Robust Deterministic Abstractions for Supervising Discrete-Time Continuous Systems

Gwendal Priser, Elena Vanneaux, Goran Frehse

Robust Deterministic Abstractions for Supervising Discrete-time Continuous Systems

Gwendal Priser, Elena Vanneaux, and Goran Frehse[0000−0002−5441−0481]

U2IS, ENSTA Paris, Institut Polytechnique de Paris
{gwendal.priser|elena.vanneaux|goran.frehse}@ensta-paris.fr

Abstract. We present a method for constructing discrete abstractions for discrete-time, continuous-state systems. Related approaches construct a discrete bisimulation, which leaves little room for non-determinism in the outputs and quickly leads to highly complex models since all concrete behavior is covered. Our approach is to relax these requirements and build a satisficing solution: a discrete abstraction that is deterministic, robust, and as complete as possible under the given parameters. This allows us to balance granularity and computational feasibility. We leverage linearization and linear feedback control to extend the approach from globally contractive systems to systems with contractive cycles. The resulting abstraction directly induces a supervisor policy. The approach is illustrated with numerical experiments and has potential applications in various domains where system safety and reversibility are essential.

Keywords: continuous dynamical systems · discrete abstraction · robustness · supervision

1 Introduction

We are interested in supervising the behavior of dynamical systems that are subject to a sequence of control inputs, also called actions. The system's reaction to an input can be nondeterministic, i.e., the same input can lead to different successor states. The supervisor's job is to ensure that the control actions always lead to runs that satisfy a given specification. Ideally, we want to identify the maximally permissive supervisor that achieves this. However, this is known to be a hard problem, so the approach we pursue in this paper is to trade off some of the maximality to reduce the computational cost.

The design of controllers and supervisors often involves the construction of an abstraction, i.e., a substitute dynamical system that is simple enough to carry out the actual synthesis process [23,4]. This is usually a two-step process: First, designing an abstraction that captures the system response to all possible control actions in all states. Second, using the abstraction to identify the control actions that ensure a given specification. One problem with this two-step approach is that a complete abstraction may be extremely complex. It stands to reason that a simpler abstraction may be sufficient to find an acceptable supervisor, so we forego the objective of finding the maximal one.

Discrete abstractions of continuous systems are usually associated with a notion of contractiveness, like global asymptotic stability [24,11,20,12]. We propose to construct *deterministic* abstractions that use feedback control policies to induce contractiveness locally. Deterministic policy abstractions provide an extraordinarily strong link to the concrete system. One can pick any succession of policies from the abstraction, apply it to the concrete system and is guaranteed to obtain the same outputs – without having to adapt to the concrete state. If even the policies are independent of the concrete state, which we call a reach set abstraction, *any trace from a reach set abstraction can be enforced completely in an open loop*. This can be advantageous in critical situations, under degraded operating conditions, or when feedback can be compromised due to damage, malfunction, or communication problems. For example, open-loop control is preferred for the control surfaces of certain missiles [6].

Given the power of a deterministic abstraction, the question is under what conditions and with which tools they can be constructed. We show that deterministic abstractions exist under conditions similar to standard bisimulations. While finding the maximal policy involves robust backward reachability, the forward reachability operator suffices to obtain a solution, and we show maximality under certain conditions. To synthesize the local feedback policies, we use linearization. The linearized system allows us to describe the sufficient conditions for the abstraction with a single linear constraint system, which can then be fed to an off-the-shelf solver. If the linear constraint system is infeasible, we can use feedback from the LP solver to select a subset of transitions that is satisfiable yet preserves properties like connectedness.

Related Work The formal basis of our abstractions is provided by the well-established notions of simulation [17] and alternating simulation [2], which were applied to continuous control systems in [20]. Inspired by the approach in [9], we admit nondeterministic outputs and consider specifications that are upwards-closed with respect to sets of outputs. While simulation preserves safety properties, bisimulation preserves much stronger properties like those captured with temporal logics like CTL* [8]. However, bisimulation with both state and input labels requires an extremely close correspondence between the two systems so that abstractions that bisimulate the original system may be found only in a restricted class of systems; see, for example, [22].

We consider nondeterministic dynamical systems, which may end up in different states for the same input. For this situation, *alternating simulation* [2] provides a stronger relationship, which is particularly suited to control systems [20], and we build on this concept. An even stronger notion, called *feedback refinement relations* [21], requires that inputs in the concrete and abstract system are matched exactly; our policy abstractions require this implicitly in the sense that applying the policy to the concrete state must in all cases lead to states that are covered by the relation. Controllers synthesized with the approach in [21] can immediately be mapped to concrete control actions; in our case, this is achieved through policies, which, however, require continuous state information. In [20], the antagonistic choice of successor states is explicitly represented by disturbance

labels. In our approach, disturbances are implicitly represented by successor sets, which encode a nondeterministic choice. We formalize a novel abstraction-based approach in which the transitions are not labeled by discretized control inputs, but by local state-feedback controllers that can ensure the determinism of the symbolic system, even when the concrete system is non-deterministic and not incrementally stable [13]. This stateful policy differentiates our approach from much of the literature, where the supervisor is often based only on the state of the concrete system.

To make our abstraction deterministic, we use closed-loop strategies to ensure contractions. To compute a closed-loop strategy, we first locally linearize the system and then design a linear feedback control to force the system to be locally contractive. Our approach is similar to [10,25] and can be seen as an instance of an ε-close bisimulation as in [11]. However, in contrast to [25], the states of our abstraction can overlap, which is crucial for building non-trivial smart abstractions [10]. As opposed to [10], we solve a linear programming problem to ensure contraction in terms of infinite norms, not quadratic norms.

Structure The remainder of the paper is structured as follows. In Sect. 2, we describe the fundamental building blocks of the abstraction, notably LTS semantics and alternating simulation. In Sect. 3, we define a particular type of alternating simulation that we call policy abstraction, which maps abstract transitions to feedback policies in the concrete system. We discuss the particular power of deterministic abstractions and show their existence for systems that are globally Lipschitz and contractive. Section 4 extends the approach to more general classes of systems by applying continuous feedback control to induce contractiveness locally. Numerical experiments illustrate the approach in Sect. 5.

2 Discrete Abstractions of Continuous Systems

In this section, we present the fundamental notions that describe the relationship between the continuous dynamical system that we want to supervise and the discrete abstractions that we use to build the supervisor: LTS semantics, simulation, and alternating simulation.

2.1 Disrete-time Dynamical Control Systems

We consider discrete-time dynamical systems with control inputs, with a set of states $\mathcal{X} \subseteq \mathbb{R}^n$ and a closed and bounded set of control actions $\mathcal{U} \subset \mathbb{R}^m$. The dynamics are described by a set-valued map $F\colon \mathcal{X} \times \mathcal{U} \rightrightarrows \mathcal{X}$ and $F(x,u) \neq \emptyset$ for any $x \in \mathcal{X}, u \in \mathcal{U}$. For a given initial state x_0, a trajectory consists of a sequence

$$x'_{k+1} \in F(x_k, u_k), \qquad x_k \in \mathcal{X},\ u_k \in \mathcal{U}. \tag{1}$$

We extend $F(x,u)$ to sets $X \in \mathcal{X},\ U \subseteq \mathcal{U}$ as $F(X,U) = \bigcup_{x\in\mathcal{X}, u\in\mathcal{U}} F(x,u)$. Based on the one-step forward reachset F, the robust one-step backward reachset

$B(\mathcal{X}',\mathcal{U}')$ is defined as the states where choosing the right action from $\mathcal{U}' \subseteq \mathcal{U}$ leads always inside the target set $\mathcal{X}' \subseteq \mathcal{X}$:

$$B(\mathcal{X}',\mathcal{U}') = \left\{ x \;\middle|\; \exists u \in \mathcal{U}' : F(x,u) \subseteq \mathcal{X}' \right\}. \tag{2}$$

A *(feedback) policy* is a set-valued function $\pi : \mathcal{X} \rightrightarrows \mathcal{U}$ associating each state with a set of available inputs. Connecting a system with dynamics F with a feedback policy π leads to the closed-loop dynamics F_π defined by $F_\pi(x) = \bigcup_{u \in \pi(x)} F(x,u)$. Let $\Pi(\mathcal{X},\mathcal{U})$ be the set of all policies over $\mathcal{X}$ and $\mathcal{U}$. A policy is called *non-blocking* if for all $x \in \mathcal{X}$, we have that $\pi(x) \neq \emptyset$.

2.2 LTS Semantics

Two main types of transition systems that are used as abstractions are Kripke structures (KS), whose states are labeled with atomic propositions and whose transitions are unlabeled, and Labeled Transition Systems (LTS), whose states are unlabeled and whose transitions are labeled. Both formalisms are essentially equivalent [7], but KS-type abstractions seem to have been favored for continuous systems, e.g., in [20]. We are concerned with both state information, wishing to direct the system to certain states, and the transitions associated with input actions. Following the example of [9], we include both kinds of labels. This has been called a *doubly labeled transition system* [8], but for the sake of simplicity, we will stick with LTS, defined as follows.

Definition 1. *A* labeled transition system *(LTS) is* $L = (S, s_0, \Sigma, \rightarrow, P, O)$ *with*

- *a set of states* S*, including a state* s_0 *called the* initial state*,*
- *a set of action labels* Σ*,*
- *a transition relation* $\rightarrow \subseteq S \times \Sigma \times S$*, where* $s \xrightarrow{\alpha} s'$ *denotes that the system can transition from state* s *to* s' *if the action* α *is applied,*
- *a set of atomic observations* P*,*
- *an observation function* $O : S \rightrightarrows P$ *that attributes to each state* s *all observations that hold in* s*.*

An LTS is called nonblocking *if every state has at least one outgoing transition. It is called* complete *over a set of labels* Σ' *if every state has at least one outgoing transition for every label in* Σ'*.*

The semantics of the LTS are defined over runs and traces of observations, which then lead us to specifications, following the approach proposed in [9]:

Definition 2 (Run, trace, specification). *A* run $\rho = s_0 \xrightarrow{\pi_0} s_1 \xrightarrow{\pi_1} s_2 \ldots$ *is a (finite or infinite) sequence of alternating states and labels, starting in the initial state and connected by the transition relation. Applying the observation function to each state in a run maps it to a* trace*, i.e., the sequence of sets of observations* $\tau = O(\rho) = o_0, o_1, o_2 \ldots$ *with* $o_k = O(s_k)$ *for all* k*. We denote with* $\tau \leq \tau'$ *for traces* $\tau = o_0, o_1, o_2 \ldots$*,* $\tau' = o'_0, o'_1, o'_2 \ldots$ *if* $o_k \subseteq o'_k$ *for all* k*, i.e., all observations in* τ *also hold in* τ'*. A* specification Spec *is a set of traces that is upwards closed, i.e., for any* $\tau \in \mathsf{Spec}$ *and* τ' *with* $\tau \leq \tau'$*, we have* $\tau' \in \mathsf{Spec}$*. We say that a run* ρ satisfies *a specification* Spec *if* $O(\rho) \in \mathsf{Spec}$*.*

The upwards-closedness of specification follows intuition: When the specification requires that observations $\{a, b\}$ hold, it should also be satisfied if $\{a, b, c\}$ hold since $a \wedge b \wedge c$ implies $a \wedge b$. Our dynamical systems are cast as LTS as follows:

Definition 3. *The semantics of the dynamical system* (1) *is a labeled transition system* $L = (\mathcal{X}, x_0, \mathcal{U}, \rightarrow_{\mathcal{X}}, P, O)$*, with state domain* $\mathcal{X}$*, the label set* $\mathcal{U}$ *and transition relation* $\rightarrow_{\mathcal{X}}$ *defined by* $x \xrightarrow{u}_{\mathcal{X}} x'$ *if and only if* $x' \in F(x, u)$.

The set of atomic observations P and the output function O depend on the type of observation we wish to consider. Assuming perfect state information, we have $P = \mathcal{X}$ and $O(x) = \{x\}$. Alternatively, O can represent a quantizer that indicates states up to a neighborhood; P consists then of the different possible neighborhoods:

Definition 4. *Let* $P = \{\mathcal{S}_1, \mathcal{S}_2, \ldots\}$ *be a collection of sets that cover* $\mathcal{X}$*. We call* state quantizer *the output function* $O_P(x) = \{\mathcal{S} \in P \mid x \in \mathcal{S}\}$.

If P is a partition of $\mathcal{X}$, the state quantizer is deterministic. If the sets in P overlap, e.g., to model noisy measurements, then O is nondeterministic.

2.3 Abstractions on LTS

We follow the classical route by defining abstractions using simulation relations. From KS, we adopt that outputs match, and from LTS, that labels match.

Definition 5 (Simulation). *Given a pair of LTS* (L_1, L_2) *and* $P_1 = P_2$*, a relation* $R \subseteq S_1 \times S_2$ *is a* KS-simulation relation *if* $(s_{0,1}, s_{0,2}) \in R$ *and for all* $(s_1, s_2) \in R$

(i) $O_2(s_2) \subseteq O_1(s_1)$ *and* [1]
(ii) $\forall s_1 \xrightarrow{u_1}_1 s_1'$ *there exists* u_2, s_2' *such that* $s_2 \xrightarrow{u_2}_2 s_2'$ *and* $(s_1', s_2') \in R$.

R *is a* LTS-simulation relation *if in (ii),* $u_1 = u_2$*.* R *is a* bisimulation relation *if it is a simulation relation of* (L_1, L_2) *and its converse* R^T *is a simulation relation of* (L_2, L_1)*.* L_2 *is said to* simulate L_1 *if there is a simulation relation* R *for* (L_1, L_2).

In typical use, L_1 would be the concrete system and L_2 the abstraction, which may admit more transitions than the concrete system. Simulation preserves safety properties: If the abstract system L_2 remains inside a set of safe states, then so does L_1. For example, assume $\mathsf{safe} \in O(s_2)$ if and only if s_2 is considered safe in L_2, and let only safe states be reachable from the initial state $s_{0,2}$. If S_2 simulates S_1, then $\mathsf{safe} \in O(s_1)$ for all reachable s_1 in L_1.

We now turn to the question of when a transition in the abstraction can be enforced in the concrete system. We consider nondeterministic dynamical

[1] [9] uses the converse condition, i.e., $O_1(s_1) \subseteq O_2(s_2)$. We consider our version consistent with the upward-closure requirement for specifications; other work also uses this direction [16].

systems, which may end up in different states for the same input. This can be modeled as a two-player game in which an agent plays the input action, and the opponent gets to pick the successor state. For this situation, *alternating simulation* [2] provides a relationship that is particularly suited to control systems [20]:

Definition 6 (Alternating Simulation, adapted from [20]). *Given a pair of LTS* (L_1, L_2) *with* $P_1 = P_2$*, a relation* $R \subseteq S_1 \times S_2$ *is an* alternating simulation relation *if* $(s_{0,1}, s_{0,2}) \in R$ *and for all* $(s_1, s_2) \in R$

- $O_1(s_1) \subseteq O_2(s_2)$,
- $\forall s_1 \xrightarrow{u_1}_1 s_1'$ *there exist* u_2, s_2' *such that* $s_2 \xrightarrow{u_2}_2 s_2'$ *with* $(s_1', s_2') \in R$ *and* $\forall s_2 \xrightarrow{u_2}_2 s_2''$ *there exist* s_1'' *such that* $s_1 \xrightarrow{u_1}_1 s_1''$ *with* $(s_1'', s_2'') \in R$.

In our case, L_2 is the concrete system and L_1 the abstraction: If the abstraction L_1 proposes a move, then L_2 must be able to realize that move with an action, and all other possible successor states for that action (s_2'') must also be in the relation. Intuitively, an alternating simulation over (L_1, L_2) guarantees that any action in L_1 can be implemented in L_2. Similarly, any sequence of observations that can be realized in L_1 can also be realized in L_2 without any risk of nondeterminism leading to different traces [26, Thm. 1].

3 Supervision with Deterministic Policy Abstractions

Our goal is to represent and synthesize supervisors efficiently. We employ discrete—ideally, finite—LTS as models for this. In what we call a *policy abstraction*, every transition $s \xrightarrow{\pi}_A s'$ attributes a policy π to the change from s to s'. In the literature, abstractions are frequently based on the forward reach set $F(\mathcal{X}', \mathcal{U}')$, where $\mathcal{X}'$ are the concrete states associated with s, so the input is the same for all states in $\mathcal{X}'$. This case, which we call *reach set abstraction*, is covered by letting $\pi(x) = \mathcal{U}'$ for all $x \in \mathcal{X}$.

3.1 Policy Abstractions

To relate the abstraction to the control system, we use a special case of alternating simulation, where the relationship between the abstract labels (policies) and concrete labels (input actions) is not entirely arbitrary: the abstract labels are policies, i.e., they map concrete states to a set of concrete labels.

Definition 7 (Reach-set and Policy Abstraction). *Given a dynamical system* (1)*, a set of observations* P *and an observation function* $O : \mathcal{X} \rightrightarrows P$*, a* policy abstraction *is an LTS* $A = (S, s_0, \Pi, \rightarrow_A, P, O_A)$*, where* Π *is a set of nonblocking policies (mapping* $\mathcal{X}$ *to nonempty subsets of* $\mathcal{U}$*), such that there exists a relation* $R \subseteq S \times \mathcal{X}$ *(an alternating simulation relation), with* $(s_0, x_0) \in R$ *and for all* $(s, x) \in R$

(i) $O_A(s) \subseteq O(x)$,

(ii) $\forall s \xrightarrow{\pi}_A s', \forall x'' \in F(x, \pi(x))$ *there exists* s''*:* $s \xrightarrow{\pi}_A s''$ *with* $(s'', x'') \in R$*.*

If all policies in $\xrightarrow{\pi}_A$ *are independent of the continuous state, i.e.,* $\pi = \mathcal{X} \times \mathcal{U}'$ *for some* $\mathcal{U}' \subseteq \mathcal{U}$*, we call* A *a* reach set abstraction*.*

Simply put, the abstraction must cover the forward reach set in the concrete system with abstract transitions that have the same label. The relation R above is an alternating simulation relation over A and the LTS semantics of F, with an additional constraint on the correspondence between labels.

We can operate a policy abstraction as a supervisor on the system. This leads to a hybrid system, i.e., discrete states representing the policy abstraction, similar to the approach in [10].

Definition 8 (Supervision by a PA). *The semantics of the dynamical system* (1) supervised *by a policy abstraction* A *is the LTS* $L||A = (\mathcal{X}_A, x_{A,0}, \mathcal{U}, \rightarrow_{X||A}, P, O)$ *with* $\mathcal{X}_A = \mathcal{X} \times S$*,* $x_{A,0} = (x_0, s_0)$*, and* $(x, s) \xrightarrow{u}_{X||A} (x', s')$ *if and only if there exists* $s \xrightarrow{\pi}_A s'$ *such that* $u \in \pi(x)$ *and* $x' \in F(x, u)$*. These dynamics correspond to the one-step forward reachset*

$$F_A\big((x,s)\big) = \Big\{(x', s') \;\Big|\; \exists \pi \in \Pi, s' \in S : s \xrightarrow{\pi}_A s', x' \in F(x, \pi(x))\Big\}.$$

The supervised control system matches the policy abstraction if the output function matches exactly. We formalize this relationship as a bisimulation.

Proposition 1. *If for all* (s, x) *in the relation* R *in Def. 7,* $O_A(s) = O(x)$*, then* $L||A$ *is a KS-bisimulation of* A*.*

We are particularly interested in the special case where the policy abstraction is *deterministic*, i.e., in Def. 7, $s' = s''$. A deterministic policy abstraction A encodes traces that can be realized by applying the corresponding sequence of policies from A to the concrete system L. We formalize this by specializing a result from [9] to deterministic policy abstractions:

Theorem 1 (Deterministic PA). *Let* $s_0 \xrightarrow{\pi_0} s_1 \xrightarrow{\pi_1} s_2 \ldots$ *be a run of a deterministic policy abstraction* A *that satisfies a given specification* Spec*. Then any run* $x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} x_2 \ldots$ *of* L *with* $u_k \in \pi_k(x_k)$ *for all* k *also satisfies* Spec*.*

Proof. Let $o_0, o_1, \ldots$ be the trace of $s_0 \xrightarrow{\pi_0}_A s_1 \xrightarrow{\pi_1}_A s_2 \ldots$ and $o'_0, o'_1, \ldots$ the trace of $x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} x_2 \ldots$ We first show by induction that $(s_k, x_k) \in R$ for all $k \geq 0$. Induction start: For $x = 0$, we have $(s_0, x_0) \in R$ from the definition of R in Def. 7. Induction step: Let $(s_k, x_k) \in R$. With $s_k \xrightarrow{\pi_k}_A s_{k+1}$, $\forall x'' \in F(x_k, \pi_k(x_k))$ we get $(s_{k+1}, x'') \in R$. Since $x_{k+1} \in F(x_k, \pi_k(x_k))$, we get $(s_{k+1}, x_{k+1}) \in R$. Since $(s_k, x_k) \in R$, $O_A(s_k) \subseteq O(x_k)$ and therefore $t \leq t'$. Since $t \in$ Spec and Spec is upwards closed, we get $t' \in$ Spec, which concludes the proof.

In the sequel, we use abstractions based on a discrete cover of the states and inputs; we give a generic definition below and will the following sections present ways to compute the corresponding parameters and identify suitable policies.

We extensively use vector norms to define sets of states since this reduces set containment relationships to linear inequalities. The results in the remainder of the paper hold for arbitrary vector and matrix norms provided that they are consistent. Let $\mathbb{B}(\varepsilon)$ denote the n-dimensional ball with vector norm ε (centered around the origin), and $\Delta(X)$ the diameter of the set X, i.e., the diameter of the smallest ball containing X.

Definition 9 (Neighborhood-based Abstraction). *Consider a dynamical system* (1) *equipped with quantizer output, i.e., with LTS semantics* $L = (\mathcal{X}, x_0, \mathcal{U}, \rightarrow_{\mathcal{X}}, P, O_P)$*, and consider covers of* $\mathcal{X}$ *and* $\mathcal{U}$*, defined with distance parameters* α *and* β *as follows. Let* $\hat{X}$ *be a set of pairwise distinct points such that* $\hat{X} \oplus \mathbb{B}(\alpha)$ *covers* $\mathcal{X}$ *and let* $\hat{U}$ *be a set of pairwise distinct points such that* $\hat{U} \oplus \mathbb{B}(\beta)$ *covers* $\mathcal{U}$*. We associate each* $\hat{x} \in \hat{X}$ *with its neighborhood of radius* $\varepsilon_{\hat{x}} \geq \alpha$*. Let the observations be* $P = \{\hat{x} \oplus \mathbb{B}(\varepsilon_{\hat{x}}) \mid \hat{x} \in \hat{X}\}$ *and let* O_P *be the state quantizer in Def. 4. Let* A *be the LTS* $(\hat{X}, \hat{x}_0, \hat{X} \times \mathcal{U}, \rightarrow_A, P, O_A)$ *defined as follows. Let* $\hat{x}_0$ *be any of the points in* $\hat{X}$ *that are closest to* x_0*. Let* $O_A(\hat{x}) = \hat{x} \oplus \mathbb{B}(\varepsilon_{\hat{x}})$*. We call* $\varepsilon = \sup\{\varepsilon_{\hat{x}} \mid \hat{x} \in \hat{X}\}$ *the* accuracy *of* A*.*

It is straightforward to show that the above abstraction satisfies the conditions of a policy abstraction:

Proposition 2. *Any Neighborhood-based Abstraction* A*, as defined in Def. 9, is a deterministic policy abstraction (witnessed by an alternating simulation relation) if for all* $\hat{x}, \pi, \hat{x}'$ *with* $\hat{x} \xrightarrow{\pi}_A \hat{x}'$ *holds that*

$$x \in \hat{x} \oplus \mathbb{B}(\varepsilon_{\hat{x}}) \quad \Rightarrow \quad \big\|F\big(x, \pi(x)\big) - \hat{x}'\big\| \leq \varepsilon_{\hat{x}'}. \tag{3}$$

3.2 Existence of Deterministic Reach Set Abstractions

We consider dynamical systems from (1) satisfying the following assumption:

Assumption 1 *1. The radius of the reach sets is globally bounded by*

$$\omega = \sup_{x \in X, u \in U} \tfrac{1}{2} \Delta\big(F(x,u)\big).$$

2. There are constants $K_x, K_u \geq 0$ *such that for all* $x, x' \in \mathcal{X}$*,* $u, u' \in \mathcal{U}$*:*

$$F(x', u') \subseteq F(x,u) \oplus \mathbb{B}\big(K_x ||x - x'|| + K_u ||u - u'||\big).$$

The above assumptions may, in general, not be satisfied. E.g., if X or U are unbounded, the sup might not exist. If the assumptions are satisfied, a discrete abstraction exists:

Proposition 3. *Consider a dynamical system satisfying Assumption 1 and let* A *be the LTS* $(\hat{X}, \hat{x}_0, \Pi(\mathcal{X}, \mathcal{U}), \rightarrow_A, P, O_A)$ *as defined in Def. 9, i.e., a neighborhood-based abstraction. Let the accuracy* ε *be uniform, i.e.,* $\varepsilon_{\hat{x}} = \varepsilon$ *for all* $\hat{x}$*. Let* $\hat{U}$ *be a set of pairwise distinct points such that* $\hat{U} \oplus \mathbb{B}(\beta)$ *covers* $\mathcal{U}$*. Let* $\rightarrow_A$ *be defined as* $\hat{x} \xrightarrow{\pi}_A \hat{x}'$ *for all combinations of* $\hat{x}, \hat{x}' \in \hat{X}, \hat{u} \in \hat{U}$ *that satisfy*

$||F(\hat{x},\hat{u}) - \hat{x}'|| \leq \omega + \alpha$, *with* $\pi = \mathcal{X} \times \big(\hat{u} \oplus \mathbb{B}(\beta)\big) \cap \mathcal{U}$. *Let the discretization parameters* α, ε, β *be such that*

$$K_x < 1 \text{ and } \varepsilon \geq \tfrac{\omega + K_u\beta + \alpha}{1 - K_x}, \tag{4}$$

Under the above conditions, the LTS A *is a deterministic reach set abstraction of the concrete system* L*, and consequently, there is an alternating simulation relation* R *that witnesses this relationship. Furthermore,* R^T *is also a witness that* A *KS-simulates* L*.*

Proof. We first show by structural induction that $R = \big\{\big(\hat{x}, x\big) \;\big|\; \hat{x} \in \hat{X}, x \in \hat{x} \oplus \mathbb{B}(\varepsilon)\big\}$ is a PA relation. Since $K_x < 1$, we have $\varepsilon > \alpha$. Hence $x_0 \in \hat{x}_0 \oplus \mathbb{B}(\varepsilon) = O_A(\hat{x}_0)$ and $(\hat{x}_0, x_0) \in R$. Assume $(\hat{x}, x) \in R$, so that $x \in \hat{x} \oplus \mathbb{B}(\varepsilon)$. For all $x \in O_A(\hat{x})$, $O_A(\hat{x}) \in O_P(x)$ by Def. 4, which satisfies condition (i). Now assume $\hat{x} \xrightarrow{\pi}_{PA} \hat{x}'$. Under the hypothesis,

$$F(\hat{x} \oplus \mathbb{B}(\varepsilon), \hat{u} \oplus \mathbb{B}(\beta)) \subseteq F(\hat{x}, \hat{u}) \oplus \mathbb{B}(K_x\varepsilon + K_u\beta).$$

Under the hypothesis, $F(\hat{x}, \hat{u}) \subseteq \hat{x}' \oplus \mathbb{B}(\omega + \alpha)$, so that with (4):

$$F(\hat{x} \oplus \mathbb{B}(\varepsilon), \hat{u} \oplus \mathbb{B}(\beta)) \subseteq \hat{x}' \oplus \mathbb{B}(\alpha + \omega + K_x\varepsilon + K_u\beta) \subseteq \hat{x}' \oplus \mathbb{B}(\varepsilon).$$

This satisfies condition (ii) and concludes the proof for R.

We now show that R^T is a simulation relation for (L, A). By definition, $x \xrightarrow{u}_X x'$ means that $x' \in F(x, u)$. Since $\mathcal{X}$ is covered by $\hat{X} \oplus \mathbb{B}(\alpha)$ and $\mathcal{U}$ is covered by $\hat{U} \oplus \mathbb{B}(\beta)$, there are $\hat{x}$, π and $\hat{u}$ such that $x \in \hat{x} \oplus \mathbb{B}(\varepsilon)$ and $u \in \pi(x) = \hat{u} \oplus \mathbb{B}(\beta)$. Since $x' \in F(x, u)$, under the hypothesis,

$$x' \in F(\hat{x} \oplus \mathbb{B}(\varepsilon), \hat{u} \oplus \mathbb{B}(\beta)) \subseteq F(\hat{x}, \hat{u}) \oplus \mathbb{B}(K_x\varepsilon + K_u\beta),$$

and there is some $\hat{x}' \in \hat{X}$ with

$$F(\hat{x}, \hat{u}) \oplus \mathbb{B}(K_x\varepsilon + K_u\beta) \subseteq \hat{x}' \oplus \mathbb{B}(\alpha + \omega + K_x\varepsilon + K_u\beta) \subseteq \hat{x}' \oplus \mathbb{B}(\varepsilon).$$

Therefore, $x' \in \hat{x}' \oplus \mathbb{B}(\varepsilon)$, and $\big(x', \hat{x}' \oplus \mathbb{B}(\varepsilon)\big) \in R^T$. The transition follows from

$$\begin{aligned} F(\hat{x}, \hat{u}) \oplus \mathbb{B}(K_x\varepsilon + K_u\beta) \subseteq \hat{x}' \oplus \mathbb{B}(\alpha + \omega + K_x\varepsilon + K_u\beta) \quad &\Leftrightarrow \\ F(\hat{x}, \hat{u}) \subseteq \hat{x}' \oplus \mathbb{B}(\alpha + \omega) \quad &\Leftrightarrow \\ ||F(\hat{x}, \hat{u}) - \hat{x}'|| \leq \alpha + \omega. \end{aligned}$$

Therefore $\hat{x} \xrightarrow{\pi}_{\mathrm{A}} \hat{x}'$. The LTS A is deterministic since the $\hat{u}$, and therefore the labels π for each transition, are pairwise distinct.

Corollary 1. *If* F *is deterministic (*$\omega = 0$*) and Lipschitz over* $\mathcal{X}$ *with constant* $K_x < 1$ *and Lipschitz over* $\mathcal{U}$*, then a deterministic forward reach set approximation can be constructed with arbitrarily small accuracy* ε *by setting* α *and* β *small enough.*

The above result on the existence of a discrete abstraction is consistent with results from the literature on nondeterministic abstractions, e.g., [20], where it is associated with global asymptotic stability. According to our above result, we can obtain a deterministic abstract and, therefore, the full power of Theorem 1, without any particular downside.

4 Policy Abstraction through Linear Feedback

In Sect. 3.2, we described a deterministic abstraction obtained using only the forward reach set operator. This works for dynamics that are globally contractive. We now extend the applicable cases to systems that are not contractive everywhere by designing policies. One way to achieve this would be to use the robust backward reach set operator: The states for which there exists a policy that drives them towards a target set is exactly the backward reach set in (2). However, the backward reach set operator is much more expensive to compute than the forward operator, even for linear systems [26]. There is a fundamental limitation to backward reachability, even without robust control: a system that is stable going forward in time is unstable going backward in time [18]. Our approach is to stick with forward reachability and use linear state feedback to achieve contractiveness. The reach set computation and feedback design is based on linear control.

4.1 Linearizing over Discrete States

We base our analysis on linearization and assume a global limit $\varepsilon_{\max}$ on the size of the neighborhoods around the states we consider. Let the input set be a ball $\mathcal{U} = \mathbb{B}(\delta)$ with $\delta > 0$. We assume the following linearization is available. Given linearization points $(\hat{x}, \hat{u}) \in \hat{X} \times \hat{U}$, let $A_{\hat{x},\hat{u}}, B_{\hat{x},\hat{u}}$ be matrices, $c_{\hat{x},\hat{u}}$ a vector and $\gamma_{\hat{x},\hat{u}}, \omega$ be scalars such that for all $\varepsilon \leq \varepsilon_{\max}$, $x \in \hat{x} \oplus \mathbb{B}(\varepsilon), u \in \mathbb{B}(\delta) \cap \big(\hat{u} \oplus \mathbb{B}(\beta)\big)$,

$$F(x,u) \subseteq c_{\hat{x},\hat{u}} \oplus A_{\hat{x},\hat{u}}(x-\hat{x}) \oplus B_{\hat{x},\hat{u}}(u-\hat{u}) \oplus \mathbb{B}\Big(\gamma_{\hat{x},\hat{u}}\big(||x-\hat{x}|| + ||u-\hat{u}||\big) + \omega\Big). \quad (5)$$

In the following, we associate each transition $\hat{x} \xrightarrow{\pi_{\hat{x},\hat{u},\hat{x}'}}_A \hat{x}'$ in the abstraction with an input $\hat{u}$ and a constant feedback control policy

$$\pi_{\hat{x},\hat{u},\hat{x}'}(x) = \left\{\hat{u} - K_{\hat{x},\hat{u},\hat{x}'} \tfrac{x-\hat{x}}{\varepsilon_{\hat{x}}} \oplus \mathbb{B}(\beta)\right\} \cap \mathcal{U}, \quad (6)$$

where $K_{\hat{x},\hat{u},\hat{x}'}$ is a real-valued matrix mapping concrete states to the space of control inputs. We exploit this feedback matrix to locally induce contractiveness as needed (but not necessarily everywhere). Note that scaling the feedback matrix by $1/\varepsilon_{\hat{x}}$ is a trick to obtain linear constraints in the sequel, e.g., in (8) and Prop. 4. To avoid saturating the input signal, we must ensure that $\pi_{\hat{x},\hat{u},\hat{x}'}(x) \in \mathcal{U}$ for all $x \in \hat{x} \oplus \mathbb{B}(\varepsilon_{\hat{x}})$, which is surely the case if

$$\big\|\hat{u} - K_{\hat{x},\hat{u},\hat{x}'}\mathbb{B}(1) + \beta\big\| \leq \delta \quad (7)$$

Substituting the policy (6) in (5), we obtain for all $x \in \hat{x} \oplus \mathbb{B}(\varepsilon)$:

$$F\big(x, \pi(x)\big) \subseteq c_{\hat{x},\hat{u}} \oplus \big(A_{\hat{x},\hat{u}} - B_{\hat{x},\hat{u}}K_{\hat{x},\hat{u},\hat{x}'}/\varepsilon_{\hat{x}}\big)(x-\hat{x}) \oplus B_{\hat{x},\hat{u}}\mathbb{B}(\beta) \\ \oplus \mathbb{B}\Big(\big(\gamma_{\hat{x},\hat{u}} + ||K_{\hat{x},\hat{u},\hat{x}'}||/\varepsilon_{\hat{x}}\big)||x-\hat{x}|| + \omega\Big).$$

We apply the above to the containment relationship (3) to derive a constraint that is sufficient for a deterministic policy abstraction:

$$\begin{aligned}\big\|c_{\hat{x},\hat{u}} - \hat{x}' \oplus \big(A_{\hat{x},\hat{u}}\varepsilon_{\hat{x}} - B_{\hat{x},\hat{u}}K_{\hat{x},\hat{u},\hat{x}'}\big)\mathbb{B}(1) \oplus B_{\hat{x},\hat{u}}\mathbb{B}(\beta)\big\| \\ + \gamma_{\hat{x},\hat{u}}\varepsilon_{\hat{x}} + ||K_{\hat{x},\hat{u},\hat{x}'}|| + \omega \leq \varepsilon_{\hat{x}'}. \quad (8)\end{aligned}$$

Proposition 4. *Consider a dynamical system* (1) *with LTS* L *and linearization* (5)*. Let* A *be the LTS* $(\hat{X}, \hat{x}_0, \Pi(\mathcal{X},\mathcal{U}), \rightarrow_A, P, O_A)$ *as defined in Def. 9, i.e., a neighborhood-based abstraction. Let* $\hat{U}$ *be a set of pairwise distinct points Let* $I \subseteq \hat{X} \times \hat{U} \times \hat{X}$ *be a given set of tuples* $(\hat{x}, \hat{u}, \hat{x}')$ *such that the following conjunction of linear constraints on the variables* $\varepsilon_{\hat{x}}, K_{\hat{x},\hat{u},\hat{x}'}, \beta$ *is satisfiable:*

$$\begin{aligned}&\bigwedge_{(\hat{x},\hat{u},\hat{x}')\in I} \big\|c_{\hat{x},\hat{u}} - \hat{x}' \oplus \big(A_{\hat{x},\hat{u}}\varepsilon_{\hat{x}} - B_{\hat{x},\hat{u}}K_{\hat{x},\hat{u},\hat{x}'}\big)\mathbb{B}(1) \oplus B_{\hat{x},\hat{u}}\mathbb{B}(\beta)\big\| + \gamma_{\hat{x},\hat{u}}\varepsilon_{\hat{x}} \\ &+ ||K_{\hat{x},\hat{u},\hat{x}'}|| + \omega \leq \varepsilon_{\hat{x}'} \wedge \quad \big\|\hat{u} - K_{\hat{x},\hat{u},\hat{x}'}\mathbb{B}(1) + \beta\big\| \leq \delta \wedge \alpha \leq \varepsilon_{\hat{x}} \leq \varepsilon_{\max}. \quad (9)\end{aligned}$$

Let the transition relation consist of $\hat{x} \xrightarrow{\pi_{\hat{x},\hat{u},\hat{x}'}}_A \hat{x}'$ *for all* $(\hat{x}, \hat{u}, \hat{x}') \in I$*, with* $\pi_{\hat{x},\hat{u},\hat{x}'}$ *as defined in* (6)*. Then* A *is a deterministic policy abstraction, witnessed by an alternating simulation relation.*

It is straightforward to show that any solution of Prop. 3 also satisfies Prop. 4 ($K_x = ||A_{\hat{x},\hat{u}}||$, $K_u = ||B_{\hat{x},\hat{u}}||$, $\gamma_{\hat{x},\hat{u}} = 0$, $||c_{\hat{x},\hat{u}} - \hat{x}'|| \leq \alpha$ and letting $K_{\hat{x},\hat{u},\hat{x}'} = 0$), so we can rest assured that Prop. 4 is strictly more powerful. The constraint system (9) is linear in the variables $\varepsilon_{\hat{x}}, K_{\hat{x},\hat{u},\hat{x}'}, \beta$ if we use the infinity norm (the 1-norm works, too). This means that a solution can be found efficiently. In addition, we cast it as a linear optimization problem to find a solution with the highest precision (minimize a global bound on $\varepsilon_{\hat{x}}$) or permissiveness (maximize β). The constraints can be further simplified; we limit the discussion for lack of space. An important special case is $B_{\hat{x},\hat{u}}$ having full row rank, since it maximizes the capacity of the feedback controller to make the system contractive. A standard reduction to this case consists of modeling the system at every p-th step for some p: The dynamics of this sub-sampled system are given by

$$F_p(x, [u_{(1)}; \ldots; u_{(p)}]) = F\Big(\cdots F\big(F(x, u_{(1)}), u_{(2)}\big) \cdots, u_{(p)}\Big),$$

where the augmented input vector $u = [u_{(1)}; \ldots; u_{(p)}]$ consists of the concatenation of p input vectors of the original system (one for each time step). The outputs of the system must, of course, be adapted accordingly to preserve the desired properties.

4.2 Selecting Transitions

The main challenge in constructing the abstraction proposed in Prop. 4 is selecting which tuples of transitions $(\hat{x}, \hat{u}, \hat{x}')$ to include in the set I. First, we note

that in cases where the linearization $A_{\hat{x},\hat{u}}, B_{\hat{x},\hat{u}}, c_{\hat{x},\hat{u}}$ is independent of $\hat{u}$, we can declare $\hat{u}$ as a variable in the constraint system (9). This reduces the search to pairs $(\hat{x}, \hat{x}')$.

Transitions that are infeasible in the concrete system can be ruled out, e.g., by forward or backward reachability analysis (which one is more precise depends on whether the system is locally stable or instable [18]). However, even if we somehow include only transitions that individually can be concretized, this does not mean that the constraint system (9) is satisfiable.

We propose an iterative approach: Starting from an initial set I_0 (possibly very conservative), we identify one or more problematic transitions, remove them to obtain I_1, and repeat the process until (9) becomes satisfiable (possibly because there are no transitions left). Each time the constraint system (9) is infeasible, the LP solver helpfully provides us with a collection of Irreducible Infeasible Subsets (IIS) of constraints. In our abstraction, each IIS corresponds to a cycle in the transition graph, and by encoding the problem accordingly, the IIS allows us to detect the transitions in the cycle.

Depending on the properties that we wish to preserve, we may be able to prioritize transitions. An approach to identify a minimal subset that preserves reachability relationships between states is described in [14]. A stronger requirement is to preserve strongly connected components (see discussion in Sect. 5); an approach to find the such minimal transition set is given in [3]. Finally, the slack variables returned by the solver can provide quantitative information on which constraints are harder to satisfy, which we can translate into priorities on transitions.

5 Experiments

We present experiments on discrete-time versions of two nonlinear systems with unstable equilibria: the inverted pendulum (IP) and the Van der Pol oscillator (VdP) [15]. Both examples have unstable equilibria, and VdP has a stable limit cycle, so constructing an abstraction is challenging. The continuous-time dynamics are given by the ODE $\dot{x} = f(x, u)$, with:

$$f_{\mathrm{IP}}(x,u) = \begin{pmatrix} x_2 \\ -\frac{g}{l}\sin x_1 + \frac{\beta}{ml^2}x_2 + u \end{pmatrix} \quad f_{\mathrm{VdP}}(x,u) = \begin{pmatrix} 2x_2 \\ -0.8x_1 + 2x_2 - 10x_1^2 x_2 + u \end{pmatrix}, \quad (10)$$

with parameters $g = -9.81, l = 1, \beta = 1, m = 1$. We consider inputs in $\mathcal{U} = [-8, 8]$ for IP and $\mathcal{U} = [-2, 2]$ for VdP. The nonlinear dynamics are linearized, and the approximation error is over-estimated using Taylor models from interval analysis [5,1]. The parameters $\gamma_{\hat{x},\hat{u}}, \omega$ of the linearization are chosen such that the linearization is guaranteed to contain the original forward reach set by taking into account all approximation and linearization errors. The linearized dynamics are then integrated to obtain the discrete dynamics over a given time step h ($h = 1$ for IP and $h = 0.04$ for VdP), keeping the inputs constant over the time interval. To achieve full rank in B, we take two time steps at a time, i.e., the input of the abstraction is two-dimensional.

Table 1. Policy Abstractions obtained through Linearization

	Inv. Pendulum	Van der Pol
Number of transitions	835	452
Maximum out-degree	14	3
Average out-degree	6.9	2.0
$\min_{\hat{x}} \varepsilon_{\hat{x}}$	0.100	0.050
$\max_{\hat{x}} \varepsilon_{\hat{x}}$	0.100	0.069
$\min_{\hat{x},\hat{u}} \left\|A_{\hat{x},\hat{u}} - B_{\hat{x},\hat{u}} K_{\hat{x},\hat{u},\hat{x}'}/\varepsilon_{\hat{x}}\right\|_\infty$	0.42	0.71
$\max_{\hat{x},\hat{u}} \left\|A_{\hat{x},\hat{u}} - B_{\hat{x},\hat{u}} K_{\hat{x},\hat{u},\hat{x}'}/\varepsilon_{\hat{x}}\right\|_\infty$	0.95	2.27
avg. $\left\|A_{\hat{x},\hat{u}} - B_{\hat{x},\hat{u}} K_{\hat{x},\hat{u},\hat{x}'}/\varepsilon_{\hat{x}}\right\|_\infty$	0.77	1.21

Some statistics on the abstractions are given in Table 1, for a discretization parameter $\alpha = 0.05$. For the inverted pendulum, the closed-loop dynamics of the system with feedback are strictly contractive everywhere, i.e.,

$$\max_{\hat{x},\hat{u}} \left\|A_{\hat{x},\hat{u}} - B_{\hat{x},\hat{u}} K_{\hat{x},\hat{u},\hat{x}'}/\varepsilon_{\hat{x}}\right\|_\infty = 0.95 < 1.$$

As a consequence, we can use the same precision $\hat{\varepsilon} = 0.1$ everywhere. Since the closed-loop system is quite contractive on average, the average out-degree is elevated, i.e., we can deterministically choose from an average of 6.9 possible successor states. Note that the average out-degree is biased by states on the border of the domain, many of which have out-degree zero.

For the Van der Pol oscillator, the closed-loop dynamics are, on average, enlarging the set of successor states, i.e.,

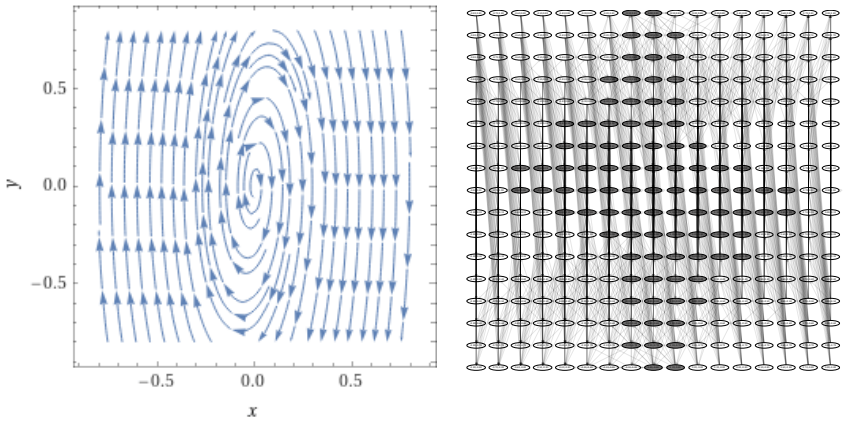
$$\text{avg.} \left\|A_{\hat{x},\hat{u}} - B_{\hat{x},\hat{u}} K_{\hat{x},\hat{u},\hat{x}'}/\varepsilon_{\hat{x}}\right\|_\infty = 1.21 > 1.$$

The expansion of the sets of successor states in some transitions is balanced by a number of contractive transitions. In consequence, the precision $\hat{\varepsilon}$ varies between 0.05 and 0.069. Because the closed-loop system is less contractive compared to the inverted pendulum, the average out-degree is also lower, with only two possible successor states to choose from.

The graphs of the obtained abstractions are shown in Fig. 1 and 2. The greyed nodes represent the strongly connected components (SSCs). The fact that the SSCs include a significant portion of the abstract states demonstrates that our abstraction provides a certain degree of completeness even for the chosen coarse grid. We are interested in computing strongly connected components since, for deterministic abstractions, they represent a set of states where all actions are reversible (they can be undone), and any state may be revisited an infinite number of times. This is particularly important when designing supervisors for a reinforcement learning agent [19], which is a goal for future research.

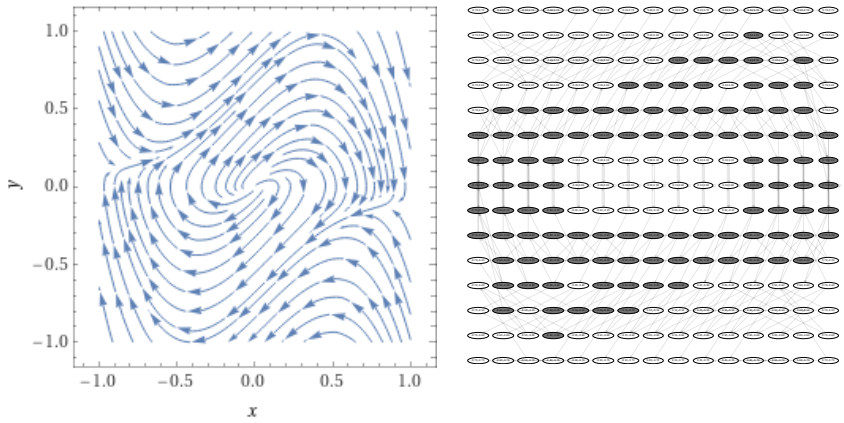
6 Conclusions

The presented method for constructing discrete abstractions for discrete-time, continuous-state systems provides a trade-off between accuracy, completeness,



(a) continuous-time system for $u = 0$ (b) discrete abstraction

Fig. 1. Streamline plot of a continuous-time version of the concrete system and a discrete abstraction of the inverted pendulum, with strongly connected components indicated in dark grey (node size not to scale)



(a) continuous system for $u = 0$ (b) discrete abstraction

Fig. 2. Streamline plot of a continuous-time version of the concrete system and a discrete abstraction of the Van der Pol oscillator, with strongly connected components indicated in dark grey (node size not to scale)

and computational feasibility. By relaxing the stringent completeness requirements of traditional discrete bisimulations, this approach produces deterministic and robust solutions. By leveraging linearization and linear feedback control, we make the approach applicable to non-contractive nonlinear systems. Numerical experiments with classic nonlinear systems illustrate the practicality of this

method. The proposed deterministic abstractions provide strong guarantees of system behavior, making them particularly useful for applications requiring high levels of safety and robustness. Numerical experiments with unstable nonlinear systems illustrate the practicality of this method. In future research, we will explore different types of properties that can be guaranteed by adapting these abstractions online, such as preserving safety and reversibility in the presence of dynamic obstacles.

Acknowledgments. This work was supported by the Traits project, under the French National Research Agency (ANR) grant number ANR-21-FAI1-0005.

References

1. Althoff, M., Grebenyuk, D., Kochdumper, N.: Implementation of taylor models in cora 2018. In: Proc. of the 5th International Workshop on Applied Verification for Continuous and Hybrid Systems (2018)
2. Alur, R., Henzinger, T.A., Kupferman, O., Vardi, M.Y.: Alternating refinement relations. In: CONCUR'98 Concurrency Theory: 9th International Conference Nice, France, September 8–11, 1998 Proceedings 9. pp. 163–178. Springer (1998)
3. Bellitto, T., Bergougnoux, B.: On minimum connecting transition sets in graphs. In: Graph-Theoretic Concepts in Computer Science: 44th International Workshop, WG 2018, Cottbus, Germany, June 27–29, 2018, Proceedings 44. pp. 40–51. Springer (2018)
4. Belta, C., Yordanov, B., Gol, E.A.: Formal Methods for Discrete-Time Dynamical Systems. Springer (2017)
5. Berz, M., Hoffstätter, G.: Computation and application of taylor polynomials with interval remainder bounds. Reliable Computing **4**(1), 83–97 (1998)
6. Chomachar, S.A., Fard, A.M.: Flight control system for guided rolling-airframe missile. In: 2016 IEEE Aerospace Conference. pp. 1–9 (2016)
7. De Nicola, R., Vaandrager, F.: Action versus state based logics for transition systems. In: LITP Spring School on Theoretical Computer Science, pp. 407–419. Springer (1990)
8. De Nicola, R., Vaandrager, F.: Three logics for branching bisimulation. Journal of the ACM (JACM) **42**(2), 458–487 (1995)
9. Demangeon, R., Dima, C., Varacca, D.: Observational preorders for alternating transition systems. In: European Conference on Multi-Agent Systems. pp. 312–327. Springer (2023)
10. Egidio, L.N., Lima, T.A., Jungers, R.M.: State-feedback abstractions for optimal control of piecewise-affine systems. In: 2022 IEEE 61st Conference on Decision and Control (CDC). pp. 7455–7460 (2022)
11. Girard, A., Pappas, G.J.: Approximation metrics for discrete and continuous systems. IEEE Transactions on Automatic Control **52**(5), 782–798 (2007)
12. Girard, A., Pappas, G.J.: Approximate bisimulation: A bridge between computer science and control theory. European Journal of Control **17**(5-6), 568–578 (2011)
13. Girard, A., Pola, G., Tabuada, P.: Approximately bisimilar symbolic models for incrementally stable switched systems. IEEE Transactions on Automatic Control **55**(1), 116–126 (2010). https://doi.org/10.1109/TAC.2009.2034922
14. Khuller, S., Raghavachari, B., Young, N.: Approximating the minimum equivalent digraph. SIAM Journal on Computing **24**(4), 859–872 (1995)

15. Korda, M., Mezić, I.: Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica **93**, 149–160 (2018)
16. Liu, J., Ozay, N.: Finite abstractions with robustness margins for temporal logic-based control synthesis. Nonlinear Analysis: Hybrid Systems **22**, 1–15 (2016)
17. Milner, R.: An algebraic definition of simulation between programs. Citeseer (1971)
18. Mitchell, I.M.: Comparing forward and backward reachability as tools for safety analysis. In: International Workshop on Hybrid Systems: Computation and Control. pp. 428–443. Springer (2007)
19. Moldovan, T.M., Abbeel, P.: Safe exploration in markov decision processes. In: Proc. Int. Conf. Machine Learning. p. 1451–1458. ICML'12, Omnipress, Madison, WI, USA (2012)
20. Pola, G., Tabuada, P.: Symbolic models for nonlinear control systems: Alternating approximate bisimulations. SIAM Journal on Control and Optimization **48**(2), 719–733 (2009)
21. Reissig, G., Weber, A., Rungger, M.: Feedback refinement relations for the synthesis of symbolic controllers. IEEE Transactions on Automatic Control **62**(4), 1781–1796 (2016)
22. Van der Schaft, A.: Equivalence of dynamical systems by bisimulation. IEEE transactions on automatic control **49**(12), 2160–2172 (2004)
23. Tabuada, P.: Verification and control of hybrid systems: a symbolic approach. Springer Science & Business Media (2009)
24. Tabuada, P., Pappas, G.J.: Finite bisimulations of controllable linear systems. In: 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475). vol. 1, pp. 634–639. IEEE (2003)
25. Tajvar, P., Meyer, P.J., Tumova, J.: Closed-loop incremental stability for efficient symbolic control of non-linear systems. IFAC-PapersOnLine **54**(5), 121–126 (2021). https://doi.org/https://doi.org/10.1016/j.ifacol.2021.08.485, https://www.sciencedirect.com/science/article/pii/S240589632101260X, 7th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2021
26. Yang, L., Zhang, H., Jeannin, J.B., Ozay, N.: Efficient backward reachability using the minkowski difference of constrained zonotopes. Trans. Comp.-Aided Des. Integ. Cir. Sys. **41**(11), 3969–3980 (nov 2022). https://doi.org/10.1109/TCAD.2022.3197971, https://doi.org/10.1109/TCAD.2022.3197971