Automatica 190 (2026) 113040

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Characterizing simulation relations through control architectures in abstraction-based control ☆

Julien Calbert [a,*], Antoine Girard [b], Raphaël M. Jungers [a]

[a] ICTEAM Institute, Department of Mathematical Engineering, UCLouvain, B-1348, Louvain-la-Neuve, Belgium
[b] Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France

ARTICLE INFO

Article history:
Received 4 October 2024
Received in revised form 24 December 2025
Accepted 19 March 2026

Keywords:
Abstraction
Symbolic model
Simulation relations

ABSTRACT

Abstraction-based control design is a promising approach for ensuring safety-critical control of complex cyber–physical systems. A key aspect of this methodology is the relation between the original and abstract systems, which ensures that the abstract controller can be transformed into a valid controller for the original system through a concretization procedure. In this paper, we provide a comprehensive and systematic framework that characterizes various simulation relations, through their associated concretization procedures. We introduce the concept of *interfaced system*, which universally enables a feedback refinement relation with the abstract system. This interfaced system encapsulates the specific characteristics of each simulation relation within an *interface*, enabling a plug-and-play control architecture. Our results demonstrate that the existence of a particular simulation relation between the concrete and abstract systems is equivalent to the implementability of a specific control architecture, which depends on the considered simulation relation. This allows us to introduce new types of relations, and to establish the advantages and drawbacks of different relations, which we exhibit through detailed examples.

1. Introduction

Abstraction-based control design is a thriving and promising proposal for safety-critical control of complex cyber–physical systems (Alur, 2015; Kim & Kumar, 2012). It stems from the idea of synthesizing a correct-by-construction controller through a systematic three-step procedure illustrated in Fig. 1. First, both the original system $\mathcal{S}_1$ and the specifications Σ_1 are transposed into an *abstract* domain, resulting in an abstract system $\mathcal{S}_2$ and corresponding abstract specifications Σ_2. We refer to the original system as the *concrete* system as opposed to the abstract system. Next, an abstract controller $\mathcal{C}_2$ is synthesized to solve this abstract control problem $(\mathcal{S}_2, \Sigma_2)$. Finally, in the third step, called *concretization* as opposed to abstraction, a controller $\mathcal{C}_1$ that enforces Σ_1 for $\mathcal{S}_1$ is derived from the abstract controller.

The effectiveness of this approach stems from replacing the concrete system, often characterized by an infinite number of states, with a finite state system. This enables the use of powerful control tools from abstract control synthesis (Belta et al., 2017; Kupferman & Vardi, 2001), such as those derived from graph theory, including methods like Dijkstra or the A-star algorithm.

The correctness of this approach is ensured by relating the concrete system with its abstraction by a simulation relation. The notion of *alternating simulation relation* (ASR) (Tabuada, 2009, Definition 4.19) is crucial; indeed, it is proved in the seminal work of Alur et al. (1998) that ASR is a sufficient condition for a system $\mathcal{S}_2$ to correctly represent a system $\mathcal{S}_1$, such that any controller $\mathcal{C}_2$ for $\mathcal{S}_2$ can be concretized into a valid controller $\mathcal{C}_1$ for $\mathcal{S}_1$.

However, it soon appeared that ASR has weaknesses: while it formally characterizes abstractions allowing for abstraction-based controller design, the exact way to reconstruct the concrete controller may be computationally expensive.

An important landmark in the abstraction-based control literature is Reissig et al. (2016). There, the authors introduce the stronger *feedback refinement relation* (FRR) (Reissig et al., 2016, Def. V.2), designed to alleviate these limitations. They show that when FRR holds between the abstract and concrete systems, one can simply concretize the controller by seamlessly plugging the abstract controller with a quantizer, see Fig. 2 (left). While this leads to a straightforward concretization procedure, where the concrete controller merely reproduces the control input of the abstract controller, it also limits the class of implementable controllers to piecewise constant ones (see discussion in Calbert, Mattenet, Girard, and Jungers (2024, Section 4.4)).

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Jun Liu under the direction of Editor Maria Elena Valcher.

* Corresponding author.
E-mail addresses: julien.calbert@uclouvain.be (J. Calbert), antoine.girard@centralesupelec.fr (A. Girard), raphael.jungers@uclouvain.be (R.M. Jungers).

https://doi.org/10.1016/j.automatica.2026.113040

Recently, several other simulation relations that refine the ASR have been proposed in the literature (Borri et al., 2018; Calbert, Mattenet, Girard, & Jungers, 2024; Egidio et al., 2022; Liu, 2017), each with its own benefits. While these notions are well understood individually, their relationships and relative strengths are usually presented separately. As a result, the landscape of simulation relations for symbolic control has become increasingly fragmented, making it harder to compare or systematically select among them.

In this paper, we unify these existing relations by interpreting each of them through the specific *control architecture* it induces. Motivated by this observation, we take a step back from the existing definitions of simulation relations and ask:

What is the connection between a particular simulation relation and the corresponding control architecture?

We propose a unified framework that captures the different types of simulation relations found in the literature and allows for their comparison in terms of the control architectures they induce. Our construction builds on the concept of an *interface*, in the spirit of Girard and Pappas (2009), which stores the output of the abstract controller $\mathcal{C}_2$, combines it with the observed state of the concrete system $\mathcal{S}_1$, and produces a valid control input. This architecture is illustrated in Fig. 2 (right). Consequently, the resulting concretization procedure exhibits a *plug-and-play* structure, enabling the abstract controller to be connected to the original system via an interface, irrespective of the particular specification we seek to enforce on the original system.

Our main theoretical contribution is a theorem (Theorem 4, Corollary 1) that explicitly links simulation relations to the control architecture that the concrete system inherits from the abstract one, as a function of the simulation relation that links them. Thus, the simulation relation becomes one of the elements in the control engineer's toolbox, with the choice of the precise relation to be used depending on the control architecture that (s)he wants to or is able to implement in practice.

In contrast to previous works, which typically prove only that a relation implies a correct interface, we also establish the converse: if the interface works for *any* abstract controller, then the corresponding simulation relation must hold. To our knowledge, this two-way characterization was previously established only for the FRR (Reissig et al., 2016, Theorems V.4, V.5). We leverage this result by proving that the *interfaced system*—that is, the system composed of the concrete system and the interface (see Fig. 2, right)—is always in feedback refinement relation with the abstract system.

Finally, as a by-product, we introduce new simulation relations that we believe are useful in practice, and we provide their explicit constructions on a common example, highlighting their respective advantages and drawbacks in terms of concretization procedures.

Notation. The sets $\mathbb{R}, \mathbb{Z}, \mathbb{N}$ denote respectively the sets of real numbers, integers and natural numbers. For example, we use $[a, b] \subseteq \mathbb{R}$ to denote a closed continuous interval and $[a; b] = [a, b] \cap \mathbb{Z}$ for discrete intervals. The symbol $\varnothing$ denotes the empty set. Given two sets A, B, we define a *single-valued map* as $f : A \to B$, while a *set-valued map* is defined as $f : A \to 2^B$, where 2^B is the power set of B, i.e., the set of all subsets of B. We denote the identity map $A \to A : a \to a$ by I_A. The image of a subset $\Omega \subseteq A$ under $f : A \to 2^B$ is denoted $f(\Omega) = \bigcup_{a\in\Omega} f(a)$. The set of maps $A \to B$ is denoted B^A, and the set of all signals that take their values in B and are defined on intervals of the form $[0; N[= [0; N-1]$ is denoted B^∞, $B^\infty = \bigcup_{N\in\mathbb{N}\cup\{\infty\}} B^{[0;N[}$. Given a set-valued map $f : X \to 2^Y$ and $\boldsymbol{x} \in X^{[0;N[}$, we denote by $f(\boldsymbol{x}) = \{\boldsymbol{y} \in Y^{[0;N[} \mid \forall k \in [0; N[\ : \ y(k) \in f(x(k))\}$. Let

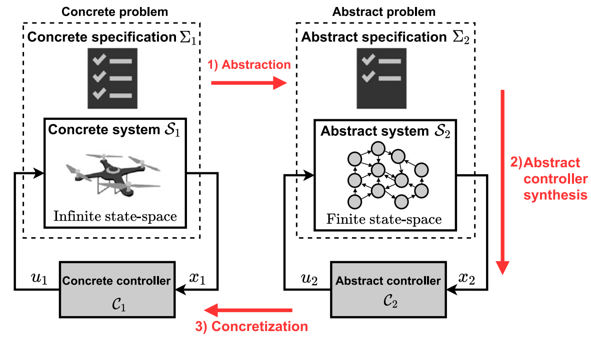


Fig. 1. The three steps of abstraction-based control. Our work focuses on the connection between the type of relation between $\mathcal{S}_1$ and $\mathcal{S}_2$ (step 1) and the concretization procedure (step 3). We show that one can tailor the abstraction relation depending on the required properties of the concretization step and of the finally obtained concrete controller, independently of the abstract controller synthesis step (step 2).

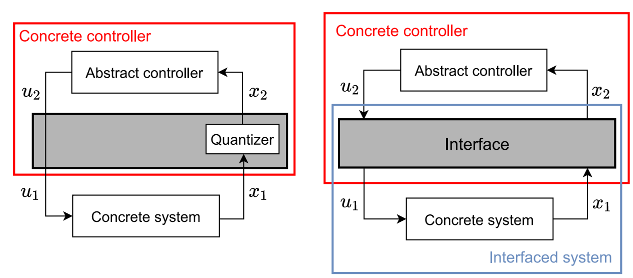


Fig. 2. Closed-loop system resulting from the abstraction and concretization approach. Left: Based on the feedback refinement relation. Right: Based on the generic hierarchical control architecture used in this paper.

$X \subseteq (V \times W)^\infty$, we define the projection onto the first component of the sequences as

$$\begin{aligned}\pi_V(X) = \{\boldsymbol{v} \in V^{[0;N[} \mid \exists \boldsymbol{w} \in W^{[0;N[}, \text{ such that} \\ ((v(0), w(0)), \ldots, (v(N-1), w(N-1))) \in X \\ \text{for } N \in \mathbb{N} \cup \{\infty\}\}.\end{aligned} \tag{1}$$

We define $\pi_W(X)$ similarly.

2. Control framework

In this section, we introduce the control framework used in this paper, as proposed in Reissig et al. (2016, Definition III.1), which provides a unified formalism to describe systems, controllers, quantizers, and interconnected systems.

Definition 1. A system is a sextuple

$$\mathcal{S} = (\mathcal{X}, \mathcal{U}, \mathcal{V}, \mathcal{Y}, F, H), \tag{2}$$

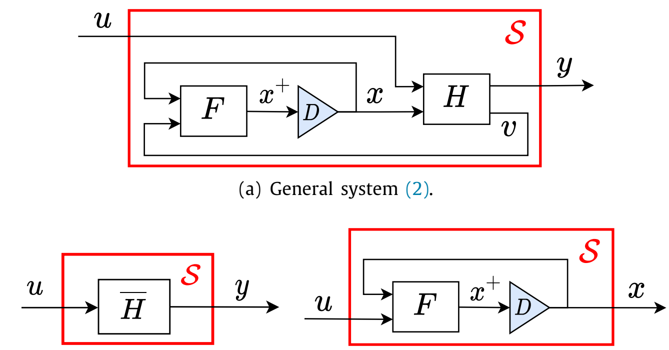
where $\mathcal{X}, \mathcal{U}, \mathcal{V}, \mathcal{Y}$ are respectively the sets of *states*, *inputs*, *internal variables* and *outputs*, and $F : \mathcal{X} \times \mathcal{V} \to 2^{\mathcal{X}}$ and $H : \mathcal{X} \times \mathcal{U} \to 2^{\mathcal{Y}\times\mathcal{V}}$ are respectively the *transition* and *output* maps such that

$$x(k+1) \in F(x(k), v(k)) \tag{3}$$

$$(y(k), v(k)) \in H(x(k), u(k)). \tag{4}$$

A quadruple $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y}) \in \mathcal{U}^{[0;N[} \times \mathcal{V}^{[0;N[} \times \mathcal{X}^{[0;N[} \times \mathcal{Y}^{[0;N[}$ is a *trajectory* of length N of system $\mathcal{S}$ starting at $x(0) \in \mathcal{X}$ if $N \in \mathbb{N} \cup \{\infty\}$, (3) holds for all $k \in [0; N-1[$, (4) holds for all $k \in [0; N[$. $\triangle$

The internal variable v is required to describe the interconnection of systems through set-valued output maps without introducing new state variables (see the discussion after Reissig



(a) General system (2).

(b) Static system (5).

(c) Simple system (6).

Fig. 3. System representations. Observe that (b) and (c) are particular cases of (a). The triangle blocks are *delay blocks* which represent memory elements that store and provide the previous value of a signal, e.g., $D(x(k)) = x(k-1)$.

et al. (2016, Definition III.1)). While this additional structure is not required to describe the systems we aim to control, it is essential for representing the specific class of controllers with set-valued output maps, introduced in this paper, as it will be formalized in Proposition 2.

The use of a set-valued function to describe the transition and output maps of a system allows to model perturbations and any other kind of non-determinism in a unified formalism.

We define the *behavior* of a system as its set of *maximal* trajectories.

Definition 2. Given the system $\mathcal{S}$ in (2), the *behavior* of $\mathcal{S}$ is the set $\mathcal{B}(\mathcal{S}) = \{\boldsymbol{y} \mid \exists \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{x}$ such that $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y})$ is a trajectory of length N of $\mathcal{S}$ on $[0; N[$ and if $N < \infty$, then $F(x(N-1), v(N-1)) = \varnothing\}$. $\triangle$

We classify the system (2) as follows.

Definition 3. The system $\mathcal{S}$ in (2) is

(i) *Autonomous* if $\mathcal{U}$ is a singleton[1];
(ii) *Static* if $\mathcal{X}$ is a singleton, i.e.,

$$\mathcal{S} = (\{0\}, \mathcal{U}, \mathcal{V}, \mathcal{Y}, F, H), \tag{5}$$

with $F(0, v) = \{0\}$ for all $v \in \mathcal{V}$.
(iii) *Simple* if $\mathcal{V} = \mathcal{U}$, $\mathcal{Y} = \mathcal{X}$, and $H = I_{\mathcal{X}\times\mathcal{U}}$, i.e.,

$$\mathcal{S} = (\mathcal{X}, \mathcal{U}, \mathcal{U}, \mathcal{X}, F, I_{\mathcal{X}\times\mathcal{U}}). \quad \triangle \tag{6}$$

Remark 1. Intuitively, a static system captures an input–output behavior governed by

$$y(k) \in \overline{H}(u(k)), \tag{7}$$

where the set-valued map $\overline{H} : \mathcal{U} \to 2^{\mathcal{Y}}$ is defined as $\overline{H}(u) = \{y \mid \exists v : (y, v) \in H(0, u)\}$. For the sake of simplifying notation, we denote the static system (7) by $\overline{H}$. In contrast, a simple system represents a fully state-observable system with dynamics

$$x(k+1) \in F(x(k), u(k)). \tag{8}$$

For the sake of simplifying notation, we denote the simple system (8) by the tuple $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$. We illustrate systems (2), (5) and (6) in Fig. 3.

[1] When $\mathcal{X}$, $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{Y}$ are singleton sets, we will consider, without loss of generality, the singleton $\{0\}$.

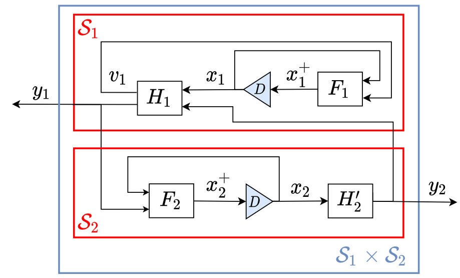


Fig. 4. Feedback composition $\mathcal{S}_1 \times \mathcal{S}_2$ of systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in (2) where $H_2'(x) = \{y_2 \mid \exists u_2 \in \mathcal{U}_2 : (y_2, u_2) \in H_2(x_2, u_2)\}$ (see Definition 5).

For a simple system $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$, we introduce the set-valued operator of *available inputs*, defined as $\mathcal{U}_{\mathcal{S}}(x) = \{\boldsymbol{u} \in \mathcal{U} \mid F(x, u) \neq \varnothing\}$, which gives the set of inputs u available at a given state x. We say that a simple system is *deterministic* if for every state $x \in \mathcal{X}$ and control input $u \in \mathcal{U}$, $F(x, u)$ is either empty or a singleton. Otherwise, we say that it is *non-deterministic*.

Given any two sets $\mathcal{U}$ and $\mathcal{Y}$, we identify a binary *relation* $R \subseteq \mathcal{U} \times \mathcal{Y}$ with set-valued maps, i.e., $R(u) = \{y \mid (u, y) \in R\}$ and $R^{-1}(y) = \{u \mid (u, y) \in R\}$. Depending on the context, we will use the symbol R interchangeably to denote the binary relation itself, the associated set-valued map $R : \mathcal{U} \to 2^{\mathcal{Y}}$, and its associated static system (5), which is then referred to as a quantizer.

We consider the *serial* and the *feedback composition* of two systems. We start by defining the serial composition of two systems (Reissig et al., 2016, Definition III.2).

Definition 4. Let $\mathcal{S}_i = (\mathcal{X}_i, \mathcal{U}_i, \mathcal{V}_i, \mathcal{Y}_i, F_i, H_i)$ be systems, $i \in \{1, 2\}$. Then, $\mathcal{S}_1$ is *serial composable* with $\mathcal{S}_2$ if $\mathcal{Y}_1 \subseteq \mathcal{U}_2$. The *serial composition* of $\mathcal{S}_1$ and $\mathcal{S}_2$, denoted by $\mathcal{S}_2 \circ \mathcal{S}_1$, is defined as

$$\mathcal{S}_2 \circ \mathcal{S}_1 = (\mathcal{X}_{12}, \mathcal{U}_1, \mathcal{V}_{12}, \mathcal{Y}_2, F_{12}, H_{12}), \tag{9}$$

where $\mathcal{X}_{12} = \mathcal{X}_1 \times \mathcal{X}_2$, $\mathcal{V}_{12} = \mathcal{V}_1 \times \mathcal{V}_2$, $F_{12} : \mathcal{X}_{12} \times \mathcal{V}_{12} \to 2^{\mathcal{X}_{12}}$, and $H_{12} : \mathcal{X}_{12} \times \mathcal{U}_1 \to 2^{\mathcal{Y}_2 \times \mathcal{V}_{12}}$ satisfy

$$\begin{aligned} F_{12}(\boldsymbol{x}, \boldsymbol{v}) &= F_1(x_1, v_1) \times F_2(x_2, v_2), \\ H_{12}(\boldsymbol{x}, u_1) &= \{(y_2, \boldsymbol{v}) \mid \exists y_1 : (y_1, v_1) \in H_1(x_1, u_1) \\ &\qquad \wedge (y_2, v_2) \in H_2(x_2, y_1)\}. \end{aligned}$$

We define the feedback composition of two systems (Reissig et al., 2016, Definition III.3) as illustrated in Fig. 4.

Definition 5. Let $\mathcal{S}_i = (\mathcal{X}_i, \mathcal{U}_i, \mathcal{V}_i, \mathcal{Y}_i, F_i, H_i)$ for $i \in \{1, 2\}$. Then, $\mathcal{S}_1$ is *feedback composable* with $\mathcal{S}_2$, denoted as $\mathcal{S}_1$ is f.c. with $\mathcal{S}_2$, if $\mathcal{Y}_2 \subseteq \mathcal{U}_1$, $\mathcal{Y}_1 \subseteq \mathcal{U}_2$, and the following hold: (i) $\mathcal{V}_2 = \mathcal{U}_2$ and $H_2(x_2, u_2) = H_2'(x_2) \times \{u_2\}$ for some $H_2' : \mathcal{X}_2 \to 2^{\mathcal{Y}_2}$; (ii) If $(y_1, v_1) \in H_1(x_1, y_2)$, $(y_2, v_2) \in H_2(x_2, y_1)$, and $F_2(x_2, v_2) = \varnothing$, then $F_1(x_1, v_1) = \varnothing$. The *feedback composition*, denoted by $\mathcal{S}_1 \times \mathcal{S}_2$, is defined as

$$\mathcal{S}_1 \times \mathcal{S}_2 = (\mathcal{X}_{12}, \{0\}, \mathcal{V}_{12}, \mathcal{Y}_{12}, F_{12}, H_{12}), \tag{10}$$

where $\mathcal{X}_{12} = \mathcal{X}_1 \times \mathcal{X}_2$, $\mathcal{V}_{12} = \mathcal{V}_1 \times \mathcal{V}_2$, $\mathcal{Y}_{12} = \mathcal{Y}_1 \times \mathcal{Y}_2$, $F_{12} : \mathcal{X}_{12} \times \mathcal{V}_{12} \to 2^{\mathcal{X}_{12}}$, and $H_{12} : \mathcal{X}_{12} \times \{0\} \to 2^{\mathcal{Y}_{12} \times \mathcal{V}_{12}}$ satisfy

$$\begin{aligned} F_{12}(\boldsymbol{x}, \boldsymbol{v}) &= F_1(x_1, v_1) \times F_2(x_2, v_2), \\ H_{12}(\boldsymbol{x}, 0) &= \{(\boldsymbol{y}, \boldsymbol{v}) \mid (y_1, v_1) \in H_1(x_1, y_2) \\ &\qquad \wedge (y_2, v_2) \in H_2(x_2, y_1)\}. \end{aligned}$$

Condition (i) in Definition 5 prevents delay-free cycles (Vidyasagar, 1981) in the closed-loop system by introducing a one-step delay. Condition (ii) will be needed later to ensure that if the concrete closed loop is non-blocking, then so is the abstract closed loop.

In our context, the system $\mathcal{S}_2$ represents the system we aim to control, while $\mathcal{S}_1$ represents the controller. Note that the system $\mathcal{S}_1 \times \mathcal{S}_2$ is autonomous.

Given $\mathcal{S}_1$ and $\mathcal{S}_2$ in Definition 5, we have $\mathcal{B}(\mathcal{S}_1 \times \mathcal{S}_2) \subseteq (\mathcal{Y}_1 \times \mathcal{Y}_2)^\infty$. To simplify further developments, we define the set $\mathcal{B}_\times(\mathcal{S}_1 \times \mathcal{S}_2)$, which contains the output trajectories of $\mathcal{S}_2$ under feedback composition with $\mathcal{S}_1$, as follows

$$\mathcal{B}_\times(\mathcal{S}_1 \times \mathcal{S}_2) = \pi_{\mathcal{Y}_2}(\mathcal{B}(\mathcal{S}_1 \times \mathcal{S}_2)), \tag{11}$$

where $\pi_{\mathcal{Y}_2}(\cdot)$ is defined in (1).

We now define the system *specifications* and, since we are focused on controller synthesis, the concept of a *control problem*.

Definition 6. Consider the system $\mathcal{S}$ in (2). A *specification* Σ for $\mathcal{S}$ is defined as any subset $\Sigma \subseteq \mathcal{Y}^\infty$. A system $\mathcal{S}$ together with a specification Σ constitute a *control problem* $(\mathcal{S}, \Sigma)$. Additionally, a system $\mathcal{C}$, called the *controller*, is said to *solve* the control problem $(\mathcal{S}, \Sigma)$ if $\mathcal{C}$ is f.c. with $\mathcal{S}$ and $\mathcal{B}_\times(\mathcal{C} \times \mathcal{S}) \subseteq \Sigma$. △

We can use linear temporal logic (Baier & Katoen, 2008, Chapter 5) to define a specification Σ for a system $\mathcal{S}$.

3. Characterization of simulation relations

3.1. Abstraction-based control

We now formalize the three-step abstraction-based control procedure described in the introduction (see Fig. 1) within our mathematical framework.

This work focuses on *state-feedback* control using an abstraction-based approach. For this reason, both the system to be controlled and its abstraction are described as simple systems, while controllers require the more general formalism of Definition 1, as it will be stated in Proposition 2. Specifically, we consider

$$\mathcal{S}_i = (\mathcal{X}_i, \mathcal{U}_i, F_i), \tag{12}$$

$$\mathcal{C}_i = (\mathcal{X}_{\mathcal{C}_i}, \mathcal{X}_i, \mathcal{V}_{\mathcal{C}_i}, \mathcal{U}_i, F_{\mathcal{C}_i}, H_{\mathcal{C}_i}), \tag{13}$$

where $\mathcal{S}_1$ is the *concrete system* and $\mathcal{S}_2$ its *abstraction*.

To solve the concrete control problem $(\mathcal{S}_1, \Sigma_1)$ with $\Sigma_1 \subseteq \mathcal{X}_1^\infty$, the abstraction-based control procedure proceeds as follows. In step 1, the system $\mathcal{S}_1$ is abstracted into a system $\mathcal{S}_2$, meaning their states are related through a relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$. Additionally, the concrete specification is abstracted into an abstract specification $\Sigma_2 \subseteq \mathcal{X}_2^\infty$, satisfying

$$R^{-1}(\Sigma_2) \subseteq \Sigma_1. \tag{14}$$

Step 2 entails designing a controller $\mathcal{C}_2$ that solves the abstract problem $(\mathcal{S}_2, \Sigma_2)$, i.e., such that

$$\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2) \subseteq \Sigma_2. \tag{15}$$

Finally, step 3 involves constructing a controller $\mathcal{C}_1$ that satisfies the *reproducibility* as defined in Calbert, Mattenet, Girard, and Jungers (2024, Property 1)

$$\mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1) \subseteq R^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2)). \tag{16}$$

The validity of this approach is established by the following inclusions, which guarantee that the controller $\mathcal{C}_1$ solves the concrete problem $(\mathcal{S}_1, \Sigma_1)$

$$\mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1) \subseteq R^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2)) \subseteq R^{-1}(\Sigma_2) \subseteq \Sigma_1$$

which follows directly from (16), (15), (14), and the monotonicity of R^{-1}.

In order to be able to construct $\mathcal{C}_1$ from $\mathcal{C}_2$ satisfying (16), the relation R must impose conditions on the local dynamics of the systems in the associated states, accounting for the impact of different inputs on state transitions. The *alternating simulation relation* (Tabuada, 2009, Definition 4.19) is a comprehensive definition of such a relation.

Definition 7. Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in (12), a relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is an *alternating simulation relation* from $\mathcal{S}_1$ to $\mathcal{S}_2$, denoted $\mathcal{S}_1 \preceq_R^{\text{ASR}} \mathcal{S}_2$, if

$$\begin{aligned} &\forall (x_1, x_2) \in R \ \forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2) \ \exists u_1 \in \mathcal{U}_{\mathcal{S}_1}(x_1) \\ &\forall x_1^+ \in F_1(x_1, u_1) \ \exists x_2^+ \in F_2(x_2, u_2) : \ (x_1^+, x_2^+) \in R. \end{aligned} \tag{17}$$

The alternating simulation relation guarantees that any controller applied to $\mathcal{S}_2$ can be concretized into a controller for $\mathcal{S}_1$ that satisfies (16).

Theorem 1 (*Tabuada (2009, Proposition 8.7)*)**.** *Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), if $\mathcal{S}_1 \preceq_R^{\text{ASR}} \mathcal{S}_2$, then for every controller $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$, there exists a controller $\mathcal{C}_1^{\text{ASR}}$ satisfying $\mathcal{B}_\times(\mathcal{C}_1^{\text{ASR}} \times \mathcal{S}_1) \subseteq R^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2))$.*

Although the alternating simulation relation provides a safety-critical framework, it has several practical drawbacks in the implementation of the concrete controller $\mathcal{C}_1$, which we will discuss in Section 4. To address these issues, various alternative relations have been introduced in the literature. Among these, the *feedback refinement relation* (Reissig et al., 2016, Definition V.2) is central to the further developments of this paper.

Definition 8. Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), a relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is a *feedback refinement relation* from $\mathcal{S}_1$ to $\mathcal{S}_2$, denoted $\mathcal{S}_1 \preceq_R^{\text{FRR}} \mathcal{S}_2$, if for every $(x_1, x_2) \in R$

$$\begin{aligned} &(i)\ \mathcal{U}_{\mathcal{S}_2}(x_2) \subseteq \mathcal{U}_{\mathcal{S}_1}(x_1); \\ &(ii)\ \forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)\ \forall x_1^+ \in F_1(x_1, u_2) : \ R(x_1^+) \neq \varnothing \\ &\quad \text{and } \ \forall x_2^+ \in R(x_1^+) : \ x_2^+ \in F_2(x_2, u_2). \end{aligned} \tag{18}$$

As a refined notion of ASR, the feedback refinement relation benefits from a straightforward concretization scheme as established by the following theorem.

Theorem 2 (*Reissig et al. (2016, Thms V.4, V.5)*)**.** *Consider the simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12). Given $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$, the following equivalence holds:*

1. *$\mathcal{S}_1 \preceq_R^{\text{FRR}} \mathcal{S}_2$;*
2. *For all $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$: $\mathcal{C}_2$ is f.c. with $R \circ \mathcal{S}_1$, $\mathcal{C}_2 \circ R$ is f.c. with $\mathcal{S}_1$, and $\mathcal{B}_\times(\mathcal{C}_2 \times (R \circ \mathcal{S}_1)) \subseteq \mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2)$.*

Additionally, it implies that

$$\mathcal{C}_1^{\text{FRR}} := \mathcal{C}_2 \circ R, \tag{19}$$

satisfies $\mathcal{B}_\times(\mathcal{C}_1^{\text{FRR}} \times \mathcal{S}_1) \subseteq R^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2))$.

The concretization procedure for FRR, defined as the serial composition of the abstract controller $\mathcal{C}_2$ with the quantizer R (19), exhibits a plug-and-play property: the control architecture (Fig. 2 (left)) remains independent of the specification to be enforced on the original system. However, this advantage comes at a cost—it constrains how the abstraction is constructed, as it forces the concretized controller to be piecewise constant. In particular, conditions (i) and (ii) in Definition 8 restrict the class of concrete controllers to those where the control input depends solely on the abstract state (19).

Thus, the feedback refinement relation is well-suited to contexts where the exact state is unknown and only abstract (or symbolic) state information is available. However, when information on the concrete state is available, this becomes a significant limitation. If the system does not exhibit local incremental stability (Angeli, 2002, Definition 2.1) (Lohmiller & Slotine, 1998), meaning that trajectories move away from each other, the use of piecewise constant controllers introduces significant non-determinism into the abstraction. This could result in an intractable or even unsolvable abstract problem $(\mathcal{S}_2, \Sigma_2)$, as illustrated by the example in Calbert, Mattenet, Girard, and Jungers (2024, Section 4.4).

Therefore, our goal is to identify, classify, and characterize refined notions of the alternating simulation relation that allow the design of more general concrete controllers than the piecewise constant controllers of the feedback refinement relation, while still benefiting from its plug-and-play property.

3.2. Key relations

In this section, we introduce refined notions of alternating simulation relations, which will be discussed throughout this paper. The first two relations are introduced in this paper, while the last is presented and discussed in Liu (2017, Definition 4) and Calbert, Mattenet, Girard, and Jungers (2024, Definition 8).

Definition 9. Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in (12), a relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is a *predictive simulation relation* from $\mathcal{S}_1$ to $\mathcal{S}_2$, denoted $\mathcal{S}_1 \preceq_R^{\mathrm{PSR}} \mathcal{S}_2$, if

$$\begin{aligned}&\forall (x_1, x_2) \in R\ \forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)\ \exists u_1 \in \mathcal{U}_{\mathcal{S}_1}(x_1)\\ &\exists x_2^+ \in F_2(x_2, u_2)\ \forall x_1^+ \in F_1(x_1, u_1):\ (x_1^+, x_2^+) \in R.\end{aligned} \tag{20}$$

Definition 10. Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in (12), a relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is a *feedforward abstraction relation* from $\mathcal{S}_1$ to $\mathcal{S}_2$, denoted $\mathcal{S}_1 \preceq_R^{\mathrm{FAR}} \mathcal{S}_2$, if

$$\begin{aligned}&\forall x_2 \in \mathcal{X}_2\ \forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)\ \exists x_2^+ \in F_2(x_2, u_2)\ \forall x_1 \in R^{-1}(x_2)\\ &\exists u_1 \in \mathcal{U}_{\mathcal{S}_1}(x_1)\ \forall x_1^+ \in F_1(x_1, u_1):\ (x_1^+, x_2^+) \in R.\end{aligned} \tag{21}$$

Definition 11. Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in (12), a relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is a *memoryless concretization relation* from $\mathcal{S}_1$ to $\mathcal{S}_2$, denoted $\mathcal{S}_1 \preceq_R^{\mathrm{MCR}} \mathcal{S}_2$, if $\forall (x_1, x_2) \in R$

$$\begin{aligned}&\forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)\ \exists u_1 \in \mathcal{U}_{\mathcal{S}_1}(x_1)\ \forall x_1^+ \in F_1(x_1, u_1)\\ &\forall x_2^+ \in R(x_1^+):\ x_2^+ \in F_2(x_2, u_2).\end{aligned} \tag{22}$$

We define the associated extended relations $R_e^{\mathrm{ASR}}, R_e^{\mathrm{PSR}}, R_e^{\mathrm{MCR}} \subseteq \mathcal{X}_2 \times \mathcal{U}_2 \times \mathcal{X}_1 \times \mathcal{U}_1$ as the sets of tuples (x_2, u_2, x_1, u_1) satisfying (17), (20), and (22) respectively. Similarly, $R_e^{\mathrm{FAR}} \subseteq \mathcal{X}_2 \times \mathcal{U}_2 \times \mathcal{X}_1 \times \mathcal{U}_1 \times \mathcal{X}_2$ is defined as the set of tuples $(x_2, u_2, x_1, u_1, x_2^+)$ satisfying (21). In addition, to extract the corresponding concrete inputs, we introduce the following set-valued maps

$$I_R^{\mathrm{T}}(x_2, u_2, x_1) = \{u_1 \mid (x_2, u_2, x_1, u_1) \in R_e^{\mathrm{T}}\}, \tag{23}$$

for $\mathrm{T} \in \{\mathrm{ASR}, \mathrm{PSR}, \mathrm{MCR}\}$, $I_R^{\mathrm{FAR}}(x_2, u_2, x_1, x_2^+) = \{u_1 \mid (x_2, u_2, x_1, u_1, x_2^+) \in R_e^{\mathrm{FAR}}\}$, $I_R^{\mathrm{FRR}}(u_2) = \{u_2\}$.

Intuitively, these maps ensure that each abstract input u_2 can be matched with a concrete input u_1 that satisfies the formula defining the relation.

The following proposition provides a partial order over the simulation relations, corresponding to the tree structure shown in Fig. 5.

Proposition 1. *Given the simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), and $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$, the following implications hold*

(1) $\mathcal{S}_1 \preceq_R^{\mathrm{FAR}} \mathcal{S}_2 \Rightarrow \mathcal{S}_1 \preceq_R^{\mathrm{PSR}} \mathcal{S}_2 \Rightarrow \mathcal{S}_1 \preceq_R^{\mathrm{ASR}} \mathcal{S}_2$.
(2) $\mathcal{S}_1 \preceq_R^{\mathrm{FRR}} \mathcal{S}_2 \Rightarrow \mathcal{S}_1 \preceq_R^{\mathrm{MCR}} \mathcal{S}_2 \Rightarrow \mathcal{S}_1 \preceq_R^{\mathrm{ASR}} \mathcal{S}_2$.

Additionally,

(3) If $\mathcal{S}_2$ is deterministic, $\mathcal{S}_1 \preceq_R^{\mathrm{ASR}} \mathcal{S}_2 \Leftrightarrow \mathcal{S}_1 \preceq_R^{\mathrm{FAR}} \mathcal{S}_2$.
(4) If R is deterministic, $\mathcal{S}_1 \preceq_R^{\mathrm{ASR}} \mathcal{S}_2 \Leftrightarrow \mathcal{S}_1 \preceq_R^{\mathrm{MCR}} \mathcal{S}_2$.
(5) If $\mathcal{S}_1 \preceq_R^{\mathrm{MCR}} \mathcal{S}_2$ and $u_1 \in I_R^{\mathrm{MCR}}(x_2, u_2, x_1) \Rightarrow u_1 = u_2$, then $\mathcal{S}_1 \preceq_R^{\mathrm{MCR}} \mathcal{S}_2 \Leftrightarrow \mathcal{S}_1 \preceq_R^{\mathrm{FRR}} \mathcal{S}_2$.

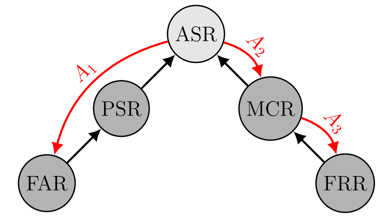


Fig. 5. Pre-order of simulation relations. For example, an arrow from PSR to ASR indicates that $\mathcal{S}_1 \preceq_R^{\mathrm{PSR}} \mathcal{S}_2$ implies $\mathcal{S}_1 \preceq_R^{\mathrm{ASR}} \mathcal{S}_2$. Red arrows represent implications that hold under specific assumptions labeled on the arrows: A_1 denotes that $\mathcal{S}_2$ is deterministic, A_2 that R is deterministic, and A_3 that $u_1 = u_2$ as in Proposition 1.

Proof.

(1), (2) These follow from basic logical manipulations.

(3) If $\mathcal{S}_2$ is deterministic, then for any $x_2 \in \mathcal{X}_2$ and $u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$: $|F_2(x_2, u_2)| = 1$. Thus, (17) is equivalent to: $\forall (x_1, x_2) \in R\ \forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)\ x_2^+ \in F_2(x_2, u_2)\ \exists u_1 \in \mathcal{U}_{\mathcal{S}_1}(x_1)\ \forall x_1^+ \in F_1(x_1, u_1):\ (x_1^+, x_2^+) \in R$, which in turn is equivalent to (21).

(4) If R is deterministic, then for any $(x_1, x_2) \in R$: $|R(x_1)| = 1$. Therefore $R(x_1^+) \cap F_2(x_2, u_2) \neq \varnothing$ if and only if $R(x_1^+) \neq \varnothing$, $R(x_1^+) \subseteq F_2(x_2, u_2)$.

(5) When the abstract and concrete inputs in the extended relation are identical, (22) simplifies to (18). ■

3.3. Main results

In this section, we present our main results, which provide an exact characterization of simulation relations in terms of their associated concretization procedures. To this end, we introduce the notion of an *interface*, which connects the system $\mathcal{S}_1$ to any controller $\mathcal{C}_2$ functionally compatible with $\mathcal{S}_2$, as illustrated in Fig. 2 (right).

Definition 12. Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), an *interface* from $\mathcal{S}_2$ to $\mathcal{S}_1$ is defined by the tuple $(\mathcal{Z}_1, h_1, h_2, \widetilde{R})$, where $\mathcal{Z}_1$ is a set, h_1 and h_2 are set-valued functions, and $\widetilde{R} \subseteq (\mathcal{X}_1 \times \mathcal{Z}_1) \times \mathcal{X}_2$. Consider the set of variables $\nu = \{x_1, x_1^+, z_1, z_1^+, u_1, u_2\}$, where $x_1, x_1^+ \in \mathcal{X}_1$, $z_1, z_1^+ \in \mathcal{Z}_1$, $u_1 \in \mathcal{U}_1$, and $u_2 \in \mathcal{U}_2$. Let h_1 (resp. h_2) be a set-valued function with argument ν_1 (resp. ν_2), a subset of the set ν, returning a subset of $\mathcal{U}_1$ (resp. $\mathcal{Z}_1$). The domain of h_1 (resp. h_2) is denoted as $\mathcal{V}_1$ (resp. $\mathcal{V}_2$), defined as the Cartesian product of the domains of the corresponding variables, i.e., $h_1 : \mathcal{V}_1 \rightarrow 2^{\mathcal{U}_1}$ and $h_2 : \mathcal{V}_2 \rightarrow 2^{\mathcal{Z}_1}$. The functions h_1 and h_2 ensure that the following holds. For all $((x_1, z_1), x_2) \in \widetilde{R}$ and $u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$, it holds that any u_1, z_1^+, x_1^+ and x_2^+ which are solution of the following equations

$$\begin{aligned}u_1 \in h_1(\nu_1),\ \ &z_1^+ \in h_2(\nu_2),\\ x_1^+ \in F_1(x_1, u_1),\ \ &x_2^+ \in \widetilde{R}((x_1^+, z_1^+)),\end{aligned} \tag{24}$$

satisfy $x_2^+ \in F_2(x_2, u_2)$. △

Intuitively, the variable z_1 serves as a state of the interface, allowing it to retain memory across transitions. Given $((x_1, z_1), x_2) \in \widetilde{R}$, the function h_1 selects a concrete input u_1 such that the resulting concrete successor $x_1^+ \in F_1(x_1, u_1)$ remains related to an abstract successor $x_2^+ \in F_2(x_2, u_2)$ via a new interface state $z_1^+ \in h_2(\nu_2)$, ensuring $((x_1^+, z_1^+), x_2^+) \in \widetilde{R}$. In this way, the interface coordinates the evolution of the concrete system so that it stays related to the abstract system for any input $u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$, i.e., for any abstract controller $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$.

We now introduce the notion of an *interfaced system*, which encapsulates the system $\mathcal{S}_1$ together with the interface, as illustrated in Fig. 7(a).

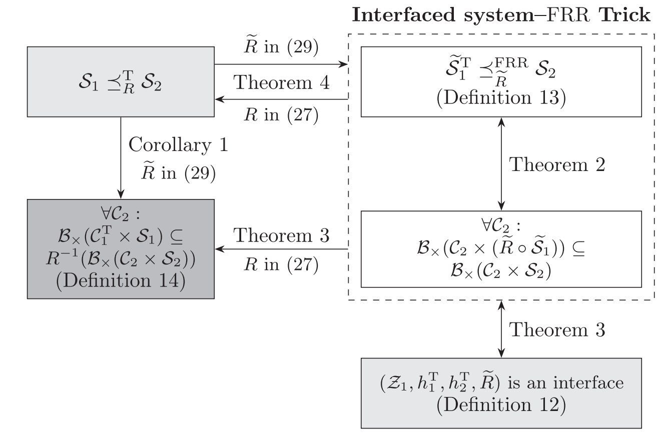


Fig. 6. Summary of the contributions in Section 3.3. The diagram illustrates how the main theorems and definitions are connected to characterize a simulation relation T in terms of a specific control architecture, defined by the interface $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$. It also shows how the resulting concretized controller $\mathcal{C}_1^{\mathrm{T}}$ ensures the reproducibility condition (16).

Definition 13. Given the simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in (12), and an interface $(\mathcal{Z}_1, h_1, h_2, \widetilde{R})$ from $\mathcal{S}_2$ to $\mathcal{S}_1$, we define the associated *interfaced system* $\widetilde{\mathcal{S}}_1$ as the simple system $\widetilde{\mathcal{S}}_1 = (\widetilde{\mathcal{X}}_1, \widetilde{\mathcal{U}}_1, \widetilde{F}_1)$, where $\widetilde{\mathcal{X}}_1 = \mathcal{X}_1 \times \mathcal{Z}_1, \widetilde{\mathcal{U}}_1 = \mathcal{U}_2$, and

$$\widetilde{F}_1((x_1, z_1), u_2) = \{(x_1^+, z_1^+) \mid u_1 \in h_1(\nu_1),\ z_1^+ \in h_2(\nu_2),\ x_1^+ \in F_1(x_1, u_1)\}. \quad \triangle \tag{25}$$

Next, we define the controller $\mathcal{C}_1$, referred to as the *concretized controller*, which results from connecting the controller $\mathcal{C}_2$ to the interface, as illustrated in Fig. 7(a).

Definition 14. Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), an interface $(\mathcal{Z}_1, h_1, h_2, \widetilde{R})$ from $\mathcal{S}_2$ to $\mathcal{S}_1$, and a controller $\mathcal{C}_2$ in (13) f.c. with $\mathcal{S}_2$, a *concretized controller* is any system $\mathcal{C}_1$ in (13) that satisfies the following closed-loop condition: $\boldsymbol{x}_1 \in \mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1)$ if and only if there exists $\boldsymbol{u}_1, \boldsymbol{z}_1, \boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{x}_{\mathcal{C}_2}, \boldsymbol{v}_{\mathcal{C}_2}$ such that

$$\begin{array}{l} \boxed{\begin{array}{l}(u_2(k), v_{\mathcal{C}_2}(k)) \in H_{\mathcal{C}_2}(x_{\mathcal{C}_2}(k), x_2(k)) \\ x_{\mathcal{C}_2}(k+1) \in F_{\mathcal{C}_2}(x_{\mathcal{C}_2}(k), v_{\mathcal{C}_2}(k))\end{array}} \\ \boxed{\begin{array}{l}u_1(k) \in h_1(\nu_1(k)) \\ z_1(k+1) \in h_2(\nu_2(k)) \\ x_2(k+1) \in \widetilde{R}((x_1(k+1), z_1(k+1)))\end{array}} \\ \boxed{x_1(k+1) \in F_1(x_1(k), u_1(k)).} \end{array} \tag{26}$$

$\triangle$

Eq. (26) describes the interaction between the dynamics of the abstract controller, the interface, and the concrete system, respectively.

In order to establish the equivalence between a simulation relation and an interface, we will show that the interfaced system is always in a feedback refinement relation with the abstract one, as illustrated in Fig. 7(a). This insight will allow us to leverage the two-way characterization of FRR (Theorem 2) and form the core of our proof strategy, summarized in Fig. 6.

The following theorem shows that an interface induces a feedback refinement relation between the interfaced system and the abstract one, and formalizes the role of interfaces in the controller concretization process.

Theorem 3. *Consider two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), and the interfaced system $\widetilde{\mathcal{S}}_1$ (Definition 13) associated with the tuple $(\mathcal{Z}_1, h_1, h_2, \widetilde{R})$. The following propositions are equivalent.*

(1) $(\mathcal{Z}_1, h_1, h_2, \widetilde{R})$ is an interface from $\mathcal{S}_2$ to $\mathcal{S}_1$;
(2) $\widetilde{\mathcal{S}}_1 \preceq^{\mathrm{FRR}}_{\widetilde{R}} \mathcal{S}_2$;
(3) for any controller $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$, it holds that $\mathcal{C}_2$ is f.c. with $\widetilde{R} \circ \widetilde{\mathcal{S}}_1$ and $\widetilde{\mathcal{C}}_1 = \mathcal{C}_2 \circ \widetilde{R}$ is f.c. with $\widetilde{\mathcal{S}}_1$, and $\mathcal{B}_\times(\mathcal{C}_2 \times (\widetilde{R} \circ \widetilde{\mathcal{S}}_1)) \subseteq \mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2)$.

Additionally, for any controllers $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$, the concretized controller $\mathcal{C}_1$ (Definition 14) satisfies (16), i.e., $\mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1) \subseteq R^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2))$, with

$$R = \{(x_1, x_2) \mid \exists z_1 \in \mathcal{Z}_1 \text{ s.t. } ((x_1, z_1), x_2) \in \widetilde{R}\}. \tag{27}$$

Proof. We first establish the equivalences.

(**1**) ⇔ (**2**): The tuple $(\mathcal{Z}_1, h_1, h_2, \widetilde{R})$ is an interface if and only if it satisfies (24). This is equivalent to requiring that for any $((x_1, z_1), x_2) \in \widetilde{R}$ and $u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$, (*i*) $\widetilde{F}_1((x_1, z_1), u_2) \neq \varnothing$, and (*ii*) for all $(x_1^+, z_1^+) \in \widetilde{F}_1((x_1, z_1), u_2)$, $\widetilde{R}((x_1^+, z_1^+)) \neq \varnothing$ and $\widetilde{R}((x_1^+, z_1^+)) \subseteq F_2(x_2, u_2)$. This, in turn, is equivalent to conditions (*i*) and (*ii*) of Definition 8, meaning that $\widetilde{\mathcal{S}}_1 \preceq^{\mathrm{FRR}}_{\widetilde{R}} \mathcal{S}_2$.

(**2**) ⇔ (**3**): Directly follows from Theorem 2.

In addition, Theorem 2 implies

$$\mathcal{B}_\times((\mathcal{C}_2 \circ \widetilde{R}) \times \widetilde{\mathcal{S}}_1) \subseteq \widetilde{R}^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2)). \tag{28}$$

Let $\boldsymbol{x}_1 \in \mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1)$ be defined on $[0; N[$, where $N \in \mathbb{N} \cup \{\infty\}$. Then, by Lemma 1, there exists a map $\boldsymbol{z}_1$ such that $\widetilde{\boldsymbol{x}}_1 \in \mathcal{B}_\times((\mathcal{C}_2 \circ \widetilde{R}) \times \widetilde{\mathcal{S}}_1)$ with $\widetilde{x}_1(k) = (x_1(k), z_1(k))$ for all $k \in [0; N[$. Then, by (28), there exists a map $\boldsymbol{x}_2 \in \mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2)$ such that $x_2(k) \in \widetilde{R}((x_1(k), z_1(k)))$ for all $k \in [0; N[$. By Lemma 2, since R satisfies (27), we can conclude that $x_2(k) \in R(x_1(k))$ for all $k \in [0; N[$, i.e., $\boldsymbol{x}_2 \in R(\boldsymbol{x}_1)$. Thus $\mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1) \subseteq R^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2))$. ■

In the next definition, we construct for each simulation relation T, the corresponding interface structure, solely determined by the variables ν_1 and ν_2 that define the domains of the functions h_1^{T} and h_2^{T}, as illustrated in Fig. 7.

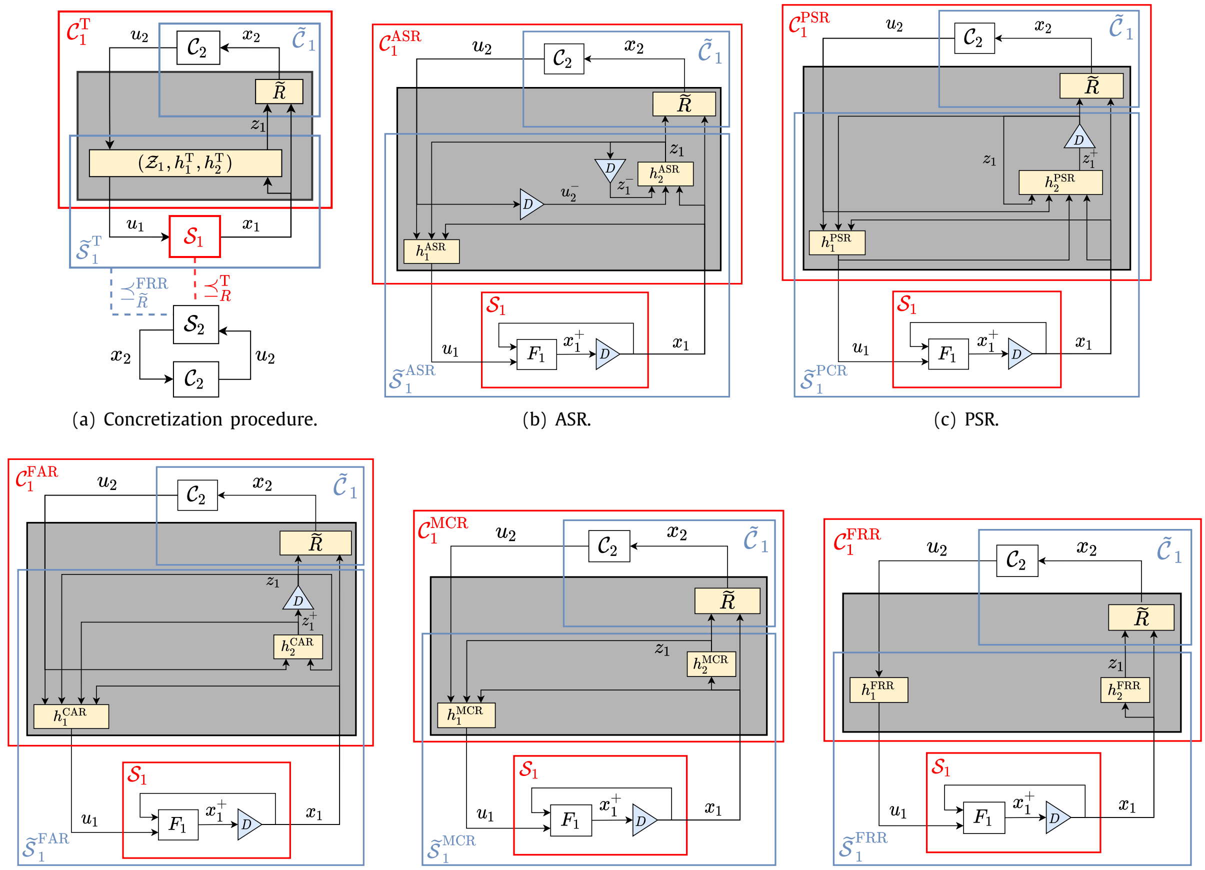
Definition 15. For each T $\in$ {ASR, PSR, FAR, MCR, FRR}, we associate a tuple $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$, where the functions h_1^{T} and h_2^{T} take as arguments the variables ν_1 and ν_2, respectively, defined as follows

$$\begin{aligned} \mathrm{ASR} &: \nu_1 = (z_1, u_2, x_1),\ \ \nu_2 = (z_1, u_2, x_1^+); \\ \mathrm{PSR} &: \nu_1 = (z_1, u_2, x_1),\ \ \nu_2 = (z_1, u_2, x_1, u_1); \\ \mathrm{FAR} &: \nu_1 = (z_1, u_2, x_1, z_1^+),\ \ \nu_2 = (z_1, u_2); \\ \mathrm{MCR} &: \nu_1 = (z_1, u_2, x_1),\ \ \nu_2 = (x_1^+); \\ \mathrm{FRR} &: \nu_1 = (u_2),\ \ \nu_2 = (x_1^+). \end{aligned}$$

The arguments ν_1 and ν_2 are directly inferred from the logical structure of the formula defining the relation T. More precisely, ν_1 comprises the variables (with x_2 and x_2^+ replaced by z_1 and z_1^+) that appear before the existential quantifier for u_1, while ν_2 includes those appearing before the quantifier for x_2^+. Intuitively, this reflects the role of h_1^{T} in selecting the concrete input u_1, and h_2^{T} in determining a consistent successor z_1^+, to satisfy the formula defining the relation. For example, in the case of ASR, the quantified variables in (17) are $x_2, u_2, x_1, u_1, x_1^+$, and x_2^+. Thus, u_1 depends on $\nu_1 = (z_1, u_2, x_1)$, and x_2^+ depends on $\nu_2 = (z_1, u_2, x_1^+)$, which defines the interface of ASR, as illustrated in Fig. 7(b).

Having identified in Definition 15 the *structure* of the interface for each simulation relation T, we now provide a specific *implementation* of h_1^{T} and h_2^{T} for the specific case where $\mathcal{Z}_1 = \mathcal{X}_2$, i.e., $z_1 = x_2$ and $z_1^+ = x_2^+$.

Definition 16. For each T $\in$ {ASR, PSR, FAR, MCR, FRR}, given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12) and a relation



(a) Concretization procedure. (b) ASR. (c) PSR.

(d) FAR. (e) MCR. (f) FRR.

Fig. 7. Concretization of an abstract controller $\mathcal{C}_2$ into a concrete controller $\mathcal{C}_1^{\mathrm{T}}$ using the interface $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$ for the simulation relation T. (a) Depicts the interfaced system $\tilde{\mathcal{S}}_1^{\mathrm{T}}$, where $\tilde{\mathcal{S}}_1 \preceq_{\widetilde{R}}^{\mathrm{FRR}} \mathcal{S}_2$, with controller $\tilde{\mathcal{C}}_1 = \mathcal{C}_2 \circ \widetilde{R}$ (19). Dotted lines show the simulation relations between systems. The remaining subfigures illustrate implementations (Definition 15) for T ∈ {ASR, PSR, FAR, MCR, FRR}.

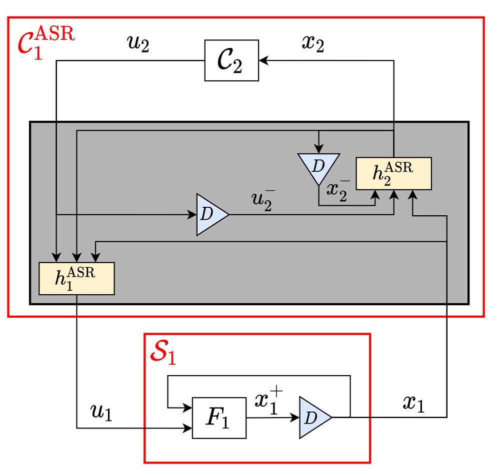


Fig. 8. Simplified control architecture of Fig. 7(b) for ASR, based on the specific interface of Definition 16.

$R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$, we define the tuple $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$ as follows. We set $\mathcal{Z}_1 = \mathcal{X}_2$, $h_1^{\mathrm{T}} = I_R^{\mathrm{T}}$ with I_R^{T} defined in (23), and

$$
\begin{aligned}
h_2^{\mathrm{ASR}}(x_2, u_2, x_1^+) =& F_2(x_2, u_2) \cap R(x_1^+),\\
h_2^{\mathrm{PSR}}(x_2, u_2, x_1, u_1) =& F_2(x_2, u_2) \cap \{x_2^+ \in \mathcal{X}_2 :\\
& F_1(x_1, u_1) \subseteq R^{-1}(x_2^+)\},\\
h_2^{\mathrm{FAR}}(x_2, u_2) =& F_2(x_2, u_2) \cap \{x_2^+ \in \mathcal{X}_2 :\\
\forall x_1 \in R^{-1}(x_2) \exists u_1 \in& \mathcal{U}_{\mathcal{S}_1}(x_1) : F_1(x_1, u_1) \subseteq R^{-1}(x_2^+)\},\\
h_2^{\mathrm{FRR}}(x_1^+) =& h_2^{\mathrm{MCR}}(x_1^+) = R(x_1^+),
\end{aligned}
$$

and define the lifted relation as

$$
\widetilde{R} = \{((x_1, x_2), x_2) \mid (x_1, x_2) \in R\}. \quad \triangle \tag{29}
$$

In the setting of Definition 16, the function h_1^{T} selects a concrete input u_1 such that the successor $x_1^+ \in F_1(x_1, u_1)$ remains related, i.e., $(x_1^+, x_2^+) \in R$, to an abstract successor $x_2^+ \in F_2(x_2, u_2)$, identified by h_2^{T}. The resulting control architecture for ASR is illustrated in Fig. 8, which simplifies the general architecture for ASR of Definition 15 shown in Fig. 7(b).

Definition 17. A relation $\widetilde{R} \subseteq (\mathcal{X}_1 \times \mathcal{Z}_1) \times \mathcal{X}_2$ satisfies the *common quantization assumption* if $\forall x_1, x_1' \in \mathcal{X}_1$, $z_1 \in \mathcal{Z}_1$ such that $\widetilde{R}((x_1, z_1)) \neq \varnothing$ and $\widetilde{R}((x_1', z_1)) \neq \varnothing$ implies $\widetilde{R}(x_1, z_1) \cap \widetilde{R}((x_1', z_1)) \neq \varnothing$. △

The following theorem establishes that the existence of a simulation relation between a system and its abstraction is equivalent to the existence of an interface, i.e., of a feedback refinement relation between the corresponding interfaced system and the abstraction.

Theorem 4. *Given* T ∈ {ASR, PSR, FAR, MCR, FRR} *and the simple systems* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *in (12), the following propositions hold.*

(1) If there exists $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ such that $\mathcal{S}_1 \preceq_R^{\mathrm{T}} \mathcal{S}_2$, then $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$ given in Definition 16 is an interface from $\mathcal{S}_2$ to $\mathcal{S}_1$, with $\widetilde{R}$ satisfying Definition 17.

(2) Conversely, if there exists an interface $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$ as given in Definition 15, where $\widetilde{R}$ satisfies Definition 17, then $\mathcal{S}_1 \preceq_R^{\mathrm{T}} \mathcal{S}_2$ with R in (27).

Proof. See Appendix A. ■

The following corollary of Theorem 4 shows that the existence of a simulation relation between the original system and its abstraction is not only sufficient to ensure the simple closed-loop structure in Fig. 2 (right), but also necessary. It also confirms that the specific interface implementation given in Definition 16 is valid.

Corollary 1. *Let $\mathrm{T} \in \{\mathrm{ASR}, \mathrm{PSR}, \mathrm{FAR}, \mathrm{MCR}, \mathrm{FRR}\}$ and two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12). The following propositions are equivalent.*

(1) There exists $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ such that $\mathcal{S}_1 \preceq_R^{\mathrm{T}} \mathcal{S}_2$;

(2) There exists an interface $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$ from $\mathcal{S}_2$ to $\mathcal{S}_1$ in Definition 15 with $\widetilde{R}$ satisfying Definition 17;

with R and $\widetilde{R}$ satisfying (27). Additionally, for any controllers $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$, the concretized controller $\mathcal{C}_1^{\mathrm{T}}$ (Definition 14) associated with the interface in Definition 16 satisfies $\mathcal{B}_\times(\mathcal{C}_1^{\mathrm{T}} \times \mathcal{S}_1) \subseteq R^{-1}(\mathcal{B}_\times(\mathcal{C}_2 \times \mathcal{S}_2))$.

Proof. It directly follows from Theorem 3 and Theorem 4.

Remark 2. The assumption that $\widetilde{R}$ satisfies Definition 17 in Theorem 4 and Corollary 1 is only required for $\mathrm{T} \in \{\mathrm{PSR}, \mathrm{FAR}\}$, and is automatically satisfied by the relation $\widetilde{R}$ in (29), as stated in Theorem 4. △

Finally, we define in our control formalism (Section 2) the concretized controllers $\mathcal{C}_1^{\mathrm{T}}$ obtained by connecting $\mathcal{C}_2$ to the interfaces $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$ of Definition 15, as illustrated in Fig. 7.

Proposition 2. *Given $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), the interfaces from Definition 15, and a controller $\mathcal{C}_2$ as in (13), let $\widetilde{\mathcal{C}}_1 := \mathcal{C}_2 \circ \widetilde{R}$. Then, the controllers $\mathcal{C}_1^{\mathrm{T}} = (\mathcal{X}_{\mathcal{C}_1^{\mathrm{T}}}, \mathcal{X}_1, \mathcal{V}_{\mathcal{C}_1^{\mathrm{T}}}, \mathcal{U}_1, F_{\mathcal{C}_1^{\mathrm{T}}}, H_{\mathcal{C}_1^{\mathrm{T}}})$ satisfy Definition 14.*

- $\mathcal{X}_{\mathcal{C}_1^{\mathrm{ASR}}} = \mathcal{X}_{\widetilde{\mathcal{C}}_1} \times \mathcal{Z}_1 \times \mathcal{U}_2,\ \mathcal{V}_{\mathcal{C}_1^{\mathrm{ASR}}} = \mathcal{V}_{\widetilde{\mathcal{C}}_1} \times \mathcal{Z}_1 \times \mathcal{U}_2,$
$$F_{\mathcal{C}_1^{\mathrm{ASR}}}((x_{\widetilde{\mathcal{C}}_1}, z_1^-, u_2^-), (v_{\widetilde{\mathcal{C}}_1}, z_1, u_2)) = \{(x_{\widetilde{\mathcal{C}}_1}^+, z_1, u_2) \mid x_{\widetilde{\mathcal{C}}_1}^+ \in F_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, v_{\widetilde{\mathcal{C}}_1})\},$$
$$H_{\mathcal{C}_1^{\mathrm{ASR}}}((x_{\widetilde{\mathcal{C}}_1}, z_1^-, u_2^-), x_1) = \{(u_1, (v_{\widetilde{\mathcal{C}}_1}, z_1, u_2)) \mid z_1 \in h_2^{\mathrm{ASR}}(z_1^-, u_2^-, x_1), (u_2, v_{\widetilde{\mathcal{C}}_1}) \in H_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, (x_1, z_1)), u_1 \in h_1^{\mathrm{ASR}}(z_1, u_2, x_1)\}.$$
- $\mathcal{X}_{\mathcal{C}_1^{\mathrm{PSR}}} = \mathcal{X}_{\widetilde{\mathcal{C}}_1} \times \mathcal{Z}_1,\ \mathcal{V}_{\mathcal{C}_1^{\mathrm{PSR}}} = \mathcal{V}_{\widetilde{\mathcal{C}}_1} \times \mathcal{Z}_1,$
$$F_{\mathcal{C}_1^{\mathrm{PSR}}}((x_{\widetilde{\mathcal{C}}_1}, z_1), (v_{\widetilde{\mathcal{C}}_1}, z_1^+)) = \{(x_{\widetilde{\mathcal{C}}_1}^+, z_1^+) \mid x_{\widetilde{\mathcal{C}}_1}^+ \in F_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, v_{\widetilde{\mathcal{C}}_1})\},$$
$$H_{\mathcal{C}_1^{\mathrm{PSR}}}((x_{\widetilde{\mathcal{C}}_1}, z_1), x_1) = \{(u_1, (v_{\widetilde{\mathcal{C}}_1}, z_1^+)) \mid (u_2, v_{\widetilde{\mathcal{C}}_1}) \in H_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, (x_1, z_1)), u_1 \in h_1^{\mathrm{PSR}}(z_1, u_2, x_1), z_1^+ \in h_2^{\mathrm{PSR}}(z_1, u_2, x_1, u_1)\}.$$
- $\mathcal{X}_{\mathcal{C}_1^{\mathrm{FAR}}} = \mathcal{X}_{\widetilde{\mathcal{C}}_1} \times \mathcal{Z}_1,\ \mathcal{V}_{\mathcal{C}_1^{\mathrm{FAR}}}, = \mathcal{V}_{\widetilde{\mathcal{C}}_1} \times \mathcal{Z}_1,$
$$F_{\mathcal{C}_1^{\mathrm{FAR}}}((x_{\widetilde{\mathcal{C}}_1}, z_1), (v_{\widetilde{\mathcal{C}}_1}, z_1^+)) = \{(x_{\widetilde{\mathcal{C}}_1}^+, z_1^+) \mid x_{\widetilde{\mathcal{C}}_1}^+ \in F_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, v_{\widetilde{\mathcal{C}}_1})\},$$
$$H_{\mathcal{C}_1^{\mathrm{FAR}}}((x_{\widetilde{\mathcal{C}}_1}, z_1), x_1) = \{(u_1, (v_{\widetilde{\mathcal{C}}_1}, z_1^+)) \mid (u_2, v_{\widetilde{\mathcal{C}}_1}) \in H_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, (x_1, z_1)), z_1^+ \in h_2^{\mathrm{FAR}}(z_1, u_2), u_1 \in h_1^{\mathrm{FAR}}(z_1, u_2, x_1, z_1^+)\}.$$
- $\mathcal{X}_{\mathcal{C}_1^{\mathrm{MCR}}} = \mathcal{X}_{\widetilde{\mathcal{C}}_1},\ \mathcal{V}_{\mathcal{C}_1^{\mathrm{MCR}}} = \mathcal{V}_{\widetilde{\mathcal{C}}_1},$
$$F_{\mathcal{C}_1^{\mathrm{MCR}}}(x_{\widetilde{\mathcal{C}}_1}, v_{\widetilde{\mathcal{C}}_1}) = F_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, v_{\widetilde{\mathcal{C}}_1}),$$
$$H_{\mathcal{C}_1^{\mathrm{MCR}}}(x_{\widetilde{\mathcal{C}}_1}, x_1) = \{(u_1, v_{\widetilde{\mathcal{C}}_1}) \mid z_1 \in h_2^{\mathrm{MCR}}(x_1), (u_2, v_{\widetilde{\mathcal{C}}_1}) \in H_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, (x_1, z_1)), u_1 \in h_1^{\mathrm{MCR}}(z_1, u_2, x_1)\}.$$
- $\mathcal{X}_{\mathcal{C}_1^{\mathrm{FRR}}} = \mathcal{X}_{\widetilde{\mathcal{C}}_1},\ \mathcal{V}_{\mathcal{C}_1^{\mathrm{FRR}}} = \mathcal{V}_{\widetilde{\mathcal{C}}_1},$
$$F_{\mathcal{C}_1^{\mathrm{FRR}}}(x_{\widetilde{\mathcal{C}}_1}, v_{\widetilde{\mathcal{C}}_1}) = F_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, v_{\widetilde{\mathcal{C}}_1}),$$
$$H_{\mathcal{C}_1^{\mathrm{FRR}}}(x_{\widetilde{\mathcal{C}}_1}, x_1) = \{(u_1, v_{\widetilde{\mathcal{C}}_1}) \mid z_1 \in h_2^{\mathrm{FRR}}(x_1), (u_2, v_{\widetilde{\mathcal{C}}_1}) \in H_{\widetilde{\mathcal{C}}_1}(x_{\widetilde{\mathcal{C}}_1}, (x_1, z_1)), u_1 \in h_1^{\mathrm{FRR}}(u_2)\}.$$

Proof. This follows from Theorem 4 and Definition 5. ■

Proposition 2 provides an implementation of the controller $\mathcal{C}_1^{\mathrm{ASR}}$ of Theorem 1 in our control framework. In addition, we recover the controller $\mathcal{C}_1^{\mathrm{FRR}}$ of Theorem 2 using our framework (Proposition 2) in the setting of Definition 16. Indeed, in that case $H_{\mathcal{C}_1^{\mathrm{FRR}}}(x_{\mathcal{C}_2}, x_1) = \{(u_2, v_{\mathcal{C}_2}) \mid x_2 \in R(x_1), (u_2, v_{\mathcal{C}_2}) \in H_{\mathcal{C}_2}(x_{\mathcal{C}_2}, x_2)\} = H_{\mathcal{C}_2}(x_{\mathcal{C}_2}, R(x_1))$, establishing that $\mathcal{C}_1^{\mathrm{FRR}} = \mathcal{C}_2 \circ R$ as in (19). Consequently, for $\mathrm{T} = \mathrm{FRR}$, the architecture in Fig. 7(f) reduces to that in Fig. 2 (left), and Corollary 1 recovers Theorem 2. Therefore, Corollary 1 generalizes the concretization result of Theorem 2 to simulation relations beyond the feedback refinement relation.

To summarize, we introduced a general framework for characterizing simulation relations through their associated control architectures by defining a generic interface structure based on two set-valued functions, h_1^{T} and h_2^{T}, which systematically capture the concretization mechanism for any relation refining the ASR. Importantly, the *arguments* of the functions h_1^{T} and h_2^{T}—which define the block representation of the interface (see Fig. 7)—depend only on the type of relation T, and not on the specific systems $\mathcal{S}_1$ and $\mathcal{S}_2$ (see Definition 15). While the *implementation* of these blocks does depend on the systems involved (see Definition 16), it remains independent of the abstract controller $\mathcal{C}_2$. This makes the interface specification-agnostic, enabling a plug-and-play concretization procedure as outlined in the introduction.

4. Properties

This section analyzes how each previously introduced simulation relation affects the abstraction structure, controller complexity, and achievable performance. These aspects are illustrated throughout using the simple one-dimensional running example shown in Fig. 9.

4.1. The concretization complexity issue

The implementations of h_2^{ASR}, h_2^{PSR}, and h_2^{FAR} in Definition 16 require online evaluation of F_2 of $\mathcal{S}_2$. Given the typically large state space of $\mathcal{S}_2$, this becomes computationally demanding, as the concretized controllers $\mathcal{C}_1^{\mathrm{ASR}}$, $\mathcal{C}_1^{\mathrm{PSR}}$, and $\mathcal{C}_1^{\mathrm{FAR}}$ must store abstract variables and query the abstraction at runtime. In contrast, h_2^{MCR} depends solely on the quantizer R, not on F_2, making the MCR concretization considerably simpler.

In the system described in Fig. 9, the optimal abstract policy for ASR, achieves a worst-case cost of 3 transitions ($q_0 \to q_2 \to q_3 \to q_4$). However, to ensure that a concrete successor x_1^+ is quantized into q_2—and not mistakenly into q_1—the concretized controller must compute $x_2^+ \in h_2^{\mathrm{ASR}}(x_2, u_2, x_1^+) = F_2(x_2, u_2) \cap$

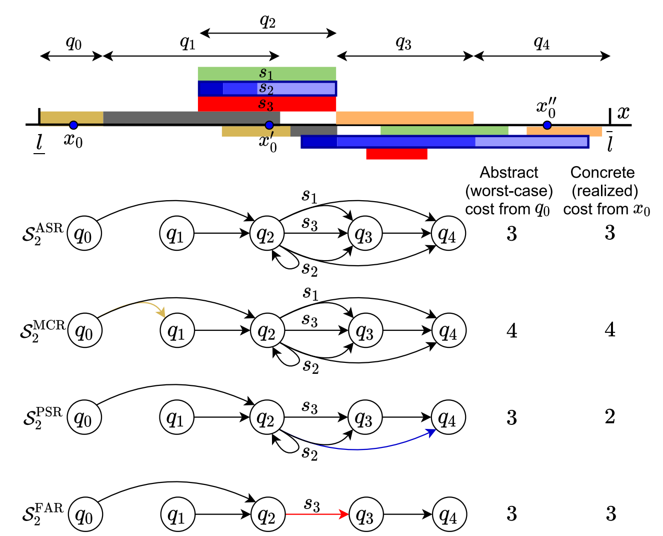


Fig. 9. Running example (1-dimensional) used to illustrate the properties of the simulation relations. The deterministic system $\mathcal{S}_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1)$ with $\mathcal{X}_1 \subseteq [\underline{l}, \bar{l}] \subset \mathbb{R}$, is discretized into five abstract cells $q_0, \dots, q_4 \in \mathcal{X}_2$. The relation is defined by $R(x) = \{q \in \mathcal{X}_2 \mid x \in q\}$. Cells q_1 and q_2 partially overlap, which makes R non-deterministic on this region. Above the x-axis, the intervals represent subsets of $\mathcal{X}_1$; below, in the same color, we show their closed-loop images $F_1(x, \kappa(x))$ under local feedbacks $\kappa(\cdot)$, which define the abstract inputs (only a few are labeled $s_1, s_2, s_3 \in \mathcal{U}_2$ for clarity). The objective is to drive the system from q_0 to q_4 in the minimum number of steps. The figure also reports the optimal abstract and concrete cost with initial condition $x_0 \in q_0$ obtained for each abstraction $\mathcal{S}_2^{\mathrm{T}}$, for which it can be verified that $\mathcal{S}_1 \preceq_R^{\mathrm{T}} \mathcal{S}_2^{\mathrm{T}}$.

$R(x_1^+)$, thus storing (x_2, u_2) and accessing F_2 online. This illustrates the *concretization complexity* discussed above.

Under MCR, the abstract transition is conservatively enlarged to include both q_1 and q_2 in $F_2(q_0, \cdot)$, which allows the concretized controller to compute $h_2^{\mathrm{MCR}}(x_1^+) = R(x_1^+)$ directly. This removes the need for storing abstract variables x_2 and u_2, and for evaluating the abstract transition map F_2 during online execution. The concretized controller $\mathcal{C}_1^{\mathrm{MCR}}$ is therefore static, as formalized in the next proposition; however, the added non-determinism increases the worst-case cost from 3 to 4 ($q_0 \to q_1 \to q_2 \to q_3 \to q_4$).

Even when the abstract controller $\mathcal{C}_2$ is static (i.e., $\mathcal{X}_{\mathcal{C}_2} = \{0\}$), the concretized controllers $\mathcal{C}_1^{\mathrm{ASR}}$, $\mathcal{C}_1^{\mathrm{PSR}}$, and $\mathcal{C}_1^{\mathrm{FAR}}$ remain dynamic, as they must store abstract variables—represented by the delay blocks in Fig. 7. In contrast, the concretized controller $\mathcal{C}_1^{\mathrm{MCR}}$ introduces no additional state (no delay block in Fig. 7(e)). Hence, if $\mathcal{C}_2$ is static, $\mathcal{C}_1^{\mathrm{MCR}}$ is also static, as formally stated in the following proposition.

Proposition 3. *Given two simple systems $\mathcal{S}_1$ and $\mathcal{S}_2$ as in (12), a relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$, and a controller $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$, if $\mathcal{C}_2$ is static, then the corresponding concretized controller $\mathcal{C}_1^{\mathrm{MCR}}$ (as defined in Proposition 2) is also static.*

Proof. This follows directly from Proposition 2, where $\mathcal{X}_{\mathcal{C}_1^{\mathrm{MCR}}} = \mathcal{X}_{\mathcal{C}_2}$, $\mathcal{V}_{\mathcal{C}_1^{\mathrm{MCR}}} = \mathcal{V}_{\mathcal{C}_2}$, and $F_{\mathcal{C}_1^{\mathrm{MCR}}} = F_{\mathcal{C}_2}$. Therefore, if $\mathcal{C}_2$ is static (i.e., $\mathcal{X}_{\mathcal{C}_2} = \{0\}$), then $\mathcal{X}_{\mathcal{C}_1^{\mathrm{MCR}}} = \{0\}$, confirming that $\mathcal{C}_1^{\mathrm{MCR}}$ is static. ■

4.2. Corrective vs predictive control architecture

The architecture induced by ASR is *corrective*: the controller stores the current abstract state x_2 and input u_2 to adjust the next abstract state according to $x_2^+ \in h_2^{\mathrm{ASR}}(x_2, u_2, x_1^+) = F_2(x_2, u_2) \cap R(x_1^+)$, at the next time step. In contrast, PSR adopts a *predictive* architecture: it stores the *next* abstract state x_2^+, determined *before* applying the current input u_1 to the concrete system. This difference is reflected in the delay blocks of Fig. 7.

Consider the abstract input s_1, s_2, s_3 available at q_2. During offline abstract synthesis under ASR (or PSR), s_2 is discarded because it allows a self-loop on q_2, leading to a worst-case abstract cost of 3. However, PSR enables an online adaptation: the function $h_2^{\mathrm{PSR}}(x_2, u_2, x_1, u_1)$ can identify the concrete subset of q_2 where the trajectory lies and predict the next abstract state accordingly. If x_1 belongs to the *favorable* (light-blue) subset of q_2, the same input s_2 deterministically drives the system directly to q_4, saving one transition (see the concrete trajectory $x_0 \to x_0' \to x_0''$). Hence, PSR preserves the same worst-case cost (3) but can *improve the realized cost* online through local policy updates—at the expense of higher controller complexity, since h_2^{PSR} must store the subdivision of q_2 and its associated transitions.

4.3. Feedforward abstract control

For FAR, the next abstract state x_2^+ is determined independently of the concrete variables, including the concrete state x_1, in contrast with the predictive architecture of PSR (see Fig. 7). As a result, the abstract trajectory can be fully designed offline and tracked online. This construction, however, typically requires local incremental stabilizability to guarantee that the image of each abstract cell lies entirely within a unique successor.

In the running example, input s_3 deterministically maps all states in q_2 into q_3, yielding a deterministic abstraction. The corresponding concretized controller $\mathcal{C}_1^{\mathrm{FAR}}$ simply tracks the predefined abstract sequence $q_0 \to q_2 \to q_3 \to q_4$ using a stabilizing feedback. The cost remains exactly 3, as the online abstract trajectory coincides with the offline one—illustrating the purely *feedforward* nature of FAR.

Finally, most abstraction-based control frameworks in the literature rely either on a deterministic abstract system $\mathcal{S}_2$ (Asselborn et al., 2013; Calbert, Egidio, & Jungers, 2024; Egidio et al., 2022; Girard et al., 2009; Kloetzer & Belta, 2008; Mouelhi et al., 2013; Wongpiromsarn et al., 2011), or on a deterministic relation R (Apaza-Perez et al., 2020; Calbert, Banse, Legat, & Jungers, 2024; Calbert et al., 2021; Hsu et al., 2018; Khaled & Zamani, 2019; Rungger & Zamani, 2016) corresponding to a partition of the state space without overlaps. According to Proposition 1, these two common settings align with simulation relations FAR and MCR, respectively.

5. Construction

This section formalizes the construction of abstractions corresponding to each simulation relation introduced earlier. Building on the qualitative discussion of Section 4, we now provide systematic procedures for deriving, from a continuous deterministic system, abstract models that satisfy the relations ASR, PSR, FAR, and MCR. To this end, we consider a class of deterministic systems equipped with a *growth-bound function*, and use a common state–space discretization to derive the corresponding abstract transition maps. This framework highlights how different relations impose distinct assumptions on the original dynamics and how these assumptions shape the resulting control architectures.

5.1. Stability preliminaries

A continuous function $\gamma : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if it is a $\mathcal{K}$ function and $\gamma(r) \to +\infty$ when $r \to +\infty$.

We introduce a relaxed version of Lyapunov function, referred to as the *growth-bound function*, to bound the distance between trajectories with different initial conditions under a specific control law. This is the discrete-time analogue of Girard et al. (2009, Definition 2.3).

Definition 18. A smooth function $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_0^+$ is a *growth-bound function* under control $\kappa : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^\nu \to \mathbb{R}^\nu$ for system $f : \mathbb{R}^n \times \mathbb{R}^\nu \to \mathbb{R}^n$ if there exist $\mathcal{K}_\infty$ functions $\underline{\alpha}, \overline{\alpha}$ and $\rho > 0$ such that $\forall x, y \in \mathbb{R}^n$, $u \in \mathbb{R}^\nu$

$$V(f(y, \kappa(y, x, u)), f(x, u)) \leq \rho V(y, x) \tag{30}$$

$$\underline{\alpha}(\|y - x\|_2) \leq V(y, x) \leq \overline{\alpha}(\|y - x\|_2). \tag{31}$$

We define the ℓ-sublevel set of $V(\cdot, x)$ as

$$\mathcal{S}(x, \ell) = \{x' \in \mathbb{R}^n \mid V(x', x) \leq \ell\}. \tag{32}$$

When $\rho < 1$, the system is said to be *incrementally stabilizable* meaning that, under a suitable feedback, all trajectories converge to a common reference trajectory regardless of their initial conditions.

5.2. System definition and grid discretization

We study a deterministic simple system $\mathcal{S}_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1)$, where $\mathcal{X}_1 \subseteq \mathbb{R}^n$ is a compact set, $\mathcal{U}_1 \subseteq \mathbb{R}^\nu$, and the transition map $F_1(x, u) = \{f(x, u)\}$ is defined by a continuous nonlinear function $f : \mathcal{X}_1 \times \mathcal{U}_1 \to \mathcal{X}_1$. We impose stabilizability assumptions on f by assuming the existence of a growth-bound function V under control κ (Definition 18) with a contraction factor ρ. We will make the following supplementary assumption on the function V: there exists a $\mathcal{K}_\infty$ function γ such that

$$\forall x, y, z \in \mathbb{R}^n : \ V(x, y) - V(x, z) \leq \gamma(\|y - z\|_2). \tag{33}$$

While this assumption may seem quite strong, it is not restrictive provided we are interested in the dynamics of the system on a compact subset of the state space $\mathbb{R}^n$, as explained in Girard et al. (2009, Section IV.B). We illustrate the constructions on the following example.

Example 1. We consider a two-dimensional deterministic simple system $\mathcal{S}_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1)$ ($\mathcal{X}_1 \subseteq \mathbb{R}^2$, $F_1(x_1, u_1) = \{f(x_1, u_1)\}$) with growth-bound function $V(x, y) = \|x - y\|_2$, where $\underline{\alpha}(x) = \overline{\alpha}(x) = \gamma(x) = x = \underline{\alpha}^{-1}(x) = \overline{\alpha}^{-1}(x) = \gamma^{-1}(x)$, under control κ for f and contraction factor ρ. The function V satisfies (33), which reduces to the triangle inequality of the 2-norm. △

The abstraction constructions rely on discretizing the state space $\mathcal{X}_1$ into overlapping cells aligned on a grid

$$[\eta\mathbb{Z}^n] = \{c \in \mathbb{R}^n \mid \exists_{k \in \mathbb{Z}^n} \forall_{i \in [1;n]} \ c_i = k_i \eta\} \tag{34}$$

with *grid parameter* $\eta > 0$. The abstract states correspond to the vertices of this lattice, i.e.,

$$\mathcal{X}_2 = [\eta\mathbb{Z}^n] \cap \mathcal{X}_1. \tag{35}$$

Given $\epsilon > 0$, each abstract state $x_2 \in \mathcal{X}_2$ is associated with the ϵ-sublevel set of the function $V(\cdot, x_2)$, such that

$$(x_1, x_2) \in R \ \Leftrightarrow \ V(x_1, x_2) \leq \epsilon \ \Leftrightarrow \ R^{-1}(x_2) = \mathcal{S}(x_2, \epsilon). \tag{36}$$

To ensure that the relation is strict, i.e., $R(x_1) \neq \varnothing$ for all $x_1 \in \mathcal{X}_1$, we impose the condition $\eta \leq \frac{2}{\sqrt{n}} \overline{\alpha}^{-1}(\epsilon)$, under which Proposition 7 guarantees the existence of some $x_2 \in \mathcal{X}_2$ such that $x_1 \in \mathcal{S}(x_2, \epsilon)$.

Given $x_2 \in \mathcal{X}_2$ and $u_2 \in \mathcal{U}_2$, the growth assumption (30) allows to determine an over-approximation of the attainable set of a cell $R^{-1}(x_2)$ under the control law $g_{x_2,u_2}(x) = f(x, \kappa(x, x_2, u_2))$ (see Proposition 8)

$$g_{x_2,u_2}(R^{-1}(x_2)) \subseteq \mathcal{S}(f(x_2, u_2), \rho\epsilon). \tag{37}$$

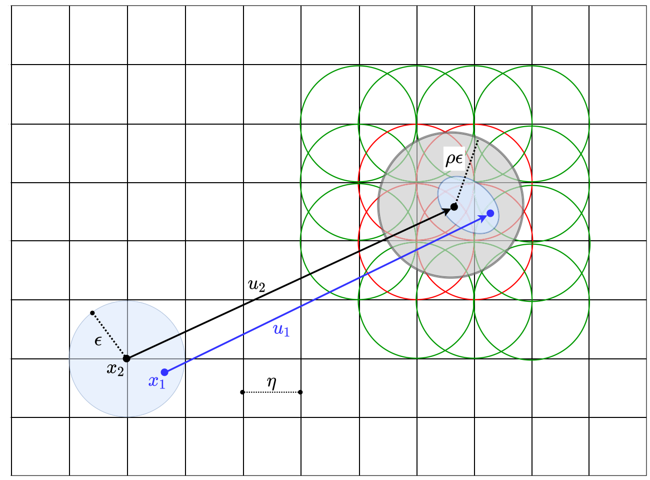


Fig. 10. Illustration of $(x_2, u_2, x_1, u_1) \in R_e^{\mathrm{ASR}}$ and R_e^{MCR} for Example 1, for some $\rho > 1$. The cell $R^{-1}(x_2) = \mathcal{S}(x_2, \epsilon)$ and its image under the dynamic F_1 are represented in blue. The gray circle $\mathcal{A} := \mathcal{S}(f(x_2, u_2), \rho\epsilon)$ is the over-approximation of the reachable set of $R^{-1}(x_2)$. The set $F_2^{\mathrm{ASR}}(x_2, u_2)$ just needs to contain enough cells so that their union covers the set $\mathcal{A}$ (38), for example, only the red cells. While the set $F_2^{\mathrm{MCR}}(x_2, u_2)$ must at least contain all the cells intersecting $\mathcal{A}$ (39), which are shown in red and green. Therefore, in this example, we have $|F_2^{\mathrm{ASR}}(x_2, u_2)| = 4$ while $|F_2^{\mathrm{MCR}}(x_2, u_2)| = 15$.

5.3. Abstraction construction per simulation relation

To facilitate comparison across the different simulation relations, all abstractions are constructed over a common state space $\mathcal{X}_2$ (35), a common input space $\mathcal{U}_2 \subseteq \mathcal{U}_1$, and the same relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ (36). Hence, their differences arise solely from the definition of the transition map F_2^{T} deigned so that the abstract system $\mathcal{S}_2^{\mathrm{T}} = (\mathcal{X}_2, \mathcal{U}_2, F_2^{\mathrm{T}})$ satisfies $\mathcal{S}_1 \preceq_R^{\mathrm{T}} \mathcal{S}_2^{\mathrm{T}}$.

Proposition 4. *The system $\mathcal{S}_2^{\mathrm{ASR}} = (\mathcal{X}_2, \mathcal{U}_2, F_2^{\mathrm{ASR}})$ with F_2^{ASR} satisfying $\forall x_2 \in \mathcal{X}_2,\ u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$:*

$$R^{-1}(F_2^{\mathrm{ASR}}(x_2, u_2)) \supseteq \mathcal{S}(f(x_2, u_2), \rho\epsilon), \tag{38}$$

ensures $\mathcal{S}_1 \preceq_R^{\mathrm{ASR}} \mathcal{S}_2^{\mathrm{ASR}}$. Similarly, the system $\mathcal{S}_2^{\mathrm{MCR}} = (\mathcal{X}_2, \mathcal{U}_2, F_2^{\mathrm{MCR}})$ with F_2^{MCR} satisfying $\forall x_2 \in \mathcal{X}_2,\ u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$:

$$F_2^{\mathrm{MCR}}(x_2, u_2) \supseteq R(\mathcal{S}(f(x_2, u_2), \rho\epsilon)), \tag{39}$$

ensures $\mathcal{S}_1 \preceq_R^{\mathrm{MCR}} \mathcal{S}_2^{\mathrm{MCR}}$. Additionally, one can use the specific implementations $h_1^{\mathrm{ASR}}(x_2, u_2, x_1) = h_1^{\mathrm{MCR}}(x_2, u_2, x_1) = \{\kappa(x_1, x_2, u_2)\}$.

Proof. Let $(x_1, x_2) \in R$ and $u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$, then for all $x_1^+ \in F_1(x_1, u_1)$ with $u_1 = \kappa(x_1, x_2, u_2)$, we have $x_1^+ \in \mathcal{S}(f(x_2, u_2), \rho\epsilon)$ by (37).

- Condition (38) involves $x_1^+ \in \mathcal{S}(f(x_2, u_2), \rho\epsilon) \subseteq R^{-1}(F_2^{\mathrm{ASR}}(x_2, u_2))$, which implies that $R(x_1^+) \cap F_2^{\mathrm{ASR}}(x_2, u_2) \neq \varnothing$. This establishes (17) and that $\kappa(x_1, x_2, u_2) \in I_R^{\mathrm{ASR}}(x_2, u_2, x_1)$.
- Condition (39) involves $R(x_1^+) \subseteq F_2(x_2, u_2)$, which establishes (22) and that $\kappa(x_1, x_2, u_2) \in I_R^{\mathrm{MCR}}(x_2, u_2, x_1)$. ■

While MCR yields a simpler concretization than ASR (Section 4.1), it increases abstraction non-determinism: $F_2^{\mathrm{ASR}}(x_2, u_2) \subseteq F_2^{\mathrm{MCR}}(x_2, u_2)$ (Fig. 10). The larger branching factor can make the abstract control problem infeasible for $\mathcal{S}_2^{\mathrm{MCR}}$ even when feasible for $\mathcal{S}_2^{\mathrm{ASR}}$.

Proposition 5. *Given the additional assumption that $\rho < 1$ and $\eta \leq \frac{2}{\sqrt{n}} \min\left(\overline{\alpha}^{-1}(\epsilon), \gamma^{-1}((1 - \rho)\epsilon)\right)$, the system $\mathcal{S}_2^{\mathrm{FAR}} =$*

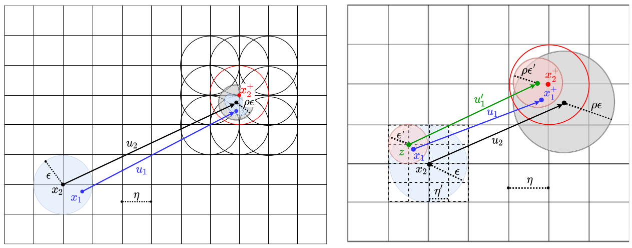


Fig. 11. Illustration of $(x_2, u_2, x_1, u_1, x_2^+) \in R_e^{\mathrm{FAR}}$ (left) and $(x_2, u_2, x_1, u_1) \in R_e^{\mathrm{PSR}}$ (right) for Example 1.

$(\mathcal{X}_2, \mathcal{U}_2, F_2^{\mathrm{FAR}})$ *with* F_2^{FAR} *satisfying* $\forall x_2 \in \mathcal{X}_2,\ u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$:

$$F_2^{\mathrm{FAR}}(x_2, u_2) \supseteq \operatorname*{argmin}_{x_2' \in \mathcal{X}_2} \|f(x_2, u_2) - x_2'\|_\infty, \tag{40}$$

ensures $\mathcal{S}_1 \preceq_R^{\mathrm{FAR}} \mathcal{S}_2^{\mathrm{FAR}}$. *Additionally, one can use the specific implementations* $h_1^{\mathrm{FAR}}(x_2, u_2, x_1, x_2^+) = \{\kappa(x_1, x_2, u_2)\}$ *and* $h_2^{\mathrm{FAR}}(x_2, u_2) = \operatorname{argmin}_{x_2' \in \mathcal{X}_2} \|f(x_2, u_2) - x_2'\|_\infty$.

Proof. Applying Proposition 9 with parameters $(\eta, \epsilon, \epsilon)$ and $\rho < 1$, the condition on η ensures that $\forall x_2 \in \mathcal{X}_2,\ \forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$: $g_{x_2,u_2}(R^{-1}(x_2)) \subseteq R^{-1}(x_2^+)$ with $x_2^+ = \operatorname{argmin}_{x_2' \in \mathcal{X}_2} \|f(x_2, u_2) - x_2'\|_\infty$. Therefore, by (40), $x_2^+ \in F_2^{\mathrm{FAR}}(x_2, u_2)$ and for all $x_1^+ \in F_1(x_1, u_1)$ with $u_1 = \kappa(x_1, x_2, u_2)$, we have $(x_1^+, x_2^+) \in R$. This establishes (21), $\kappa(x_1, x_2, u_2) \in I_R^{\mathrm{FAR}}(x_2, u_2, x_1, x_2^+)$ and h_2^{FAR} implements Definition 16. ■

The choice of $F_2^{\mathrm{FAR}}(x_2, u_2) = h_2^{\mathrm{FAR}}(x_2, u_2)$ results in a deterministic abstraction as illustrated in Fig. 11 (left). In this context, constructing $\mathcal{S}_2^{\mathrm{FAR}}$ involves low computational complexity. Designing an abstract transition only requires computing the trajectory of a single point $f(x_2, u_2)$ and identifying the closest grid point. However, unlike other approaches, this method requires a strong assumption on the system's stability, specifically incremental stabilizability ($\rho < 1$). In Calbert, Banse, Legat, and Jungers (2024, Section 6), the authors construct a feedforward abstraction relation $\mathcal{S}_2^{\mathrm{FAR}}$ for an incrementally stabilizable dynamical system, specifically a DC–DC converter defined in Rungger and Zamani (2016, Section 4.2), and illustrate the speed-up in the construction of the abstraction resulting from this efficient implementation of h_2^{FAR}.

We now construct a finer grid $[\eta'\mathbb{Z}^n]$ with $0 < \eta' < \eta$; see Fig. 11 (right). Given $\epsilon' > 0$, we choose η' sufficiently small, satisfying $\eta' \le \frac{2}{\sqrt{n}}\,\overline{\alpha}^{-1}(\epsilon')$, so that every state $x_1 \in \mathcal{X}_1$ lies within an ϵ'-sublevel set $\mathcal{S}(z, \epsilon')$ of V, centered at some vertex $z \in [\eta'\mathbb{Z}^n]$ of the refined lattice (see Proposition 7). Given $x_2 \in \mathcal{X}_2$, we denote $\mathcal{Z}_{\eta,\eta',\epsilon,\epsilon'}(x_2) \subseteq [\eta'\mathbb{Z}^n]$ a set of minimal cardinality s.t.

$$R^{-1}(x_2) \subseteq \bigcup_{z \in \mathcal{Z}_{\eta,\eta',\epsilon,\epsilon'}(x_2)} \mathcal{S}(z, \epsilon'). \tag{41}$$

Proposition 6. *Let* $\mathcal{Z}_{\eta,\eta',\epsilon,\epsilon'}(x_2)$ *defined in* (41) *with parameters satisfying*

$$\begin{cases} \eta' \le \frac{2}{\sqrt{n}}\overline{\alpha}^{-1}(\epsilon'), & (42)\\ \rho\epsilon' < \epsilon, & (43)\\ \eta \le \frac{2}{\sqrt{n}}\min\left(\overline{\alpha}^{-1}(\epsilon), \gamma^{-1}(\epsilon - \rho\epsilon')\right). & (44) \end{cases}$$

The system $\mathcal{S}_2^{\mathrm{PSR}} = (\mathcal{X}_2, \mathcal{U}_2, F_2^{\mathrm{PSR}})$ *with* F_2^{PSR} *satisfying* $\forall x_2 \in \mathcal{X}_2,\ u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$:

$$F_2^{\mathrm{PSR}}(x_2, u_2) \supseteq \bigcup_{z \in \mathcal{Z}_{\eta,\eta',\epsilon,\epsilon'}(x_2)} \operatorname*{argmin}_{x_2' \in \mathcal{X}_2} \|f(z, \kappa(z, x_2, u_2)) - x_2'\|_\infty,$$

ensures $\mathcal{S}_1 \preceq_R^{\mathrm{PSR}} \mathcal{S}_2^{\mathrm{PSR}}$. *Additionally, one can use the specific implementations*

$$h_1^{\mathrm{PSR}}(x_2, u_2, x_1) = \{\kappa(x_1, z, u_1')\}$$
$$h_2^{\mathrm{PSR}}(x_2, u_2, x_1, u_1) = \operatorname*{argmin}_{x_2' \in \mathcal{X}_2} \|f(z, u_1') - x_2'\|_\infty,$$

where $z = \operatorname{argmin}_{z' \in \mathcal{Z}_{\eta,\eta',\epsilon,\epsilon'}(x_2)} \|x_1 - z'\|_\infty$ *and* $u_1' = \kappa(z, x_2, u_2)$.

Proof. Let $(x_1, x_2) \in R$ and $u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$. By applying Proposition 7 with parameters (η', ϵ') satisfying (42), we have $x_1 \in \mathcal{S}(z, \epsilon')$ with $z = \operatorname{argmin}_{z' \in \mathcal{Z}_{\eta,\eta',\epsilon,\epsilon'}(x_2)} \|x_1 - z'\|_\infty$. Next, applying Proposition 9 with parameters $(\eta, \epsilon', \epsilon)$ satisfying (43) and (44), we ensure that $x_1^+ = f(x_1, u_1) \in \mathcal{S}(x_2^+, \epsilon)$ with $u_1 = \kappa(x_1, z, u_1')$, $x_2^+ = \operatorname{argmin}_{x_2' \in \mathcal{X}_2} \|f(z, u_1') - x_2'\|_\infty$, and $u_1' = \kappa(z, x_2, u_2)$. Thus, $x_1^+ \in R^{-1}(x_2^+)$ and by definition of F_2^{PSR}, $x_2^+ \in F_2^{\mathrm{PSR}}(x_2, u_2)$, which establishes (20). Thus, $u_1 \in I_R^{\mathrm{PSR}}(x_2, u_2, x_1)$ and h_2^{PSR} satisfies Definition 16. ■

As discussed in Section 4.2, it is possible to predict the next abstract state x_2^+ using $h_2^{\mathrm{PSR}}(x_2, u_2, x_1, u_1)$ before applying the concrete input u_1 given by $h_1^{\mathrm{PSR}}(x_2, u_2, x_1)$. Unlike FAR, the PSR relation can be constructed without requiring the incremental stabilizability ($\rho < 1$).

6. Conclusion

Abstraction-based control relies on two core elements: the *relation*, ensuring consistent transitions between concrete and abstract states, and the *concretization procedure*, which derives a valid controller for the original system from the abstract one.

In this article, we proposed a systematic approach to characterize simulation relations that refine the ASR, through a concretization procedure featuring a plug-and-play control architecture. To this end, we introduce a universal system transformation, called the *interfaced system*, which is universal in the sense that it is always in FRR with the abstract system. This allowed us to leverage the necessary and sufficient characterization of FRR in terms of concretization procedure (Theorem 2) to a broader class of simulation relations. Specifically, we demonstrated (Corollary 1) that if the concrete system is related to the abstraction via a simulation relation, then the abstract controller can be seamlessly connected to the original system through the corresponding interface, irrespective of the specific specification to be enforced, and vice versa. This generalizes previous results and provides a unifying framework for systematically characterizing simulation relations. Finally, we discussed the pros and cons of different relations using a common example.

Our future research will focus on three main directions. First, while we have illustrated our approach using five key relations relevant for the design of smart abstractions, we plan to develop a systematic procedure to identify the interface characterizing any refined notion of the alternating simulation relation. Second, while the objective of this work is to highlight the benefits of simpler or more structured control architectures enabled by conservative simulation relations refining ASR, we aim to investigate which classes of systems allow these relations to be practically constructed. Section 5 provides a first step in this direction by showing that relations such as ASR, MCR, and PSR can be built for arbitrary systems, whereas FAR additionally requires incremental stabilizability. Third, we aim to extend our current framework, which focuses on state-feedback control of simple systems, to *output feedback* control of systems with output maps. Various types of relations for such systems are well-established in the literature (Apaza-Perez et al., 2020; Coppola et al., 2022; Majumdar et al., 2020; Schmuck & Raisch, 2014), and we plan to classify and characterize them in terms of their associated control architectures.

Acknowledgments

JC is a FRIA Research Fellow. RJ is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 864017 - L2C, from the Horizon Europe programme under grant agreement No101177842 - Unimaas, and from the ARC (French Community of Belgium)-SIDDARTA.

Appendix A. Proof of Theorem 4

We start by proving (1). From the definition of I_R^{T} in (23), it follows that the tuple $(\mathcal{Z}_1, h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, \widetilde{R})$ in Definition 16 is an interface from $\mathcal{S}_2$ to $\mathcal{S}_1$. Finally, we show that $\widetilde{R}$ in (29) satisfies Definition 17. Let x_1, x_1' be such that $\widetilde{R}((x_1, x_2)) \neq \varnothing$ and $\widetilde{R}((x_1', x_2)) \neq \varnothing$. Then we have $\widetilde{R}((x_1, x_2)) \cap \widetilde{R}((x_1', x_2)) = \{x_2\} \neq \varnothing$. We now turn to proving (2). By Theorem 3, we have $\widetilde{\mathcal{S}}_1^{\mathrm{T}} \preceq_{\widetilde{R}}^{\mathrm{FRR}} \mathcal{S}_2$. From Definition 8, it follows that for any $((x_1, z_1), x_2) \in \widetilde{R}$

(i) $\mathcal{U}_{\mathcal{S}_2}(x_2) \subseteq \mathcal{U}_{\widetilde{\mathcal{S}}_1^{\mathrm{T}}}((x_1, z_1))$;

(ii) $\forall u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)\ \forall (x_1^+, z_1^+) \in \widetilde{F}_1((x_1, z_1), u_2) :\ \widetilde{R}((x_1^+, z_1^+)) \neq \varnothing$ and $\widetilde{R}((x_1^+, z_1^+)) \subseteq F_2(x_2, u_2)$.

Let $(x_1, x_2) \in R$ and $u_2 \in \mathcal{U}_{\mathcal{S}_2}(x_2)$. This implies the existence of z_1 such that $((x_1, z_1), x_2) \in \widetilde{R}$. By condition (i), we have $u_2 \in \mathcal{U}_{\widetilde{\mathcal{S}}_1}((x_1, z_1))$, meaning $\widetilde{F}_1((x_1, z_1), u_2) \neq \varnothing$. Thus, by Definition 13, for any $u_1 \in h_1^{\mathrm{T}}(v_1)$, $u_1 \in \mathcal{U}_{\mathcal{S}_1}(x_1)$, and for all $x_1^+ \in F_1(x_1, u_1)$, $z_1^+ \in h_2^{\mathrm{T}}(v_2)$, it holds that $(x_1^+, z_1^+) \in \widetilde{F}_1((x_1, z_1), u_2)$. In addition, by (ii), $\widetilde{R}((x_1^+, z_1^+)) \neq \varnothing$ and $\widetilde{R}((x_1^+, z_1^+)) \subseteq F_2(x_2, u_2)$.
The rest of the proof is specific to each relation type:

- ASR: According to Lemma 2, $\widetilde{R}((x_1^+, z_1^+)) \subseteq R(x_1^+)$. Therefore, $F_2(x_2, u_2) \cap R(x_1^+) \supseteq \widetilde{R}((x_1^+, z_1^+)) \neq \varnothing$, which is equivalent to (17).
- PSR: We consider $u_1 \in h_1^{\mathrm{PSR}}(z_1, u_2, x_1)$ and $z_1^+ \in h_2^{\mathrm{PSR}}(z_1, u_2, u_1, x_1)$. Since $\widetilde{R}$ satisfies Definition 17, it holds that
$$\mathcal{A} := \left(\bigcap_{x_1^+ \in F_1(x_1, u_1)} \widetilde{R}((x_1^+, z_1^+)) \right) \neq \varnothing,$$
as $\widetilde{R}((x_1^+, z_1^+)) \neq \varnothing$ by (ii). Let $x_2^+ \in \mathcal{A}$. Firstly, $x_2^+ \in F_2(x_2, u_2)$ since $\widetilde{R}((x_1^+, z_1^+)) \subseteq F_2(x_2, u_2)$ by (ii). Secondly, according to Lemma 2, $\widetilde{R}((x_1^+, z_1^+)) \subseteq R(x_1^+)$, therefore $x_2^+ \in R(x_1^+)$, which is equivalent to (20).
- FAR: Let $z_1^+ \in h_2^{\mathrm{FAR}}(z_1, u_2)$. Since $\widetilde{R}$ satisfies Definition 17, it holds that
$$\mathcal{A} := \bigcap_{\substack{u_1 \in h_1(z_1, u_2, x_1, z_1^+) \\ x_1^+ \in F_1(x_1, u_1)}} \widetilde{R}((x_1^+, z_1^+)),$$
as $\widetilde{R}((x_1^+, z_1^+)) \neq \varnothing$ by (ii). Let $x_2^+ \in \mathcal{A}$. Firstly, $x_2^+ \in F_2(x_2, u_2)$ since $\widetilde{R}((x_1^+, z_1^+)) \subseteq F_2(x_2, u_2)$ by (ii). Secondly, according to Lemma 2, $\widetilde{R}((x_1^+, z_1^+)) \subseteq R(x_1^+)$, therefore $x_2^+ \in R(x_1^+)$, which is equivalent to (21).
- MCR: By definition of R in (27), we have
$$R(x_1^+) = \bigcup_{z_1^+ \in h_2^{\mathrm{MCR}}(x_1^+)} \widetilde{R}((x_1^+, z_1^+)),$$
which establishes that $R(x_1^+) \neq \varnothing$ and $R(x_1^+) \subseteq F_2(x_2, u_2)$, which is equivalent to (22). ■

Appendix B. Useful results

Lemma 1. *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ in (12), and $(\mathcal{Z}_1, h_1, h_2, \widetilde{R})$ an interface from $\mathcal{S}_2$ to $\mathcal{S}_1$ with associated interfaced system $\widetilde{\mathcal{S}}_1$ (Definition 13). Then, for any controller $\mathcal{C}_2$ f.c. with $\mathcal{S}_2$, the concretized controller $\mathcal{C}_1$ (Definition 14) and $\widetilde{\mathcal{C}}_1 = \mathcal{C}_2 \circ \widetilde{R}$ satisfy $\boldsymbol{x}_1 \in \mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1) \Leftrightarrow \widetilde{\boldsymbol{x}}_1 \in \mathcal{B}_\times(\widetilde{\mathcal{C}}_1 \times \widetilde{\mathcal{S}}_1)$, where $\boldsymbol{x}_1 = \pi_{\mathcal{X}_1}(\widetilde{\boldsymbol{x}}_1)$ (1).*

Proof. By Definition 4, $\widetilde{\mathcal{C}}_1 = \mathcal{C}_2 \circ \widetilde{R} = (\mathcal{X}_{\mathcal{C}_2}, \mathcal{X}_1 \times \mathcal{Z}_1, \mathcal{V}_{\mathcal{C}_2}, \mathcal{U}_2, F_{\mathcal{C}_2}, H_{\widetilde{\mathcal{C}}_1})$ where $H_{\widetilde{\mathcal{C}}_1}(x_{\mathcal{C}_2}, (x_1, z_1)) = H_{\mathcal{C}_2}(x_{\mathcal{C}_2}, \widetilde{R}(x_1, z_1))$, since $\widetilde{R}$ is a static system. Thus, $\widetilde{\boldsymbol{x}}_1 \in \mathcal{B}_\times(\widetilde{\mathcal{C}}_1 \times \widetilde{\mathcal{S}}_1)$ if and only if there exist sequences $\boldsymbol{u}_1, \boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{x}_{\mathcal{C}_2}, \boldsymbol{v}_{\mathcal{C}_2}$ satisfying (26) with $\boldsymbol{x}_1 = \pi_{\mathcal{X}_1}(\widetilde{\boldsymbol{x}}_1)$ and $\boldsymbol{z}_1 = \pi_{\mathcal{Z}_1}(\widetilde{\boldsymbol{x}}_1)$. This establishes that $\boldsymbol{x}_1 \in \mathcal{B}_\times(\mathcal{C}_1 \times \mathcal{S}_1)$. ■

Lemma 2. *Let $\widetilde{R} \subseteq (\mathcal{X}_1 \times \mathcal{Z}_1) \times \mathcal{X}_2$, and $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ in (27). If $\widetilde{R}((x_1, z_1)) \neq \varnothing$, then $\widetilde{R}((x_1, z_1)) \subseteq R(x_1)$.*

Proof. By definition,
$$R(x_1) = \cup_{(x_1, z_1) | \widetilde{R}((x_1, z_1)) \neq \varnothing} \widetilde{R}((x_1, z_1)). \quad ■$$

Proposition 7. *Let V be a growth-bound function as in Definition 18 and parameters $(\eta, \epsilon) > 0$. If $\eta \leq \frac{2}{\sqrt{n}} \overline{\alpha}^{-1}(\epsilon)$, then for all $x \in \mathbb{R}^n$, $x \in \mathcal{S}(x_2, \epsilon)$ with $x_2 = \mathrm{argmin}_{c \in [\eta \mathbb{Z}^n]} \|x - c\|_\infty$.*

Proof. Let $x \in \mathbb{R}^n$ and $x_2 = \mathrm{argmin}_{c \in [\eta \mathbb{Z}^n]} \|x - c\|_\infty$. Then $\|x - x_2\|_\infty \leq \frac{\eta}{2}$, and we have $V(x, x_2) \leq \overline{\alpha}(\|x - x_2\|_2) \leq \overline{\alpha}(\sqrt{n}\|x - x_2\|_\infty) \leq \overline{\alpha}\left(\sqrt{n}\frac{\eta}{2}\right) \leq \epsilon$, because $\overline{\alpha}$ is a $\mathcal{K}$ function. ■

Proposition 8. *Let V be a growth-bound function as in Definition 18. Given $x_2 \in \mathbb{R}^n$, $u_2 \in \mathbb{R}^\nu$, and $\epsilon > 0$. Then, the following holds $g_{x_2, u_2}(\mathcal{S}(x_2, \epsilon)) \subseteq \mathcal{S}(f(x_2, u_2), \rho\epsilon)$, where $g_{x_2, u_2}(x) = f(x, \kappa(x, x_2, u_2))$.*

Proof. Let $x \in \mathcal{S}(x_2, \epsilon)$. By (30), it holds
$$V(f(x, \kappa(x, x_2, u_2)), f(x_2, u_2)) \leq \rho V(x, x_2) \leq \rho\epsilon,$$
which involves $g_{x_2, u_2}(x) \in \mathcal{S}(f(x_2, u_2), \rho\epsilon)$. ■

Proposition 9 (Girard et al. (2009, Eq. (9))). *Let V be a growth-bound function as in Definition 18 satisfying (33) and parameters $(\eta, \epsilon, \epsilon') > 0$. If $\epsilon' - \rho\epsilon > 0$, then the choice $\eta \leq \frac{2}{\sqrt{n}} \gamma^{-1}(\epsilon' - \rho\epsilon)$ ensures that*
$$\forall x_2 \in \mathbb{R}^n\ \forall u_2 \in \mathbb{R}^\nu :\ g_{x_2, u_2}(\mathcal{S}(x_2, \epsilon)) \subseteq \mathcal{S}(x_2', \epsilon'),$$
with $x_2' = \mathrm{argmin}_{x' \in [\eta \mathbb{Z}^n]} \|x' - f(x_2, u_2)\|_\infty$ and $g_{x_2, u_2}(x) = f(x, \kappa(x, x_2, u_2))$. Special case: when $V(x, y) = \|x - y\|_2$ and $\epsilon = \epsilon'$, the condition on η reduces to $\eta \leq \frac{2}{\sqrt{n}}\epsilon(1 - \rho)$.

Proof. Let $x_1 \in \mathcal{S}(x_2, \epsilon)$, $x_1^+ = f(x_1, \kappa(x_1, x_2, u_2))$, $x_2^+ = f(x_2, u_2)$ and $x_2' = \mathrm{argmin}_{x' \in [\eta \mathbb{Z}^n]} \|x' - x_2^+\|_\infty$. Then, $\|x_2' - x_2^+\|_\infty \leq \frac{\eta}{2}$. Using (30) and (33), and since γ is a $\mathcal{K}_\infty$ function, we have
$$\begin{aligned} V(x_1^+, x_2') &\leq V(x_1^+, x_2^+) + \gamma(\|x_2' - x_2^+\|_2) \\ &\leq \rho V(x_1, x_2) + \gamma(\sqrt{n}\|x_2' - x_2^+\|_\infty) \\ &\leq \rho\epsilon + \gamma\left(\sqrt{n}\tfrac{\eta}{2}\right) \\ &\leq \epsilon' \end{aligned}$$
which implies that $x_1^+ \in \mathcal{S}(x_2', \epsilon')$. ■

References

Alur, R. (2015). *Principles of cyber-physical systems*. MIT Press.

Alur, R., Henzinger, T. A., Kupferman, O., & Vardi, M. Y. (1998). Alternating refinement relations. In *CONCUR'98 concurrency theory: 9th international conference nice, France, September 8–11, 1998 proceedings 9* (pp. 163–178). Springer.

Angeli, D. (2002). A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, *47*(3), 410–421.

Apaza-Perez, W. A., Girard, A., Combastel, C., & Zolghadri, A. (2020). Symbolic observer-based controller for uncertain nonlinear systems. *IEEE Control Systems Letters*, *5*(4), 1297–1302.

Asselborn, L., Gross, D., & Stursberg, O. (2013). Control of uncertain nonlinear systems using ellipsoidal reachability calculus. *IFAC Proceedings Volumes*, *46*(23), 50–55.

Baier, C., & Katoen, J.-P. (2008). *Principles of model checking*. MIT Press.

Belta, C., Yordanov, B., & Gol, E. A. (2017). *Formal methods for discrete-time dynamical systems*: *vol. 89*, Springer.

Borri, A., Pola, G., & Di Benedetto, M. D. (2018). Design of symbolic controllers for networked control systems. *IEEE Transactions on Automatic Control*, *64*(3), 1034–1046.

Calbert, J., Banse, A., Legat, B., & Jungers, R. M. (2024). Dionysos. jl: a modular platform for smart symbolic control. arXiv preprint arXiv:2404.14114.

Calbert, J., Egidio, L. N., & Jungers, R. M. (2024). Smart abstraction based on iterative cover and non-uniform cells. *IEEE Control Systems Letters*.

Calbert, J., Legat, B., Egidio, L. N., & Jungers, R. (2021). Alternating simulation on hierarchical abstractions. In *2021 60th IEEE conference on decision and control* (pp. 593–598). IEEE.

Calbert, J., Mattenet, S., Girard, A., & Jungers, R. (2024). Memoryless concretization relation. In *Proceedings of the 27th ACM international conference on hybrid systems: computation and control* (pp. 1–9).

Coppola, R., Peruffo, A., & Mazo, M. (2022). Data-driven abstractions for verification of deterministic systems. arXiv preprint arXiv:2211.01793.

Egidio, L. N., Lima, T. A., & Jungers, R. M. (2022). State-feedback abstractions for optimal control of piecewise-affine systems. In *2022 IEEE 61st conference on decision and control* (pp. 7455–7460). IEEE.

Girard, A., & Pappas, G. J. (2009). Hierarchical control system design using approximate simulation. *Automatica*, *45*(2), 566–571.

Girard, A., Pola, G., & Tabuada, P. (2009). Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control*, *55*(1), 116–126.

Hsu, K., Majumdar, R., Mallik, K., & Schmuck, A.-K. (2018). Multi-layered abstraction-based controller synthesis for continuous-time systems. In *Proceedings of the 21st international conference on hybrid systems: computation and control* (pp. 120–129).

Khaled, M., & Zamani, M. (2019). pFaces: An acceleration ecosystem for symbolic control. In *Proceedings of the 22nd ACM international conference on hybrid systems: computation and control* (pp. 252–257).

Kim, K.-D., & Kumar, P. R. (2012). Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, *100*(Special Centennial Issue), 1287–1308.

Kloetzer, M., & Belta, C. (2008). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, *53*(1), 287–297.

Kupferman, O., & Vardi, M. Y. (2001). Model checking of safety properties. *Formal Methods in System Design*, *19*, 291–314.

Liu, J. (2017). Robust abstractions for control synthesis: Completeness via robustness for linear-time properties. In *Proceedings of the 20th international conference on hybrid systems: computation and control* (pp. 101–110).

Lohmiller, W., & Slotine, J.-J. E. (1998). On contraction analysis for non-linear systems. *Automatica*, *34*(6), 683–696.

Majumdar, R., Ozay, N., & Schmuck, A.-K. (2020). On abstraction-based controller design with output feedback. In *Proceedings of the 23rd international conference on hybrid systems: computation and control* (pp. 1–11).

Mouelhi, S., Girard, A., & Gössler, G. (2013). Cosyma: a tool for controller synthesis using multi-scale abstractions. In *Proceedings of the 16th international conference on hybrid systems: computation and control* (pp. 83–88).

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

Rungger, M., & Zamani, M. (2016). SCOTS: A tool for the synthesis of symbolic controllers. In *Proceedings of the 19th international conference on hybrid systems: computation and control* (pp. 99–104).

Schmuck, A.-K., & Raisch, J. (2014). Asynchronous l-complete approximations. *Systems & Control Letters*, *73*, 67–75.

Tabuada, P. (2009). *Verification and control of hybrid systems: A symbolic approach*. Springer Science & Business Media.

Vidyasagar, M. (1981). *Input-output analysis of large-scale interconnected systems: decomposition, well-posedness and stability*. Springer.

Wongpiromsarn, T., Topcu, U., Ozay, N., Xu, H., & Murray, R. M. (2011). Tulip: a software toolbox for receding horizon temporal logic planning. In *Proceedings of the 14th international conference on hybrid systems: computation and control* (pp. 313–314).

Julien Calbert received the B.S. degree in computer engineering from UCLouvain, Belgium, in 2018, and the M.Eng. degree in applied mathematics from UCLouvain in 2020. He obtained the Ph.D. degree in applied mathematics from UCLouvain in 2024, as a FRIA fellow (F.R.S.-FNRS), within the ICTEAM Institute under the supervision of Prof. Raphaël M. Jungers. His research interests include verification and control of dynamical systems, control of hybrid systems, optimal control, and symbolic control.

Antoine Girard is a Senior Researcher at CNRS and a member of the Laboratory of Signals and Systems. He is also an Adjunct Professor at CentraleSupélec, Université Paris-Saclay. He received the Ph.D. degree from Grenoble Institute of Technology, in 2004. From 2004 to 2006, he held postdoctoral positions at University of Pennsylvania and Université Grenoble-Alpes. From 2006 to 2015, he was an Assistant/Associate Professor at the Université Grenoble-Alpes. His main research interests deal with analysis and control of hybrid systems with an emphasis on computational approaches, formal methods and applications to cyber–physical and autonomous systems. Antoine Girard is an IEEE Fellow. In 2015, he was appointed as a junior member of the Institut Universitaire de France (IUF). In 2016, he was awarded an ERC Consolidator Grant . He received the George S. Axelby Outstanding Paper Award from the IEEE Control Systems Society in 2009, the CNRS Bronze Medal in 2014, and the European Control Award in 2018.

Raphaël M. Jungers is a Professor at UCLouvain, Belgium. His main interests lie in the fields of Computer Science, Graph Theory, Optimization and Control. He received a Ph.D. in Mathematical Engineering from UCLouvain (2008), and a M.Sc. in Applied Mathematics, both from the Ecole Centrale Paris, (2004), and from UCLouvain (2005). He has held various invited positions, at the Université Libre de Bruxelles (2008–2009), at the Laboratory for Information and Decision Systems of the Massachusetts Institute of Technology (2009–2010), at the University of L'Aquila (2011, 2013, 2016), at the University of California Los Angeles (2016–2017), and at Oxford University (2022–2023). He has been serving as Vice President of IEEE CSS, in charge of Conference Activities (2026–2027). He is a FNRS, BAEF, and Fulbright fellow. He has been an Editor at large for the IEEE CDC, Associate Editor for the IEEE CSS Conference Editorial Board, and the journals NAHS (2015–2016), Systems and Control Letters (2016–2017), IEEE Transactions on Automatic Control (2015–2020), Automatica (2020–2025). He is currently serving as deputy Editor in Chief for NAHS. He was the recipient of the IBM Belgium 2009 award and a finalist of the ERCIM Cor Baayen award 2011. He was the co-recipient of the SICON best paper award 2013–2014, the HSCC2020 best paper award, the NAHS 2020–2022 best paper award, and he is the recipient of an ERC 2019 award. He is an IEEE Fellow (class 2025).