstraction by composing a set of symbolic subsystems that are overlapping in the sense that they can share some common state variables. We develop compositional safety synthesis techniques using such overlapping symbolic subsystems. Comparisons, in terms of conservativeness and of computational complexity, between abstractions and controllers obtained from different system decompositions are provided. Numerical experiments show that the proposed approach for symbolic control synthesis enables a significant complexity reduction with respect to the centralized approach, while reducing the conservatism with respect to compositional approaches using non-overlapping subsystems.**

I. Introduction

Symbolic control deals with the use of discrete synthesis techniques for controlling complex continuous or hybrid systems [5], [26]. In such approaches, one relies on symbolic abstractions of the orignal system; i.e. dynamical systems with finitely many state and input values, each of which symbolizes sets of states and inputs of the concrete system [2]. This enables the use of discrete controller synthesis techniques, such as supervisory control [9] or algorithmic game theory [7], which allows us to address high-level specifications such as safety, reachability or more general properties specified by automata or temporal logic formula [4]. When the behaviors of the concrete system and of its abstraction are related by some formal inclusion relationship (such as alternating simulation [26] or feedback refinement relations [25]), the discrete controller of the abstraction can be refined to control the concrete system, with guarantees of correctness.

Several approaches exist for computing symbolic abstractions for a wide range of dynamical systems (see e.g. [27], [21], [31], [30], [10], [25]), based on partitions or discretizations of the state and input spaces. The numbers of symbolic states and inputs are then typically exponential in the dimension of the concrete state and input spaces, respectively. This limits the application of these approaches to low-dimensional systems. Several works have been done for improving the scalability of symbolic control. In [17], [29], an approach, which does not require state space discretization, has been presented for computing symbolic abstractions of incrementally stable systems. In [20], [13], algorithms combining discrete controller synthesis with on-the-fly computation of symbolic abstractions have been developed. Compositional approaches have also been explored in several papers [28], [24], [19], [8], [15], [11], [23], [22]. In such approaches, a system with a control specification is decomposed into subsystems with local control specifications. Then, for each subsystem, a symbolic abstraction can be computed and a local controller is synthesized while assuming that the other subsystems meet their local specifications. This approach, called *assume-guarantee* reasoning [14], enables the use of symbolic control techniques for higher dimensional systems.

In this paper, we develop a novel compositional approach for symbolic control synthesis for a general class of discrete time nonlinear systems. Our approach clearly differs from the previously mentioned works (and particularly from our previous work [19]) by the possibility for subsystems to share common state variables through the definition for each subsystem of locally modeled but uncontrolled variables, which are accessible to the local controller. Hence, this makes it possible for local controllers to share information on some of the states of the system. In this setting, we develop compositional approaches for computing symbolic abstractions and synthesizing controllers that maintain the state of the system in some specified safe set.

The paper is organized as follows. Section II introduces the class of systems, safety controllers and the abstraction framework considered in the paper. Section III presents a compositional approach for computing abstractions from symbolic subsystems with overlapping sets of states. Compositional controller synthesis is addressed in Section IV. Section V provides results to compare abstractions and controllers obtained from different system decompositions, and a discussion on the computational complexity of the approach. Numerical experiments are then reported in Section VI.

II. Preliminaries

A. System description

We consider a class of discrete time nonlinear control systems modeled by the difference inclusion:

$$x(t+1) \in F(x(t),u(t)),\ t \in \mathbb{N} \tag{1}$$

where $\mathbb{N} = \{0,1,2,\dots\}$, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathcal{U} \subseteq \mathbb{R}^p$ denote the state and the control input, respectively, and the set-valued map $F : \mathbb{R}^n \times \mathcal{U} \to 2^{\mathbb{R}^n}$. System (1) is discrete time; however,

P.-J. Meyer is with KTH Royal Institute of Technology, Department of Automatic Control, 10044 Stockholm, Sweden (email: pjmeyer@kth.se).

A. Girard is with Laboratoire des signaux et systèmes (L2S), CNRS, CentraleSupélec, Université Paris-Sud, Université Paris-Saclay, 3, rue Joliot-Curie, 91192 Gif-sur-Yvette, cedex, France (email: Antoine.Girard@l2s.centralesupelec.fr). His work was partially supported by the laboratory of excellence DigiCosme (CODECSYS project).

E. Witrant is with Univ. Grenoble Alpes/CNRS, GIPSA-Lab, F-38000 Grenoble, France (email: Emmanuel.Witrant@ujf-grenoble.fr).

it encompasses sampled versions of continuous time systems, possibly subject to disturbances (see e.g. [19], [25]).

Throughout the paper, we assume, for simplicity, that for all $x \in \mathbb{R}^n$, $u \in \mathcal{U}$, $F(x,u) \neq \emptyset$. For a subset of states $\mathcal{X}' \subseteq \mathbb{R}^n$ and inputs $\mathcal{U}' \subseteq \mathcal{U}$ we denote

$$F(\mathcal{X}',\mathcal{U}') = \bigcup_{x \in \mathcal{X}', u \in \mathcal{U}'} F(x,u).$$

Exact computation of $F(\mathcal{X}',\mathcal{U}')$ may not always be possible, especially when (1) corresponds to the sampled dynamics of a continuous time system. Therefore, we will assume throughout the paper that we are able to compute, for all sets of states $\mathcal{X}' \subseteq \mathbb{R}^n$ and of inputs $\mathcal{U}' \subseteq \mathcal{U}$, a set $\overline{F}(\mathcal{X}',\mathcal{U}')$ verifying

$$F(\mathcal{X}',\mathcal{U}') \subseteq \overline{F}(\mathcal{X}',\mathcal{U}'). \tag{2}$$

Several methods exist for computing such over-approximations for linear [12], [16], [18] and nonlinear [6], [1], [10], [25] systems.

B. Transition systems and safety controllers

A *transition system* is defined as a triple $S = (X, U, \delta)$ consisting of:

- a set of states X;
- a set of inputs U;
- a transition map $\delta : X \times U \to 2^X$.

A transition $x' \in \delta(x,u)$ means that S can evolve from state x to state x' under input u. $U(x)$ denotes the set of enabled inputs at state x: i.e. $u \in U(x)$ if and only if $\delta(x,u) \neq \emptyset$. A trajectory of S is a finite or infinite sequence of transitions $(x^0, u^0, x^1, u^1, \dots)$ such that $x^{t+1} \in \delta(x^t, u^t)$, for $t \in \mathbb{N}$

In the following, we consider a safety synthesis problem for transition system S: let $\mathcal{X} \subseteq X$ be a subset of safe states, a *safety controller* for system S and safe set $\mathcal{X}$ is a map $C : X \to 2^U$ such that:

- for all $x \in X$, $C(x) \subseteq U(x)$;
- its domain $dom(C) = \{x \in X | \ C(x) \neq \emptyset\} \subseteq \mathcal{X}$;
- for all $x \in dom(C)$ and $u \in C(x)$, $\delta(x,u) \subseteq dom(C)$.

Essentially, a safety controller makes it possible to generate infinite trajectories of S, $(x^0, u^0, x^1, u^1, \dots)$ such that $x^t \in \mathcal{X}$, for all $t \in \mathbb{N}$ as follows: $x^0 \in dom(C)$, $u^t \in C(x^t)$ and $x^{t+1} \in \delta(x^t, u^t)$, for all $t \in \mathbb{N}$. It is known (see e.g. [26]) that there exists a *maximal safety controller* C^* for system S and safe sate $\mathcal{X}$ such that for all safety controllers C, for all $x \in X$, it holds $C(x) \subseteq C^*(x)$.

C. Feedback refinement relations

Complex transition systems motivate the use of abstractions, since finding a control strategy for an abstraction is generally simpler than for the original system. However, to derive a controller for the original system from that of the abstraction, the systems must satisfy a formal behavioral relationship such as alternating simulation [26]. In this paper, we will rely on the notion of feedback refinement relations [25], which form a special case of alternating simulation relations:

Definition 1 (Feedback refinement)**.** *Given two transition systems $S_a = (X_a, U_a, \delta_a)$ and $S_b = (X_b, U_b, \delta_b)$, with $U_b \subseteq U_a$, a map $H : X_a \to X_b$ defines a feedback refinement relation from S_a to S_b if for all $(x_a, x_b) \in X_a \times X_b$ with $x_b = H(x_a)$:*

- *$U_b(x_b) \subseteq U_a(x_a)$;*
- *for all $u \in U_b(x_b)$, $H(\delta_a(x_a,u)) \subseteq \delta_b(x_b,u)$.*

We denote $S_a \preceq_{\mathcal{FR}} S_b$.

In the previous definition, S_a represents a complex concrete system while S_b is a simpler abstraction. From Definition 1, it follows that all abstract inputs u of S_b can also be used in S_a such that all concrete transitions in S_a are matched by an abstract transition in S_b. As a result, controllers synthesized using the abstraction S_b can be interfaced with the map H to obtain a controller for the concrete system S_a (see [25]). In particular, if $C_b : X_b \to 2^{U_b}$ is a safety controller for transition system S_b and safe set $\mathcal{X}_b \subseteq X_b$, then $C_a : X_a \to 2^{U_a}$, given by $C_a(x_a) = C_b(H(x_a))$ for all $x_a \in X_a$, is a safety controller for transition system S_a and safe set $\mathcal{X}_a = H^{-1}(\mathcal{X}_b) \subseteq X_a$.

III. Compositional abstraction

System (1) can be described as a transition system $S = (X, U, \delta)$ where, $X = \mathbb{R}^n$, $U = \mathcal{U}$ and $\delta = F$; let $\mathcal{X} \subseteq \mathbb{R}^n$ be a subset of states of interest. In this section, we present a compositional approach for computing symbolic abstractions of transition system S.

In order to allow for system decomposition, we will make the following assumption on the structure of the state and input sets $\mathcal{X}$ and $\mathcal{U}$:

Assumption 1. *The following equalities hold:*

$$\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_{\bar{n}}, \textit{ with } \mathcal{X}_i \subseteq \mathbb{R}^{n_i},\ i \in I = \{1, \dots, \bar{n}\};$$
$$\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_{\bar{p}}, \ \textit{ with } \mathcal{U}_j \subseteq \mathbb{R}^{p_j},\ j \in J = \{1, \dots, \bar{p}\}.$$

States $x \in \mathbb{R}^n$ and inputs $u \in \mathbb{R}^p$ can thus be seen as vectors of elementary components: $x = (x_1, \dots, x_{\bar{n}})$ with $x_i \in \mathbb{R}^{n_i}$ for $i \in I$, and $u = (u_1, \dots, u_{\bar{p}})$ with $u_j \in \mathbb{R}^{p_j}$ for $j \in J$.

For $i \in I$, let $\mathcal{P}_i$ be a finite partition of the set $\mathcal{X}_i$, then let $\mathcal{P}$ be the finite partition of the safe set $\mathcal{X}$ obtained from the partitions $\mathcal{P}_i$ as follows:

$$\mathcal{P} = \{s_1 \times \cdots \times s_{\bar{n}} | \ s_i \in \mathcal{P}_i,\ i \in I\}.$$

Similarly, for $j \in J$, let $\mathcal{V}_j$ be a finite subset of $\mathcal{U}_j$, then let $\mathcal{V}$ be the finite subset of $\mathcal{U}$ given by the Cartesian product of the sets $\mathcal{V}_j$:

$$\mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_{\bar{p}}.$$

A. System decomposition

Let $m \in \mathbb{N}$, with $1 \leq m \leq \min(\bar{n}, \bar{p})$, let $\Sigma = \{1, \dots, m\}$, the symbolic abstraction of S is obtained by composition of m symbolic subsystems S_σ, $\sigma \in \Sigma$.

In the following, we use two types of indices:

- Latin letters $i \in I$, $j \in J$, refer to x_i and u_j the components of the state and input x and u of system S.
- Greek letters $\sigma \in \Sigma$ refer to S_σ the σ-th symbolic subsystem, s_σ and u_σ denote the state and input of system S_σ respectively.

We will use $\pi_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$ and $\pi_j : \mathbb{R}^p \to \mathbb{R}^{p_j}$ to denote the projections over components x_i and u_j, with $i \in I$, $j \in J$, respectively. For $\mathcal{X}' \subseteq \mathbb{R}^n$ and $\mathcal{U}' \subseteq \mathbb{R}^p$, we denote $\mathcal{X}'_i = \pi_i(\mathcal{X}')$ and $\mathcal{U}'_j = \pi_j(\mathcal{U}')$. Similarly, for subset of indices $I' \subseteq I$, $J' \subseteq J$, $\pi_{I'} : \mathbb{R}^n \to \prod_{i \in I'} \mathbb{R}^{n_i}$ and $\pi_{J'} : \mathbb{R}^p \to \prod_{j \in J'} \mathbb{R}^{p_j}$ denote the projections over the set of components $\{x_i | \; i \in I'\}$ and $\{u_j | \; j \in J'\}$, respectively; we use the notation $x_{I'} = \pi_{I'}(x)$, $\mathcal{X}'_{I'} = \pi_{I'}(\mathcal{X}')$, $u_{J'} = \pi_{J'}(u)$ and $\mathcal{U}'_{J'} = \pi_{J'}(\mathcal{U}')$.

For $\sigma \in \Sigma$, subsystem S_σ can be described using the following sets of indices:

- $I^c_\sigma \subseteq I$, with $I^c_\sigma \neq \emptyset$, denotes the state components to be controlled in S_σ, $(I^c_1, \dots, I^c_m)$ is a partition of the state indices I;
- $I_\sigma \subseteq I$, with $I^c_\sigma \subseteq I_\sigma$, denotes the state components modeled in S_σ;
- $I^o_\sigma \subseteq I$, with $I^o_\sigma = I_\sigma \backslash I^c_\sigma$, denotes the state components that are modeled but not controlled in S_σ;
- $I^u_\sigma \subseteq I$, with $I^u_\sigma = I \backslash I_\sigma$, denotes the remaining state components that are unmodeled in S_σ;
- $J_\sigma \subseteq J$, with $J_\sigma \neq \emptyset$, denotes the control input components modeled in S_σ, $(J_1, \dots, J_m)$ is a partition of the control input indices J;
- $J^u_\sigma \subseteq J$ with $J^u_\sigma = J \backslash J_\sigma$, denotes the remaining control input components that are unmodeled in S_σ.

It is important to note that the subsystems may share common modeled state components (i.e. the sets of indices I_σ may overlap), though the sets of controlled state components I^c_σ and modeled control input components J_σ are necessarily disjoints. Intuitively, S_σ will be used to control state components I^c_σ using input components J_σ; other state components $I^o_\sigma \cup I^u_\sigma$ will be controlled in other subsystems using input components J^u_σ. Though state components I^o_σ will be controlled in other subsystems, they are modeled in S_σ and thus information on their dynamics is available for the control of S_σ.

Let us remark that the sets of indices I^o_σ and I^u_σ may possibly be empty if $I_\sigma = I^c_\sigma$ and $I_\sigma = I$, respectively. If $m = 1$, there is only one subsystem and we encompass the usual centralized abstraction approach (see e.g. [26], [31], [10], [25]).

Remark 1. *In theory, the choice of the sets of indices can be made arbitrarily. However, if the considered system has some structure, i.e. if it consists of interconnected components, a natural decomposition is to associate to each component $\mathcal{C}$ one subsystem S_σ where: the controlled states I^c_σ and the modeled control input J_σ are the states and control inputs of $\mathcal{C}$ and the modeled but uncontrolled states I^o_σ are the states of other components that have the strongest interactions with $\mathcal{C}$.*

B. Symbolic subsystems

Let $\sigma \in \Sigma$, the symbolic subsystem S_σ is an abstraction of S, which models only state and input components x_{I_σ} and u_{J_σ} respectively. Formally, subsystem S_σ is defined as a transition system $S_\sigma = (X_\sigma, U_\sigma, \delta_\sigma)$ where:

- the set of states X_σ is a finite partition of $\pi_{I_\sigma}(\mathbb{R}^n)$, given by $X_\sigma = X^0_\sigma \cup \{Out_\sigma\}$ where $Out_\sigma = \pi_{I_\sigma}(\mathbb{R}^n) \setminus \mathcal{X}_{I_\sigma}$ and
$$X^0_\sigma = \left\{ \prod_{i \in I_\sigma} s_i \middle| \; s_i \in \mathcal{P}_i, \; i \in I_\sigma \right\}$$
is a finite partition of $\mathcal{X}_{I_\sigma}$;
- the set of inputs U_σ is a finite subset of $\mathcal{U}_{J_\sigma}$ given by
$$U_\sigma = \prod_{j \in J_\sigma} \mathcal{V}_j.$$

To define the transition relation of S_σ, let us first define the following map: given $s_\sigma \in X^0_\sigma$ and $u_\sigma \in U_\sigma$, we define the set $\Phi_\sigma(s_\sigma, u_\sigma) \subseteq \mathbb{R}^n$ as follows:

$$\Phi_\sigma(s_\sigma, u_\sigma) = \overline{F}(\mathcal{X} \cap \pi^{-1}_{I_\sigma}(s_\sigma), \mathcal{U} \cap \pi^{-1}_{J_\sigma}(\{u_\sigma\})). \qquad (3)$$

The set $\Phi_\sigma(s_\sigma, u_\sigma)$ is therefore an over-approximation of successors of states $x \in \mathcal{X}$ with $\pi_{I_\sigma}(x) \in s_\sigma$, for control inputs $u \in \mathcal{U}$ with $\pi_{J_\sigma}(u) = u_\sigma$. Then, we define the transition relation of S_σ as follows:

- for all $s_\sigma \in X^0_\sigma, \; u_\sigma \in U_\sigma, \; s'_\sigma \in X^0_\sigma$,
$$s'_\sigma \in \delta_\sigma(s_\sigma, u_\sigma) \iff s'_\sigma \cap \pi_{I_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma)) \neq \emptyset; \quad (4)$$
- for all $s_\sigma \in X^0_\sigma, \; u_\sigma \in U_\sigma$,
$$Out_\sigma \in \delta_\sigma(s_\sigma, u_\sigma) \iff \left\{ \begin{array}{l} \pi_{I_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma)) \cap \mathcal{X}_{I_\sigma} = \emptyset \\ \text{or } \pi_{I^c_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma)) \nsubseteq \mathcal{X}_{I^c_\sigma}. \end{array} \right. \quad (5)$$

Remark 2. *The first condition in (5) holds if and only if there does not exist any transition defined by (4), because X^0_σ is a partition of $\mathcal{X}_{I_\sigma}$. As a consequence, it follows that for all $s_\sigma \in X^0_\sigma$, $u_\sigma \in U_\sigma$, $\delta_\sigma(s_\sigma, u_\sigma) \neq \emptyset$ and thus $U_\sigma(s_\sigma) = U_\sigma$.*

Remark 3. *According to (5), a transition to Out_σ exists if $\pi_{I_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma))$ is entirely outside $\mathcal{X}_{I_\sigma}$ (first condition and Figure 1.a); or if $\pi_{I^c_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma))$ is not contained in $\mathcal{X}_{I^c_\sigma}$ (second condition and Figure 1.b). It should be noted that in the case where the reachable set $\pi_{I^c_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma))$ is contained in $\mathcal{X}_{I^c_\sigma}$ but $\pi_{I^o_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma))$ is not contained in $\mathcal{X}_{I^o_\sigma}$ as in Figure 1.c, no transition is created towards Out_σ. Finally, if $I_\sigma = I^c_\sigma$, (5) becomes equivalent to*

$$Out_\sigma \in \delta_\sigma(s_\sigma, u_\sigma) \iff \pi_{I_\sigma}(\Phi_\sigma(s_\sigma, u_\sigma)) \nsubseteq \mathcal{X}_{I_\sigma},$$

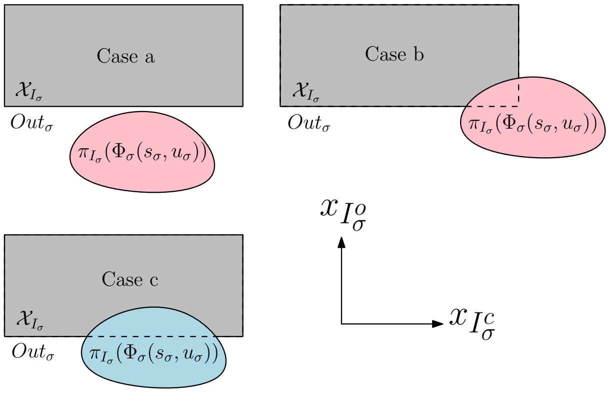

Fig. 1. Illustration of (5): a transition towards Out_σ is created in cases a and b, but not in case c.

which is the condition used in [19], for compositional abstractions where the set of modeled state components I_σ do not overlap (i.e. $I_\sigma = I^c_\sigma$, for all $\sigma \in \Sigma$).

C. Composition

In this section, we show how the previous subsystems S_σ, with $\sigma \in \Sigma$, can be composed in order to define a symbolic abstraction S_c of the original system S. The main result of the section is Theorem 3, which shows that there exists a feedback refinement relation from S to S_c.

The composition of the subsystems S_σ, $\sigma \in \Sigma$, is given by the transition system $S_c = (X_c, U_c, \delta_c)$ where:

- the set of states X_c is a finite partition of $\mathbb{R}^n$, given by $X_c = X^0_c \cup \{Out\}$ where $Out = \mathbb{R}^n \setminus \mathcal{X}$ and $X^0_c = \mathcal{P}$ is a finite partition of $\mathcal{X}$;
- the set of inputs $U_c = \mathcal{V}$ is a finite subset of $\mathcal{U}$.

Let us remark that by definition of X^0_c and X^0_σ, we have that for all $s \in X^0_c$, its projection $s_{I_\sigma} \in X^0_\sigma$. Similarly, for all $u \in U_c$, its projection $u_{J_\sigma} \in U_\sigma$. The transition relation of S_c can therefore be defined as follows:

- for all $s \in X^0_c,\ u \in U_c,\ s' \in X^0_c$,
$$s' \in \delta_c(s,u) \iff \forall \sigma \in \Sigma,\ s'_{I_\sigma} \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma}); \tag{6}$$
- for all $s \in X^0_c,\ u \in U_c$,
$$Out \in \delta_c(s,u) \iff \exists \sigma \in \Sigma,\ Out_\sigma \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma}). \tag{7}$$

Remark 4. *Because the sets of modeled state components I_σ are allowed to overlap, the transition relation of S_c cannot simply be obtained as the Cartesian product of the transition relations of the subsystems S_σ, as in [19]. Indeed, for $s \in X^0_c$, $u \in U_c$, it is possible that for all $\sigma \in \Sigma$, there exists $s'_\sigma \in X^0_\sigma$, such that $s'_\sigma \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$. However, a transition to X^0_c will exist in S_c if and only if there exists $s' \in X^0_c$ such that $s'_{I_\sigma} = s'_\sigma$, for all $\sigma \in \Sigma$.*

In view of the previous remark, it is legitimate to ask if the composition of the subsystems can lead to couples of states and inputs $(s,u) \in X^0_c \times U_c$ without a successor. The following proposition shows that this is not the case:

Proposition 2. *Under Assumption 1, for all $s \in X^0_c$ we have $U_c(s) = U_c$, i.e. $\delta_c(s,u) \neq \emptyset$, for all $u \in U_c$.*

Proof. Let $s \in X^0_c$ and $u \in U_c$. Then for all $\sigma \in \Sigma$, $s_{I_\sigma} \in X^0_\sigma$, $u_{J_\sigma} \in U_\sigma$ and by construction, $\delta_\sigma(s_{I_\sigma}, u_{J_\sigma}) \neq \emptyset$ (see Remark 2). If there exists a subsystem S_σ such that $Out_\sigma \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$, then by definition of S_c we have $Out \in \delta_c(s,u)$. Otherwise, we have that $Out_\sigma \notin \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$ for all $\sigma \in \Sigma$, which from the second condition of (5) implies that
$$\forall \sigma \in \Sigma,\ \pi_{I^c_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \subseteq \mathcal{X}_{I^c_\sigma}. \tag{8}$$
Remarking that $s \subseteq \mathcal{X} \cap \pi^{-1}_{I_\sigma}(s_{I_\sigma})$ and $\{u\} \subseteq \mathcal{U} \cap \pi^{-1}_{J_\sigma}(\{u_{J_\sigma}\})$, the following inclusion follows from (2) and (3):
$$\begin{aligned} F(s,\{u\}) &\subseteq F(\mathcal{X} \cap \pi^{-1}_{I_\sigma}(s_{I_\sigma}), \mathcal{U} \cap \pi^{-1}_{J_\sigma}(\{u_{J_\sigma}\})) \\ &\subseteq \Phi_\sigma(s_{I_\sigma}, u_{J_\sigma}). \end{aligned} \tag{9}$$
Therefore, from (8) and (9) it follows
$$\forall \sigma \in \Sigma,\ \pi_{I^c_\sigma}(F(s,\{u\})) \subseteq \mathcal{X}_{I^c_\sigma}.$$
This, together with Assumption 1 and the fact that $(I^c_1, \dots, I^c_m)$ is a partition of I, implies that $F(s,\{u\}) \subseteq \mathcal{X}$. Since X^0_c is a partition of $\mathcal{X}$, there exists $s' \in X^0_c$ such that $s' \cap F(s,\{u\}) \neq \emptyset$. Then, (9) gives $s' \cap \Phi_\sigma(s_{I_\sigma}, u_{J_\sigma}) \neq \emptyset$, for all $\sigma \in \Sigma$. Thus, for all $\sigma \in \Sigma$, $s'_{I_\sigma} \cap \pi_{I_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \neq \emptyset$. It follows from (4) that $s'_{I_\sigma} \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$ for all $\sigma \in \Sigma$, which gives, by (6), $s' \in \delta_c(s,u)$. □

We can now state the main result of the section:

Theorem 3. *Let the map $H : X \to X_c$ be given by $H(x) = s$ if and only if $x \in s$. Then, under Assumption 1, H defines a feedback refinement relation from S to S_c: $S \preceq_{\mathcal{FR}} S_c$.*

Proof. Let $s \in X^0_c$, $x \in s$, $u \in U_c(s) = U_c \subseteq U = U(x)$, $x' \in \delta(x,u) = F(x,u)$ and $s' = H(x')$. Since $x \in s$, we have $x' \in F(s,\{u\})$. Then, let us consider the two possible cases:

- $x' \in \mathcal{X}$ – We have by (9), $x' \in \Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})$, for all $\sigma \in \Sigma$. Since $x' \in \mathcal{X}$, then $s' \in X^0_c$, it follows from $x' \in s'$ that $s' \cap \Phi_\sigma(s_{I_\sigma}, u_{J_\sigma}) \neq \emptyset$, for all $\sigma \in \Sigma$. Then, for all $\sigma \in \Sigma$, $s'_{I_\sigma} \in X^0_\sigma$ and $s'_{I_\sigma} \cap \pi_{I_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \neq \emptyset$. From (4), $s'_{I_\sigma} \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$, for all $\sigma \in \Sigma$ and by (6) we have $s' \in \delta_c(s,u)$.
- $x' \notin \mathcal{X}$ – Then, $F(s,\{u\}) \not\subseteq \mathcal{X}$. Then, from Assumption 1 and the fact that $(I^c_1, \dots, I^c_m)$ is a partition of I, it follows that there exists $\sigma \in \Sigma$ such that $\pi_{I^c_\sigma}(F(s,\{u\})) \not\subseteq \mathcal{X}_{I^c_\sigma}$. From (9), we have $\pi_{I^c_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \not\subseteq \mathcal{X}_{I^c_\sigma}$. Then, from (5), $Out_\sigma \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$, and from (7), $Out \in \delta_c(s,u)$. Since $x' \notin \mathcal{X}$, $s' = Out$.

The case $s = Out$ trivially satisfies Definition 1 since $U_c(Out) = \emptyset$ by definition of S_c. □

Note that the composed abstraction S_c is only created in this section to prove the feedback refinement relationship but one should avoid computing it in practice since it would defeat the purpose of the compositional approach. We end the section by stating an instrumental result, which will be used in Section V when comparing abstractions obtained from different system decompositions.

Lemma 4. *Under Assumption 1, for all $s \in X^0_c$ and $u \in U_c$, $Out \in \delta_c(s,u)$ if and only if there exists $\sigma \in \Sigma$ such that $\pi_{I^c_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \not\subseteq \mathcal{X}_{I^c_\sigma}$.*

Proof. Sufficiency is straightforward from (5) and (7). As for necessity, if $Out \in \delta_c(s,u)$, then there exists a subsystem σ such that $Out_\sigma \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$. From (5), either $\pi_{I^c_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \not\subseteq \mathcal{X}_{I^c_\sigma}$ (in which case the property holds), or $\pi_{I^c_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \subseteq \mathcal{X}_{I^c_\sigma}$ and $\pi_{I_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \cap \mathcal{X}_{I_\sigma} = \emptyset$. Then, by Assumption 1 and since $I^o_\sigma = I_\sigma \setminus I^c_\sigma$, it follows that $\pi_{I^o_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \cap \mathcal{X}_{I^o_\sigma} = \emptyset$. Then, by (9), $\pi_{I^o_\sigma}(F(s,\{u\})) \cap \mathcal{X}_{I^o_\sigma} = \emptyset$. Thus, it follows that $\pi_{I^o_\sigma}(F(s,\{u\})) \not\subseteq \mathcal{X}_{I^o_\sigma}$. From Assumption 1, there exists $i \in I^o_\sigma$, such that $\pi_i(F(s,\{u\})) \not\subseteq \mathcal{X}_i$. Then, let $\sigma' \in \Sigma$ such that $i \in I^c_{\sigma'}$, then $\pi_{I^c_{\sigma'}}(F(s,\{u\})) \not\subseteq \mathcal{X}_{I^c_{\sigma'}}$. By (9), it follows that $\pi_{I^c_{\sigma'}}(\Phi_\sigma(s_{I_{\sigma'}}, u_{J_{\sigma'}})) \not\subseteq \mathcal{X}_{I^c_{\sigma'}}$ and the property holds. □

IV. Compositional safety synthesis

In this section, we consider the problem of synthesizing a safety controller for transition system S and safe set $\mathcal{X}$. Because of the feedback refinement relation from S to S_c, this can be done by solving the safety synthesis problem for transition system S_c and safe set X_c^0. We propose a compositional approach, which works on the symbolic subsystems S_σ and does not require computing the composed abstraction S_c.

For $\sigma \in \Sigma$, let $C_\sigma^* : X_\sigma \to 2^{U_\sigma}$ be the maximal safety controller for transition system S_σ and safe set X_σ^0. Since S_σ has only finitely many states and inputs, C_σ^* can be computed in finite time using a fixed point algorithm [26]. Now, let the controller $C_c : X_c \to 2^{U_c}$ be defined by $C_c(Out) = \emptyset$ and

$$\forall s \in X_c^0,\ C_c(s) = \{u \in U_c |\ u_{J_\sigma} \in C_\sigma^*(s_{I_\sigma}), \forall \sigma \in \Sigma\}. \quad (10)$$

Theorem 5. *Under Assumption 1, C_c is a safety controller for transition system S_c and safe set X_c^0.*

Proof. From Proposition 2 and since $C_c(Out) = \emptyset$, it is clear that for all $s \in X_c$, we have $C_c(s) \subseteq U_c(s)$. $C_c(Out) = \emptyset$ also gives $dom(C_c) \subseteq X_c^0$. Then, let $s \in dom(C_c) \subseteq X_c^0$, $u \in C_c(s)$ and $s' \in \delta_c(s,u)$. If $s' \notin X_c^0$, then $s' = Out$ and from (7), there exists $\sigma \in \Sigma$, such that $Out_\sigma \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$, which contradicts the fact that $u_{J_\sigma} \in C_\sigma^*(s_{I_\sigma})$ with C_σ^* safety controller for transition system S_σ and safe set X_σ^0. Hence, we necessarily have $s' \in X_c^0$, and from (6), it follows that $s'_{I_\sigma} \in \delta_\sigma(s_{I_\sigma}, u_{J_\sigma})$, for all $\sigma \in \Sigma$. Moreover, $u_{J_\sigma} \in C_\sigma^*(s_{I_\sigma})$ gives that $s'_{I_\sigma} \in dom(C_\sigma^*)$. Then for all $\sigma \in \Sigma$, let $u'_\sigma \in C_\sigma^*(s'_{I_\sigma})$. Since $(J_1, \dots, J_m)$ is a partition of J, there exists $u' \in U_c$ such that $u'_{J_\sigma} = u'_\sigma$ for all $\sigma \in \Sigma$. Then, by (10), $u' \in C_c(s')$ and thus $s' \in dom(C_c)$. It follows that C_c is a safety controller for transition system S_c and safe set X_c^0. □

Remark 5. *Since the sets of modeled state components I_σ may overlap, it is in principle possible that $dom(C_c) = \emptyset$ while $dom(C_\sigma^*) \neq \emptyset$, for all $\sigma \in \Sigma$. The reason is that an element of $dom(C_c)$ is obtained from states in $dom(C_\sigma^*)$, which coincide on their common modeled states, as shown in (10).*

A. Particular case: non-overlapping state sets

Though C_σ^* is a maximal safety controller for all $\sigma \in \Sigma$, the safety controller C_c is generally not maximal. Maximality can be obtained when the set of modeled states I_σ, $\sigma \in \Sigma$ do not overlap (or equivalently when for all $\sigma \in \Sigma$, $I_\sigma^c = I_\sigma$). In that case, the following result holds:

Proposition 6. *Under Assumption 1, let $I_\sigma^c = I_\sigma$, for all $\sigma \in \Sigma$. Then, C_c is the maximal safety controller for transition system S_c and safe set X_c^0.*

Proof. Let $C'_c : X_c \to 2^{U_c}$ be a safety controller for transition system S_c and safe set X_c^0. For $\sigma \in \Sigma$, let the controllers $C'_\sigma : X_\sigma \to 2^{U_\sigma}$ be defined by $C'_\sigma(Out_\sigma) = \emptyset$ and for all $s_\sigma \in X_\sigma^0$,

$$C'_\sigma(s_\sigma) = \{u_\sigma \in \pi_{J_\sigma}(C'_c(s)) |\ s \in X_c^0,\ s_{I_\sigma} = s_\sigma\}. \quad (11)$$

Let us show that C'_σ is a safety controller for system S_σ and safe set X_σ^0. Following Remark 2, and since $C'_\sigma(Out_\sigma) = \emptyset$, it is clear that for all $s_\sigma \in X_\sigma$, we have $C'_\sigma(s_\sigma) \subseteq U_\sigma(s_\sigma)$. $C'_\sigma(Out_\sigma) = \emptyset$ also gives $dom(C'_\sigma) \subseteq X_\sigma^0$. Then, let $s_\sigma \in dom(C'_\sigma)$, $u_\sigma \in C'_\sigma(s_\sigma)$ and $s'_\sigma \in \delta_\sigma(s_\sigma, u_\sigma)$, let us prove that $s'_\sigma \in dom(C'_\sigma)$. By (11), there exists $s \in dom(C'_c)$ and $u \in C'_c(s)$ such that $s_{I_\sigma} = s_\sigma$ and $u_{J_\sigma} = u_\sigma$. Since C'_c is a safety controller, $\delta_c(s,u) \subseteq dom(C'_c) \subseteq X_c^0$. Moreover, since the sets I_σ are not overlapping, it follows from (6) that there exists $s' \in \delta_c(s,u)$ such that $s'_{I_\sigma} = s'_\sigma$. Then, $s' \in dom(C'_c)$ and (11) give that $s'_\sigma \in dom(C'_\sigma)$. Hence C'_σ is a safety controller for system S_σ and safe set X_σ^0. Then, by maximality of C_σ^*, it follows that for all $s_\sigma \in X_\sigma^0$, $C'_\sigma(s_\sigma) \subseteq C_\sigma^*(s_\sigma)$. Finally, let $s \in dom(C'_c)$ and $u \in C'_c(s)$, then by (11), $u_{J_\sigma} \in C'_\sigma(s_{I_\sigma}) \subseteq C_\sigma^*(s_{I_\sigma})$ for all $\sigma \in \Sigma$. By (10), $u \in C_c(s)$, which shows the maximality of C_c. □

V. Comparisons

In this section, we provide theoretical comparisons between abstractions and controllers given by the previous approach using two different system decompositions.

In addition to the set of state and input indices defined in Section III-A, let us consider partitions $(\hat{I}_1^c, \dots, \hat{I}_{\hat{m}}^c)$ and $(\hat{J}_1, \dots, \hat{J}_{\hat{m}})$ of the state and input indices and subsets of state indices $(\hat{I}_1, \dots, \hat{I}_{\hat{m}})$ with $\hat{I}_{\hat{\sigma}}^c \subseteq \hat{I}_{\hat{\sigma}}$, for all $\hat{\sigma} \in \hat{\Sigma} = \{1, \dots, \hat{m}\}$. We define the same objects as before (i.e. subsystems, abstraction, controllers, etc.) for this system decomposition and denote them with hatted notations. We make the following assumption on the two system decompositions under consideration.

Assumption 2. *There exists a surjective map $\gamma : \hat{\Sigma} \to \Sigma$ such that, for all $\hat{\sigma} \in \hat{\Sigma}$ and $\sigma = \gamma(\hat{\sigma}) \in \Sigma$,*

$$\hat{I}_{\hat{\sigma}}^c \subseteq I_\sigma^c,\ \hat{I}_{\hat{\sigma}} \subseteq I_\sigma,\ \hat{J}_{\hat{\sigma}} \subseteq J_\sigma.$$

From the previous assumption, and since $(\hat{I}_1^c, \dots, \hat{I}_{\hat{m}}^c)$ and $(\hat{J}_1, \dots, \hat{J}_{\hat{m}})$ are partitions of the state and input indices, we have that

$$\forall \sigma \in \Sigma, \bigcup_{\hat{\sigma} \in \gamma^{-1}(\sigma)} \hat{I}_{\hat{\sigma}}^c = I_\sigma^c \text{ and } \bigcup_{\hat{\sigma} \in \gamma^{-1}(\sigma)} \hat{J}_{\hat{\sigma}} = J_\sigma. \quad (12)$$

In addition, we will make the following mild assumption on the over-approximations of the reachable sets:

Assumption 3. *For all $\mathcal{X}'' \subseteq \mathcal{X}' \subseteq \mathbb{R}^n$, $\mathcal{U}'' \subseteq \mathcal{U}' \subseteq \mathcal{U}$, the following inclusion holds*

$$\overline{F}(\mathcal{X}'', \mathcal{U}'') \subseteq \overline{F}(\mathcal{X}', \mathcal{U}').$$

This assumption can be shown to be satisfied by most existing techniques for over-approximating the reachable set, and in particular by those mentioned in Section II-A. In addition, under Assumptions 2 and 3, it follows from (3), that for all $\hat{\sigma} \in \hat{\Sigma}$ and $\sigma = \gamma(\hat{\sigma}) \in \Sigma$,

$$\forall s \in \mathcal{P},\ u \in \mathcal{V},\ \Phi_\sigma(s_{I_\sigma}, u_{J_\sigma}) \subseteq \hat{\Phi}_{\hat{\sigma}}(s_{\hat{I}_{\hat{\sigma}}}, u_{\hat{J}_{\hat{\sigma}}}). \quad (13)$$

A. Abstractions

We start by comparing the compositional abstractions S_c and $\hat{S}_c$ resulting from the two different decompositions:

Theorem 7. *Under Assumptions 1, 2 and 3, the identity map is a feedback refinement relation from S_c to $\hat{S}_c$: $S_c \preceq_{\mathcal{FR}} \hat{S}_c$.*

Proof. Let us first remark that $X_c^0 = \hat{X}_c^0$, $X_c = \hat{X}_c$ and $U_c = \hat{U}_c$. Then, from Proposition 2, for all $s \in X_c^0 = \hat{X}_c^0$, $U_c(s) = U_c = \hat{U}_c = \hat{U}_c(s)$. Since $\hat{U}_c(Out) = \emptyset$, Definition 1 holds if $\delta_c(s,u) \subseteq \hat{\delta}_c(s,u)$, for all $s \in X_c^0$, $u \in U_c$.

Hence, let $s \in X_c^0$, $u \in U_c$ and $s' \in \delta_c(s,u)$, then let us consider the two possible cases:

- $s' \in X_c^0$ – We have by (6) and (4) that
$$\forall \sigma \in \Sigma,\ s'_{I_\sigma} \cap \pi_{I_\sigma}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \neq \emptyset.$$
Then, from (13), follows that
$$\forall \hat{\sigma} \in \hat{\Sigma} \text{ and } \sigma = \gamma(\hat{\sigma}),\ s'_{I_\sigma} \cap \pi_{I_\sigma}(\hat{\Phi}_{\hat{\sigma}}(s_{\hat{I}_{\hat{\sigma}}}, u_{\hat{J}_{\hat{\sigma}}})) \neq \emptyset.$$
By Assumption 2, $\hat{I}_{\hat{\sigma}} \subseteq I_\sigma$, for all $\hat{\sigma} \in \hat{\Sigma}$ and $\sigma = \gamma(\hat{\sigma})$. Thus it follows that
$$\forall \hat{\sigma} \in \hat{\Sigma},\ s'_{\hat{I}_{\hat{\sigma}}} \cap \pi_{\hat{I}_{\hat{\sigma}}}(\hat{\Phi}_{\hat{\sigma}}(s_{\hat{I}_{\hat{\sigma}}}, u_{\hat{J}_{\hat{\sigma}}})) \neq \emptyset.$$
Then, from (4) and (6), we have $s' \in \hat{\delta}_c(s,u)$.
- $s' = Out$ – From Lemma 4, we know that there exists $\sigma \in \Sigma$, such that $\pi_{I_\sigma^c}(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \not\subseteq \mathcal{X}_{I_\sigma^c}$. Then, from Assumption 1, there exists $i \in I_\sigma^c$ such that $\pi_i(\Phi_\sigma(s_{I_\sigma}, u_{J_\sigma})) \not\subseteq \mathcal{X}_i$. From (12), there exists $\hat{\sigma} \in \hat{\Sigma}$, such that $\sigma = \gamma(\hat{\sigma})$ and $i \in \hat{I}_{\hat{\sigma}}^c$. From (13), it follows that $\pi_i(\hat{\Phi}_{\hat{\sigma}}(s_{\hat{I}_{\hat{\sigma}}}, u_{\hat{J}_{\hat{\sigma}}})) \not\subseteq \mathcal{X}_i$ and $\pi_{\hat{I}_{\hat{\sigma}}^c}(\hat{\Phi}_{\hat{\sigma}}(s_{\hat{I}_{\hat{\sigma}}}, u_{\hat{J}_{\hat{\sigma}}})) \not\subseteq \mathcal{X}_{\hat{I}_{\hat{\sigma}}^c}$. Then, from (5) and (7), we have $Out \in \hat{\delta}_c(s,u)$.

□

Note that the conditions in the previous Theorem are only sufficient conditions, since depending on the dynamics of the system, a feedback refinement relation could also exist between two unrelated decompositions (in terms of index set inclusion).

Remark 6. *Theorem 7 gives an indication on how one should modify the sets of indices to reduce the conservatism of the compositional symbolic abstraction. Firstly, one can keep the same number of subsystems and the same controlled states I_σ^c and modeled control input J_σ, while considering additional modeled but uncontrolled states in I_σ^o. Secondly, one can merge two or more subsystems by merging their controlled states, modeled control inputs and modeled but uncontrolled states.*

B. Controllers

We now compare the controllers obtained by the approach described in Section IV. The comparison of controllers is more delicate than the comparison of abstractions and we shall need the additional assumption that the sets of indices I_σ do not overlap (note that the sets $\hat{I}_{\hat{\sigma}}$ may still overlap).

Corollary 8. *Under Assumptions 1, 2 and 3, let $I_\sigma^c = I_\sigma$, for all $\sigma \in \Sigma$. Then, for all $s \in X_c$, $\hat{C}_c(s) \subseteq C_c(s)$.*

Proof. From Theorem 5, $\hat{C}_c$ is a safety controller for system $\hat{S}_c$ and safe set $\hat{X}_c$. From Theorem 7, it follows that $\hat{C}_c$ is also a safety controller for system S_c and safe set X_c. Then, by Proposition 6, the maximality of C_c gives us for all $s \in X_c$, $\hat{C}_c(s) \subseteq C_c(s)$. □

Let us remark that the assumption that the sets of indices I_σ do not overlap is instrumental in the proof since it uses Proposition 6. The question whether similar results hold in the absence of such assumption is an open question, which is left as future research.

C. Complexity

In this section, we discuss the computational complexity of the approach and show the advantage of using a compositional approach rather than a centralized one. Let $|.|$ denote the cardinality of a set.

The computation of symbolic subsystem S_σ requires a number of reachable set approximations equal to $\prod_{i \in I_\sigma} |\mathcal{P}_i| \times \prod_{j \in J_\sigma} |\mathcal{V}_j|$, each creating up to $(1 + \prod_{i \in I_\sigma} |\mathcal{P}_i|)$ successors. This results in an overall time and space complexity $\mathcal{C}_1$ of computing all symbolic subsystems S_σ, $\sigma \in \Sigma$:
$$\mathcal{C}_1 = \mathcal{O}\Big(\sum_{\sigma \in \Sigma} \big(\prod_{i \in I_\sigma} |\mathcal{P}_i|^2 \times \prod_{j \in J_\sigma} |\mathcal{V}_j| \big) \Big).$$

The computation of the safety controller C_σ by a fixed point algorithm requires a number of iteration which is bounded by the number of states in the safe set X_σ^0: $\prod_{i \in I_\sigma} |\mathcal{P}_i|$. The complexity order of computing an iteration can be bounded by the number of transitions in S_σ. This results in an overall time and space complexity $\mathcal{C}_2$ of computing all safety controllers C_σ, $\sigma \in \Sigma$:
$$\mathcal{C}_2 = \mathcal{O}\Big(\sum_{\sigma \in \Sigma} \big(\prod_{i \in I_\sigma} |\mathcal{P}_i|^3 \times \prod_{j \in J_\sigma} |\mathcal{V}_j| \big) \Big).$$

To illustrate the advantage of using a compositional approach, let us consider two extremal cases in the particular case where the number of state and input component are equal $I = J$. The centralized case corresponds to $\Sigma = \{1\}$, with $I_1 = J_1 = I$. In that case the complexity of the overall approach is of order $\mathcal{O}\Big(\prod_{i \in I} |\mathcal{P}_i|^3 \times |\mathcal{V}_i| \Big)$. The fully decentralized case corresponds to $\Sigma = I = J$, with $I_\sigma = J_\sigma = \{\sigma\}$ for all $\sigma \in \Sigma$. In that case the complexity of the overall approach is of order $\mathcal{O}\Big(\sum_{i \in I} |\mathcal{P}_i|^3 \times |\mathcal{V}_i| \Big)$. Hence, one can see that while the complexity of the centralized approach is exponential in the number of state and input components $|I|$, it becomes linear with the fully decentralized approach. Intermediate decompositions enable to balance the computational complexity and the conservativeness of the approach, in view of the discussions in Sections V-A and V-B.

VI. Numerical illustration

In this section, we illustrate the results of this paper on the temperature regulation in a circular building of $n \geq 3$ rooms, each equipped with a heater. For each room $i \in \{1, \dots, n\}$, the variations of the temperature T_i are described by the discrete-time model adapted from [22]:
$$T_i^+ = T_i + \alpha(T_{i+1} + T_{i-1} - 2T_i) + \beta(T_e - T_i) + \gamma(T_h - T_i)u_i,$$
where T_{i+1} and T_{i-1} are the temperature of the neighbor rooms (with $T_0 = T_n$ and $T_{n+1} = T_1$), $T_e = -1\,°C$ is the outside temperature, $T_h = 50\,°C$ is the heater temperature, $u_i \in [0, 0.6]$ is the control input for room i and the

λ_T	λ_u	$\lvert\mathcal{P}\rvert = \lambda_T^4$	$\lvert dom(C_c^1)\rvert$	$\lvert dom(C_c^2)\rvert$	$\lvert dom(C_c^3)\rvert$
5	3	625	525	0	0
5	4	625	525	0	0
10	3	10000	8900	8710	0
10	4	10000	8900	8710	0
20	3	160000	145180	143480	0
20	4	160000	145180	143480	0

TABLE I
NUMBER OF ELEMENTS IN THE DOMAINS OF THE SAFETY CONTROLLERS FOR THE SAFE SET $\mathcal{X} = [17, 22] \times [19, 22] \times [20, 23] \times [20, 22]$.

λ_T	λ_u	S_c^1	S_c^2	S_c^3
5	3	1.80	0.17	0.07
5	4	5.49	0.20	0.07
10	3	64	0.46	0.06
10	4	210	0.56	0.06
20	3	6044	2.87	0.09
20	4	18339	3.84	0.44

TABLE II
COMPUTATION TIMES (IN SECONDS) FOR THE SAFE SET $\mathcal{X} = [17, 22] \times [19, 22] \times [20, 23] \times [20, 22]$.

conduction factors are given by $\alpha = 0.45$, $\beta = 0.045$ and $\gamma = 0.09$. This model can be proved to be monotone as defined in [3], which allows us to use efficient algorithms for over-approximating the reachable sets [19], [10]. Moreover, the over-approximation scheme satisfies Assumption 3.

The safe set $\mathcal{X}$ is given by a n-dimensional interval (specified later) which is uniformly partitioned into λ_T intervals per component (for a total of λ_T^n symbols in $\mathcal{P}$) and the control set $\mathcal{U} = [0, 0.6]^n$ is uniformly discretized into λ_u values per component (for a total of λ_u^n values in $\mathcal{V}$). We consider 3 possible system decompositions, which provides us with 3 different abstractions:

- S_c^1, the centralized case (i.e. $m = 1$), with a single subsystem containing all states and controls, with $I_1^1 = I_1^{c1} = J_1^1 = \{1, \dots, n\}$;
- S_c^2, a general case from Section III with $m = n$ subsystems, $I_\sigma^{c2} = J_\sigma^2 = \{\sigma\}$ and $I_\sigma^2 = \{\sigma - 1, \sigma, \sigma + 1\}$ for all $\sigma \in \Sigma = \{1, \dots, m\}$;
- S_c^3, a case with $m = n$ subsystems and non-overlapping state sets as in Section IV-A, with $I_\sigma^3 = I_\sigma^{c3} = J_\sigma^3 = \{\sigma\}$ for all $\sigma \in \Sigma$.

Both S_c^2 and S_c^3 have one subsystem per room, but subsystems of S_c^3 only focus on the state and control of the considered room, while subsystems of S_c^2 also model (but do not control) the temperatures of both neighbor rooms.

Since Assumption 2 holds for both pairs (S_c^1, S_c^2) and (S_c^2, S_c^3), Theorem 7 immediately gives the feedback refinements $S_c^1 \preceq_{\mathcal{FR}} S_c^2 \preceq_{\mathcal{FR}} S_c^3$. Corollary 8 also holds for the pair (S_c^1, S_c^2) since $I_1^{c1} = I_1^1$, but it is not guaranteed to hold for the pair (S_c^2, S_c^3) since $I_\sigma^{c2} \neq I_\sigma^2$. In the following, we report numerical results in two different conditions. The numerical implementation has been done using Matlab on a laptop with a 2.6 GHz CPU and 8 GB of RAM.

Case 1: $n = 4$, $\mathcal{X} = [17, 22] \times [19, 22] \times [20, 23] \times [20, 22]$.

The abstractions and syntheses are generated in the 6 cases corresponding to state partitions with $\lambda_T \in \{5, 10, 20\}$ and input discretizations with $\lambda_u \in \{3, 4\}$. Table I reports the cardinalities $|\mathcal{P}| = \lambda_T^4$ of the state partition $\mathcal{P}$, and $|dom(C_c)|$ of the domain of the safety controllers for each abstraction S_c^1, S_c^2 and S_c^3. Table II reports the computation times (in seconds) required to create the abstractions and synthesize safety controllers on all subsystems of S_c^1, S_c^2 and S_c^3.

We check numerically that Theorem 7 and Corollary 8 hold. In particular, in these conditions, the safety inclusion $C_c^3(s) \subseteq C_c^2(s)$ for all $s \in \mathcal{P}$ trivially holds due to $dom(C_c^3) = \emptyset$, although Corollary 8 could not provide theoretical guarantees in this case.

Two main conclusions on the proposed compositional approach can be obtained from Tables I and II. Firstly, while the compositional case without state overlap (as in S_c^3, Section IV-A and [19]) fails to synthesize safety controllers, the general case allowing state overlaps (as in S_c^2 and Section III) provides significantly better safety results for a relatively small addition to the computation time. Secondly, the compositional approach with state overlaps S_c^2 requires a negligible computation time compared to the large computational cost of the centralized approach S_c^1 (e.g. in the last row of Table II, we need less than 4 *seconds* for S_c^2 and more than 5 *hours* for S_c^1), while still obtaining similar safety results as long as the state partition $\mathcal{P}$ is not too coarse.

In addition to having more information in each subsystem of S_c^2 compared to those in the non-overlapping case S_c^3, the better safety results in S_c^2 can also be explained by the shapes that can be taken by the domain of the safety controllers with each approach. On the one hand, the safety domain $dom(C_c^3)$ in the non-overlapping case S_c^3 can only take the form of a hyper-rectangle in $\mathbb{R}^4$ since it is obtained by the Cartesian product of the one-dimensional safety domains $dom(C_\sigma^3)$ of its subsystems. On the other hand, the general case with state overlaps S_c^2 is more permissive since its subsystems S_σ^2 have a three-dimensional state space, thus allowing more complicated shapes of their safety domains $dom(C_\sigma^2)$ as displayed in Figure 2 for subsystem $\sigma = 4$. S_c^2 thus has more chances finding a safety domain compatible with the considered system dynamics and control objective.

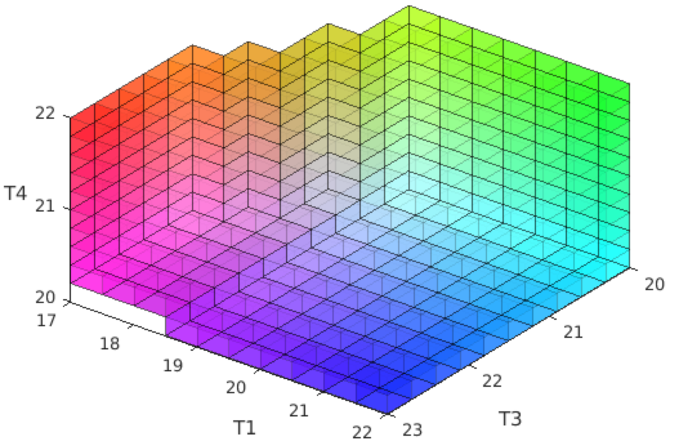

Fig. 2. Visualization of the domain $dom(C_\sigma^2)$ of the safety controller for subsystem $\sigma = 4$ of S_c^2. Each axis is associated with one component of the RGB color model to facilitate the visualization of depth.

Case 2: $n = 20$, $\mathcal{X} = [19, 21]^{20}$, $\lambda_T = 10$, $\lambda_u = 5$.

A second example is proposed to demonstrate the scalability of the compositional approach in a 20-room building. Note that the safe set $\mathcal{X} = [19, 21]^{20}$ is only chosen homogeneous in all rooms for convenience of notation, and the proposed approach is still applicable for other safe sets. Since this case is clearly out of reach from the centralized approach of S_c^1, we focus on the compositional abstractions S_c^2 and S_c^3 with and without state overlaps, respectively.

For the non-overlapping case of S_c^3, the total computation time is 0.12 second and the resulting safety controller is empty ($dom(C_c^3) = \emptyset$). For the case with state overlaps of S_c^2, the total computation time is 3.04 seconds and the resulting safety controller covers the whole safe set $\mathcal{X}$ ($dom(C_c^2) = \mathcal{P}$). Therefore, in addition to the scalability of both these compositional approaches, this simulation also confirms the conclusions of the previous example that the method with state overlaps provides significantly better safety results at a reduced computational cost. We also obtain similarly low computation times while not having to rely on the homogeneity of the specifications as it is the case in [22].

VII. Conclusion

In this paper, we presented a new compositional approach for symbolic controller synthesis. The dynamics are decomposed into subsystems that give a partial description of the global model. It is remarkable that the sets of states of subsystems can overlap. Symbolic abstractions can be computed for each subsystem, and a local safety controller can be synthesized such that the composition of the obtained controllers is proved to realize the global safety specification. Numerical experiments demonstrate the significant complexity reduction compared to centralized approaches and the advantages obtained from the introduction of state overlaps in the subsystems.

Future work will focus on extending the approach to other types of specifications such as reachability or more general properties specified by automata or temporal logic formula.

References

[1] M. Althoff and B. H. Krogh. Reachability analysis of nonlinear differential-algebraic systems. *IEEE Transactions on Automatic Control*, 59(2):371–383, 2014.

[2] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[3] D. Angeli and E. D. Sontag. Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10):1684–1698, 2003.

[4] C. Baier, J.-P. Katoen, and K. G. Larsen. *Principles of model checking*. MIT press, 2008.

[5] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):61–70, 2007.

[6] M. A. Ben Sassi, R. Testylier, T. Dang, and A. Girard. Reachability analysis of polynomial systems using linear programming relaxations. In *Automated Technology for Verification and Analysis*, pages 137–151. 2012.

[7] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar. Synthesis of reactive (1) designs. *Journal of Computer and System Sciences*, 78(3):911–938, 2012.

[8] D. Boskos and D. V. Dimarogonas. Decentralized abstractions for feedback interconnected multi-agent systems. In *IEEE Conference on Decision and Control*, pages 282–287, 2015.

[9] C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[10] S. Coogan and M. Arcak. Efficient finite abstraction of mixed monotone systems. In *Hybrid Systems: Computation and Control*, pages 58–67. 2015.

[11] E. Dallal and P. Tabuada. On compositional symbolic controller synthesis inspired by small-gain theorems. In *IEEE Conference on Decision and Control*, pages 6133–6138, 2015.

[12] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, pages 291–305. 2005.

[13] A. Girard, G. Gössler, and S. Mouelhi. Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models. *IEEE Transactions on Automatic Control*, 61(6):1537–1549, 2016.

[14] T. A. Henzinger, S. Qadeer, and S. K. Rajamani. You assume, we guarantee: Methodology and case studies. In *Computer aided verification*, pages 440–451, 1998.

[15] E. S. Kim, M. Arcak, and S. A. Seshia. Compositional controller synthesis for vehicular traffic networks. In *IEEE Conference on Decision and Control*, pages 6165–6171, 2015.

[16] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 52(1):26–38, 2007.

[17] E. Le Corronc, A. Girard, and G. Goessler. Mode sequences as symbolic states in abstractions of incrementally stable switched systems. In *IEEE Conference on Decision and Control*, pages 3225–3230, 2013.

[18] C. Le Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.

[19] P.-J. Meyer, A. Girard, and E. Witrant. Safety control with performance guarantees of cooperative systems using compositional abstractions. In *IFAC Conference on Analysis and Design of Hyrbid Systems*, pages 317–322, 2015.

[20] G. Pola, A. Borri, and M. D. Di Benedetto. Integrated design of symbolic controllers for nonlinear systems. *IEEE Transactions on Automatic Control*, 57(2):534–539, 2012.

[21] G. Pola, A. Girard, and P. Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

[22] G. Pola, P. Pepe, and M. D. Di Benedetto. Decentralized supervisory control of networks of nonlinear control systems. *arXiv preprint arXiv:1606.04647*, 2016.

[23] G. Pola, P. Pepe, and M. D. Di Benedetto. Symbolic models for networks of control systems. *IEEE Transactions on Automatic Control*, 61(11):3663–3668, 2016.

[24] G. Reissig. Abstraction based solution of complex attainability problems for decomposable continuous plants. In *IEEE Conference on Decision and Control*, pages 5911–5917, 2010.

[25] G. Reissig, A. Weber, and M. Rungger. Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Transactions on Automatic Control*, 62(4):1781–1796, 2016.

[26] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer, 2009.

[27] P. Tabuada and G. J. Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, 2006.

[28] Y. Tazaki and J.-i. Imura. Bisimilar finite abstractions of interconnected systems. In *Hybrid Systems: Computation and Control*, pages 514–527. 2008.

[29] M. Zamani, A. Abate, and A. Girard. Symbolic models for stochastic switched systems: A discretization and a discretization-free approach. *Automatica*, 55:183–196, 2015.

[30] M. Zamani, P. M. Esfahani, R. Majumdar, A. Abate, and J. Lygeros. Symbolic control of stochastic systems via approximately bisimilar finite abstractions. *IEEE Transactions on Automatic Control*, 59(12):3135–3150, 2014.

[31] M. Zamani, G. Pola, M. Mazo, and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.