Robust Abstractions for Control Synthesis: Completeness via Robustness for Linear-Time Properties

Jun Liu
Department of Applied Mathematics
University of Waterloo
j.liu@uwaterloo.ca

ABSTRACT

We define robust abstractions for synthesizing provably correct and robust controllers for (possibly infinite) uncertain transition systems. It is shown that robust abstractions are sound in the sense that they preserve robust satisfaction of linear-time properties. We then focus on discrete-time control systems modelled by nonlinear difference equations with inputs and define concrete robust abstractions for them. While most abstraction techniques in the literature for nonlinear systems focus on constructing sound abstractions, we present computational procedures for constructing both sound and approximately complete robust abstractions for general nonlinear control systems without stability assumptions. Such procedures are approximately complete in the sense that, given a concrete discrete-time control system and an arbitrarily small perturbation of this system, there exists a finite transition system that robustly abstracts the concrete system and is abstracted by the slightly perturbed system simultaneously. A direct consequence of this result is that robust control synthesis for discrete-time nonlinear systems and linear-time specifications is robustly decidable. More specifically, if there exists a robust control strategy that realizes a given linear-time specification, we can algorithmically construct a (potentially less) robust control strategy that realizes the same specification. The theoretical results are illustrated with a simple motion planning example.

Keywords

Nonlinear systems; control synthesis; abstraction; robustness; linear-time property; linear temporal logic; decidability

1. INTRODUCTION

Abstraction serves as a bridge for connecting control theory and formal methods in the sense that hybrid control design for dynamical systems and high-level specifications can be done using finite abstractions of these systems [1, 23]. There has been a rich literature on computing abstractions for linear and nonlinear dynamical systems in the past decade (see, e.g., [10, 13–16, 18, 19, 25, 27]). Early work on abstraction focuses on constructing symbolic models that are bisimilar (equivalent) to the original system. The seminal work in [24] shows that bisimilar symbolic models exist for controllable linear systems. As a result, existence of controllers for such systems to meet linear-time properties (such as those specified by linear temporal logic [6]) is decidable. For nonlinear systems that are incrementally stable [3], it is shown in [18] that approximately bisimilar models can be constructed (see also [10], for construction of approximately bisimilar models for switched systems, and [9] for its use in control synthesis). The work in [27] considered symbolic models for nonlinear systems without stability assumptions, in which it is shown that symbolic models that approximately alternatingly simulate the sample-data representation of a general nonlinear control system can be constructed. The work in [19] and [25] both proposes computational procedures for constructing finite abstractions of discrete-time nonlinear systems. The abstraction techniques in [19, 25, 27] are conservative and sound in the sense that they are useful in the design of provably correct controllers, but do not necessarily yield a feasible design because the computational procedures for constructing abstractions for potentially unstable nonlinear systems are not complete.

Robustness is a central property to consider in control design, because all practical control systems need to be robust to imperfections in all aspects of control design and implementation, such as modelling, sensing, computation, communication, and actuation. For abstraction-based control design, how to preserve robustness poses a particular challenge because the hierarchical control design approach based on abstraction often use quantized state measurements (modelled as symbolic states in the abstraction) to compute appropriate control signals. Because of the state quantizers by definition are discontinuous, special attention is required to ensure that the resulting design is actually robust to measurement errors and disturbances. The work in [14] (see also [15]) proposes a novel notion of abstractions that are equipped with additional robustness margins to cope with different types of uncertainties in modelling, such as measurement errors, delays, and disturbances. The work in [20] (see also [21]) defines a new notation of system relations for abstraction-based control design. By explicitly considering the interconnection of state quantizers and feedback controllers, it is shown that the new system relation can

HSCC'17, April 18 - 20, 2017, Pittsburgh, PA, USA

ISBN 978-1-4503-4590-3/17/04…$15.00

DOI: http://dx.doi.org/10.1145/3049797.3054970

also be used to design robust controllers against uncertainties and disturbances. The type of abstractions considered in [14, 15, 20, 21] resemble the approximate alternating simulations considered in [27] for nonlinear systems. These abstractions, nonetheless, are all conservative and sound. To the best knowledge of the authors, how to compute complete abstractions (or approximately complete) abstractions for general nonlinear systems without stability assumptions remains an open problem.

As an attempt to bridge this gap, in this paper, we define robust abstractions as a system relation from a (possibly infinite) transition system subject to uncertainty to anther transition system. We show that, while this abstraction relation is to some extent similar to the type of system relations considered in [15, 21, 27], it also has some subtle differences that are important for proving the approximate completeness results later in the paper. We show that robust abstractions are sound in the sense that they preserve robust satisfaction of linear-time properties. The main contributions of the paper include computational procedures for constructing both sound and approximately complete robust abstractions for general discrete-time nonlinear control systems without stability assumptions. We show that such procedures are complete in the sense that, given a concrete discrete-time control system and an arbitrarily small perturbation of this system, there exists a finite transition system that robustly abstracts the concrete system, whereas the perturbed system abstracts this finite transition system. An important consequence of this main result asserts that existence of robust controllers for discrete-time nonlinear systems and linear-time specifications is decidable. Finally, we would like to make clear upfront that the main point of this paper is not on providing more efficient algorithms for computing abstractions. Therefore, complexity issues, though important, are not a concern for the current paper and will be investigated in future work. promises...

The organization of the paper is very straightforward. Section 2 presents some background material on transition systems and define robust abstractions. We highlight some similarities and subtle differences of the new abstraction relation with several variants of simulation relations in the literature. Section 3 presents the main results of the paper on construction of sound and approximately complete robust abstractions for discrete-time nonlinear control systems. A numerical example is used to illustrate the effectiveness of robust abstractions in Section 4. The paper is concluded in Section 5.

Notation: Let f be a (binary) relation from A to B, i.e., f is a subset of the Cartesian product $A \times B$. For each $a \in A$, $f(a)$ denotes the set $\{b : b \in B \text{ such that } (a,b) \in f\}$; for each $b \in B$, $f^{-1}(b)$ denotes the set $\{a : a \in A, (a,b) \in f\}$; for $A' \subseteq A$, $f(A') = \cup_{a \in A'} f(a)$; and for $B' \subseteq B$, $f^{-1}(B) = \cup_{b \in B'} f^{-1}(B)$. Let g be a relation from A to B and f be a relation from B to C. The composition of f and g, denoted by $f \circ g$, is a relation from A to C defined by

$$f \circ g = \{(a,c) : \exists b \in B \text{ s.t. } (a,b) \in g \text{ and } (b,c) \in f\}.$$

For two sets $A, B \subseteq \mathbb{R}^n$,

$$A + B = \{c : \exists a \in A, \exists b \in B \text{ s.t. } a + b = c\}$$

and $A \backslash B = \{a : a \in A, a \notin B\}$. For $a \in \mathbb{R}^n$ and $B \subseteq \mathbb{R}^n$, $a + B = \{a\} + B$. Let $|\cdot|$ denote the infinity norm in $\mathbb{R}^n$ and $\mathbb{B}$ denote the unit closed ball in infinity norm centred at the origin, i.e. $\mathbb{B} = \{x \in \mathbb{R}^n : |x| \leq 1\}$. The dimension of $\mathbb{B}$ will be clear from the context.

2. TRANSITION SYSTEMS AND ROBUST ABSTRACTIONS

2.1 Transition systems

DEFINITION 1. A *transition system* is a tuple

$$\mathcal{T} = (Q, A, R, \Pi, L),$$

where

- Q is the set of states;
- A is the set of actions;
- $R \subseteq Q \times A \times Q$ is the transition relation;
- Π is the set of atomic propositions;
- $L : Q \to 2^{\Pi}$ is the labelling function.

Consider the transition system $\mathcal{T}$ above. For each action $a \in A$ and $q \in Q$, the a-successor of q, denoted by $\mathrm{Post}_{\mathcal{T}}(q,a)$, is defined by

$$\mathrm{Post}_{\mathcal{T}}(q,a) = \{q' : q' \in Q \text{ s.t. } (q,a,q') \in R\}.$$

For each $q \in Q$, the set of admissible actions for q, denoted by $A_{\mathcal{T}}(q)$, is defined by

$$A_{\mathcal{T}}(q) = \{a : \mathrm{Post}_{\mathcal{T}}(q,a) \neq \emptyset\}.$$

In this paper, we assume that all transition systems have no terminal states in the sense that $A_{\mathcal{T}}(q) \neq \emptyset$ for all $q \in Q$.

An *execution* of $\mathcal{T}$ is an infinite alternating sequence of states and actions

$$\rho = q_0 a_0 q_1 a_1 q_2 a_2 \cdots,$$

where q_0 is some initial state and $(q_i, a_i, q_{i+1}) \in R$ for all $i \geq 0$. The *path* resulting from the execution ρ above is

$$\mathrm{Path}(\rho) = q_0 q_1 q_2 \cdots.$$

The *trace* of the execution ρ is defined by

$$\mathrm{Trace}(\rho) = L(q_0)L(q_1)L(q_2)\cdots.$$

A *control strategy* for a transition system $\mathcal{T}$ is a partial function $s : (q_0, q_1, \cdots, q_i) \mapsto a_i$ that maps the state history to the next action. An *s-controlled execution* of a transition system $\mathcal{T}$ is an execution of $\mathcal{T}$, where for each $i \geq 0$, the action a_i is chosen according to the control strategy s; s-controlled paths and traces are defined in a similar fashion.

2.2 Uncertainty transition systems

Δ can be $\emptyset$

DEFINITION 2. A transition relation $\Delta \subseteq Q \times A \times Q$ is called an *uncertain transition relation* for $\mathcal{T} = (Q, A, R, \Pi, L)$, if the following two conditions hold:

(i) $R \cap \Delta = \emptyset$;

(ii) for each $(q,a,q') \in \Delta$, there exists some $(q,a,q'') \in R$.

where necessarily $q'' \neq q'$

DEFINITION 3. An *uncertain transition system* consisting of $\mathcal{T} = (Q, A, R, \Pi, L)$ as a nominal transition system and Δ as an uncertain transition relation for $\mathcal{T}$, denoted by $\mathcal{T} \oplus \Delta$, is defined by

$$\mathcal{T} \oplus \Delta = (Q, A, R \cup \Delta, \Pi, L).$$

It is clear from the above definition that, while Δ introduces additional transitions for the transition system $\mathcal{T}$, it does not add more admissible actions for any state. In other words, for all $q \in Q$, $A_{\mathcal{T}}(q) = A_{\mathcal{T}\oplus\Delta}(q)$.

Since an uncertain transition system is simply a transition system with additional transitions introduced by some uncertain transition relation, the execution (path, trace), control strategy, and controlled execution (path, trace) for an uncertain transition system are defined in the same way as for a nominal transition system.

2.3 Robust abstractions

We first define a notion of abstraction between transition systems for control synthesis.

DEFINITION 4. For two transition systems

$$\mathcal{T}_1 = (Q_1, A_1, R_1, \Pi, L_1)$$

and

$$\mathcal{T}_2 = (Q_2, A_2, R_2, \Pi, L_2),$$

a relation $\alpha \subseteq Q_1 \times Q_2$ is said to be an *abstraction from $\mathcal{T}_1$ to $\mathcal{T}_2$* if the following conditions are satisfied:

(i) for all $q_1 \in Q_1$, there exists $q_2 \in Q_2$ such that $(q_1, q_2) \in \alpha$ (i.e., $\alpha(q_1) \neq \emptyset$);

(ii) for all $(q_1, q_2) \in \alpha$ and $a_2 \in A_{\mathcal{T}_2}(q_2)$, there exists $a_1 \in A_{\mathcal{T}_1}(q_1)$ such that

$$\alpha(\mathrm{Post}_{\mathcal{T}_1}(q, a_1)) \subseteq \mathrm{Post}_{\mathcal{T}_2}(q_2, a_2); \quad (1)$$

for all $q \in \alpha^{-1}(q_2)$;

(iii) for all $(q_1, q_2) \in \alpha$, $L_2(q_2) \subseteq L_1(q_1)$.

If such a relation α exists, we say that $\mathcal{T}_2$ *abstracts* $\mathcal{T}_1$ and write $\mathcal{T}_1 \preceq_\alpha \mathcal{T}_2$ or simply $\mathcal{T}_1 \preceq \mathcal{T}_2$.

We then define robust abstractions as abstractions of uncertain transition systems.

DEFINITION 5. Let Δ be an uncertain transition relation for $\mathcal{T}_1$. If there exists an abstraction α from $\mathcal{T}_1 \oplus \Delta$ to $\mathcal{T}_2$, i.e., $\mathcal{T}_1 \oplus \Delta \preceq_\alpha \mathcal{T}_2$, we say that α is a Δ-robust abstraction from $\mathcal{T}_1$ to $\mathcal{T}_2$ and $\mathcal{T}_2$ *Δ-robustly abstracts* $\mathcal{T}_1$. With a slight abuse of terminology, we sometimes also say that $\mathcal{T}_2$ is a Δ-robust abstraction of $\mathcal{T}_1$.

REMARK 1. We highlight several differences between the notation of abstraction proposed in Definition 4 and other similar system relations in the literature. Apart from the obvious distinction that, in Definition 4, an explicit model of the uncertainty is considered (following [26]), the abstraction defined by Definition 4 differs from several variants of simulation relations in the literature as elaborated below:

Finite abstractions with robustness margins: This notion of abstractions introduced in [14, 15] is defined by introducing two positive parameters (γ_1, γ_2), which define the extra transitions to be added to the abstractions to ensure robustness. Suppose there is a metric d defined on Q_1. Then finite abstractions with robustness margins (γ_1, γ_2) amount to defining

$$\Delta = \{(q, a, q') : \exists (q_1, a, q_1') \in R_1 \text{ s.t. } d(q_1, q) \leq \gamma_1,\ d(q_1', q') \leq \gamma_2\} \backslash R_1.$$

To establish $\mathcal{T}_1 \oplus \Delta \preceq_\alpha \mathcal{T}_2$, condition (1), which can be equivalently written as

$$\bigcup_{q \in \alpha^{-1}(q_2)} \alpha(\mathrm{Post}_{\mathcal{T}_1 \oplus \Delta}(q, a_1)) = \alpha\Big(\bigcup_{q \in \alpha^{-1}(q_2)} \mathrm{Post}_{\mathcal{T}_1 \oplus \Delta}(q, a_1)\Big) \subseteq \mathrm{Post}_{\mathcal{T}_2}(q_2, a_2)$$

is essentially the over-approximation (of transitions) condition in [14, 15]. The main difference lies in that Definition 4 does not assume that a metric is defined on Q_1 and the uncertainty model is not restricted to that defined by level sets of the distance function. Furthermore, here we define the abstraction relation on a general Kripke structure, whereas the work in [14, 15] defines concrete abstractions from ordinary differential/difference equations with inputs to finite transition systems.

Feedback refinement relations [20, 21]: Similar to [14, 15], the abstraction relation considered in [20, 21] also requires that, for each $(q_1, q_2) \in \alpha$, the admissible actions for each q_2 is a subset of the admissible actions for q_1. In Definition 4, for each $(q_1, q_2) \in \alpha$, it is not required that $A_{\mathcal{T}_2}(q_2) \subseteq A_{\mathcal{T}_1}(q_1)$, i.e., the admissible actions for q_1 do not have to be a subset of the admissible actions for q_2. This difference enables us to formulate and prove the approximate completeness results later in this paper (Section 3.3). Note that, when $A_{\mathcal{T}_2}(q_2) \subseteq A_{\mathcal{T}_1}(q_1)$, condition (1) can be simplified to: for each $(q_1, q_2) \in \alpha$ and every $a \in A_{\mathcal{T}_2}(q_2)$,

$$\alpha(\mathrm{Post}_{\mathcal{T}_1}(q_1, a)) \subseteq \mathrm{Post}_{\mathcal{T}_2}(q_2, a). \quad (2)$$

In other words, the same action a used by q_2 is assumed to be available (and used) for all $q_1 \in \alpha^{-1}(q_2)$, because $A_{\mathcal{T}_2}(q_2) \subseteq A_{\mathcal{T}_1}(q_1)$.

Alternating simulations [18, 27]: The notion of alternating simulations [18, 27] stipulates that, for each $(q_1, q_2) \in \alpha$ and every $a_2 \in A_{\mathcal{T}_2}(q_2)$, there exists $a_1 \in A_{\mathcal{T}_2}(q_1)$ such that, for every $q_1' \in \mathrm{Post}_{\mathcal{T}_1}(q_1, a_1)$, there exists some state $q_2' \in \mathrm{Post}_{\mathcal{T}_2}(q_2, a_2)$ such that $(q_1', q_2') \in \alpha$. In other words, for each $(q_1, q_2) \in \alpha$ and every $a_2 \in A_{\mathcal{T}_2}(q_2)$, there exists $a_1 \in A_{\mathcal{T}_1}(q_1)$ such that

$$\alpha(q_1') \cap \mathrm{Post}_{\mathcal{T}_2}(q_2, a_2) \neq \emptyset, \quad (3)$$

for all $q_1' \in \mathrm{Post}_{\mathcal{T}_1}(q_1, a_1)$, as articulated in [20, 21]. Clearly, (3) is a weaker condition than (1) or (2), unless α is single-valued. Furthermore, and more importantly, (3) does not stipulate the use of the same action a_1 for all $q \in \alpha^{-1}(q_2)$, i.e., a_1 may depend on q (concrete states corresponding to q_2). A consequence of the latter is that, to implement the controller, one needs knowledge of the concrete state rather than the abstract (symbolic) state alone.

We use a simple example to illustrate the differences discussed above.

EXAMPLE 1. Consider three transition systems

$$\mathcal{T}_i = (Q_i, A_i, R_i, \Pi, L_i), \quad i = 1, 2, 3,$$

where $Q_1 = \{x_0, x_1, x_2\}$, $Q_2 = Q_3 = \{q_0, q_1\}$, $A_1 = \{a, b\}$, $A_2 = A_3 = \{1, 2, 3\}$, $\Pi = \{\text{Initial}, \text{Goal}\}$, $L_1(x_0) = L_1(x_1) = L_2(q_0) = L_3(q_0) = \{\text{Initial}\}$, and $L_1(x_2) = L_2(q_1) = L_3(q_1) = \{\text{Goal}\}$. The transition relations are shown in Figure 1.

Define an abstraction relation from $\mathcal{T}_1$ to $\mathcal{T}_2$ by

$$\alpha = \{(x_0, q_0), (x_1, q_0), (x_2, q_1)\}.$$

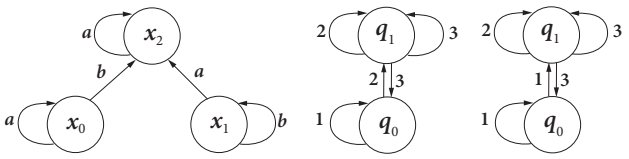

Figure 1: Transition systems $\mathcal{T}_1$ (left), $\mathcal{T}_2$ (middle), and $\mathcal{T}_3$ (right).

Then it can be easily verified that (3) is satisfied and α is an alternating simulation from $\mathcal{T}_1$ to $\mathcal{T}_2$. In fact, we can check that, for $(x_0, q_0) \in \alpha$, and action $1 \in A_2$, there exists $a \in A_1$ such that

$$\alpha(\mathrm{Post}_{\mathcal{T}_1}(x_0, a)) = \alpha(\{x_0\}) = \{q_0\} = \mathrm{Post}_{\mathcal{T}_2}(q_0, 1),$$

which implies (3). Similarly, for $(x_0, q_0) \in \alpha$, and action $2 \in A_2$, there exists $b \in A_1$ such that

$$\alpha(\mathrm{Post}_{\mathcal{T}_1}(x_0, b)) = \alpha(\{x_2\}) = \{q_1\} = \mathrm{Post}_{\mathcal{T}_2}(q_0, 2),$$

which also implies (3). For $(x_2, q_1) \in \alpha$, and action $3 \in A_2$, there exists $a \in A_1$ such that

$$\alpha(\mathrm{Post}_{\mathcal{T}_1}(x_2, a)) = \alpha(\{x_2\}) = \{q_1\} \subseteq \{q_0, q_1\} = \mathrm{Post}_{\mathcal{T}_2}(q_1, 3),$$

which implies (3). The rest can be checked in a similar fashion.

Suppose that one needs to design a control strategy for $\mathcal{T}_1$ such that all controlled executions of $\mathcal{T}_1$ starting from the 'Initial' set will eventually reach the 'Goal' set. Then, while one can find such a control strategy for $\mathcal{T}_2$, to implement this strategy on $\mathcal{T}_1$, however, $\mathcal{T}_1$ needs to be able to discriminate x_0 and x_1 and choose the appropriate actions (b for x_0 and a for x_1). This is not the case if only symbolic state information from the abstraction is available.

Note that, according to Definition 4, we do not have $\mathcal{T}_1 \preceq_\alpha \mathcal{T}_2$ because, for $(x_0, q_0) \in \alpha$ and action $1 \in A_2$, we have

$$\begin{aligned}\bigcup_{x \in \alpha^{-1}(q_0)} \alpha(\mathrm{Post}_{\mathcal{T}_1}(x, a)) &= \alpha(\{x_0\}, \{x_2\}) \\ &= \{q_0, q_1\} \not\subseteq \{q_0\} = \mathrm{Post}_{\mathcal{T}_2}(q_0, 1),\end{aligned}$$

$$\begin{aligned}\bigcup_{x \in \alpha^{-1}(q_0)} \alpha(\mathrm{Post}_{\mathcal{T}_1}(x, b)) &= \alpha(\{x_1\}, \{x_2\}) \\ &= \{q_0, q_1\} \not\subseteq \{q_0\} = \mathrm{Post}_{\mathcal{T}_2}(q_0, 1),\end{aligned}$$

Thus, (1) does not hold for either action a or b.

We can check that $\mathcal{T}_1 \preceq_\alpha \mathcal{T}_3$. Because the set of actions in $\mathcal{T}_2$ (and $\mathcal{T}_3$) is not a subset of the actions of $\mathcal{T}_1$ (in fact there are more actions in $\mathcal{T}_2$ and $\mathcal{T}_3$ than $\mathcal{T}_1$), α does not provide an abstraction relation from $\mathcal{T}_1$ to $\mathcal{T}_2$ or from $\mathcal{T}_1$ to $\mathcal{T}_3$ in the strict sense of the notions of simulation relations considered in [14, 15, 20, 21].

To consider a robust abstraction for $\mathcal{T}_1$, let $\Delta = \{(x_2, a, x_1)\}$. Then it can be verified that the transition system $\mathcal{T}_3$ is also a Δ-robust abstraction of $\mathcal{T}_1$.

We will state some immediate results that follow from Definition 4.

PROPOSITION 1. *Let $\mathcal{T}$ be a transition system and Δ be an uncertain transition relation for $\mathcal{T}$. Then $\mathcal{T} \preceq \mathcal{T} \oplus \Delta$.*

PROOF. Let $\mathcal{T} = (Q, A, R, \Pi, L)$. It is straightforward to check by Definitions 2 and 4 that the identity relation from Q to Q defines a Δ-robust abstraction from $\mathcal{T}$ to $\mathcal{T} \oplus \Delta$. □

Setting $\Delta = \emptyset$, a special case of Proposition 1 asserts that $\mathcal{T} \preceq \mathcal{T}$ for any transition system. It is also straightforward to verify that abstraction relations are transitive in the following sense.

PROPOSITION 2. *Let $\mathcal{T}_i$ $(i = 1, 2, 3)$ be transition systems and Δ be an uncertain transition relation for $\mathcal{T}_1$. If $\mathcal{T}_1 \preceq_{\alpha_1} \mathcal{T}_2$ and $\mathcal{T}_2 \preceq_{\alpha_2} \mathcal{T}_3$, then $\mathcal{T}_1 \preceq_{\alpha_2 \circ \alpha_1} \mathcal{T}_3$.*

PROOF. Let $\alpha_3 = \alpha_2 \circ \alpha_1$. We verify that conditions (i)–(iii) of Definition 4 are satisfied:

(i) For all $q_1 \in Q$, $\alpha_3(q_1)$ is non-empty, because $\alpha_1(q_1)$ is non-empty and $\alpha_2(q_2)$ is non-empty for any $q_2 \in Q_2$.

(ii) For any $(q_1, q_3) \in \alpha_3$, there exists $q_2 \in Q_2$ such that $(q_1, q_2) \in \alpha_1$ and $(q_2, q_3) \in \alpha_2$. For any $q_3 \in A_{\mathcal{T}_3}(q_3)$, there exists $a_2 \in A_{\mathcal{T}_2}(q_2)$ such that

$$\alpha_2(\mathrm{Post}_{\mathcal{T}_2}(q, a_2)) \subseteq \mathrm{Post}_{\mathcal{T}_3}(q_3, a_3),$$

for all $q \in \alpha_2^{-1}(q_3)$. For $a_2 \in A_{\mathcal{T}_2}(q_2)$, there exists $a_1 \in A_{\mathcal{T}_1}(q_1)$ such that

$$\alpha_1(\mathrm{Post}_{\mathcal{T}_1}(q, a_1)) \subseteq \mathrm{Post}_{\mathcal{T}_2}(q_2, a_2),$$

for all $q \in \alpha_1^{-1}(q_2)$. It follows that

$$\begin{aligned}&\bigcup_{q \in \alpha_3^{-1}(q_3)} \alpha_3(\mathrm{Post}_{\mathcal{T}_1}(q, a_1)) \\ &\quad = \bigcup_{q \in \alpha_3^{-1}(q_3)} \alpha_2 \circ \alpha_1(\mathrm{Post}_{\mathcal{T}_1}(q, a_1)) \\ &\quad \subseteq \bigcup_{q \in \alpha_2^{-1}(q_2)} \alpha_2(\mathrm{Post}_{\mathcal{T}_2}(q, a_2)) \\ &\quad \subseteq \mathrm{Post}_{\mathcal{T}_3}(q_3, a_3).\end{aligned}$$

(iii) For any $(q_1, q_3) \in \alpha_3$, there exists $q_2 \in Q_2$ such that $(q_1, q_2) \in \alpha_1$ and $(q_2, q_3) \in \alpha_2$. Hence

$$L_3(q_3) \subseteq L_2(q_2) \subseteq L_1(q_1).$$

□

2.4 Soundness of abstractions

In this section, we prove that abstractions given by Definition 4 are sound in the sense of preserving realizability of linear-time properties.

A *linear-time (LT) property* [6] over a set of atomic propositions Π is a subset of $(2^\Pi)^\omega$, which is the set of all infinite words over the alphabet 2^Π, defined by

$$(2^\Pi)^\omega = \left\{ A_0 A_1 A_2 \cdots : A_i \in 2^\Pi, \quad i \geq 0 \right\}.$$

A particular class of LT properties can be conveniently specified by *linear temporal logic* (LTL [17]). This logic consists of propositional logic operators (e.g., **true**, **false**, *negation* ($\neg$), *disjunction* ($\vee$), *conjunction* ($\wedge$) and *implication* ($\rightarrow$)), and temporal operators (e.g., *next* ($\bigcirc$), *always* ($\Box$), *eventually* ($\Diamond$), *until* ($\mathcal{U}$) and *weak until* ($\mathcal{W}$)).

The syntax of LTL over a set of atomic propositions Π is defined inductively follows:

- **true** and **false** are LTL formulae;

why introducing general negation and then disallow it?

- an atomic proposition $\pi \in \Pi$ is an LTL formula;
- if φ and ψ are LTL formulas, then $\neg\varphi$, $\varphi \vee \varphi$, $\bigcirc\varphi$, and $\varphi \mathcal{U} \varphi$ are LTL formulas.

The semantics of LTL is defined on infinite words over the alphabet 2^{Π}. Given a sequence $\sigma = A_0 A_1 A_2 \cdots$ in 2^{Π}, we define $\sigma, i \vDash \varphi$, meaning that σ satisfies an LTL formula φ at position i, inductively as follows:

- $\sigma, i \vDash \mathbf{true}$;
- $\sigma, i \vDash \pi$ if and only if $\pi \in A_i$;
- $\sigma, i \vDash \neg\varphi$ if and only if $\sigma, i \nvDash \varphi$;
- $\sigma, i \vDash \varphi_1 \vee \varphi_2$ if and only if $\sigma, i \vDash \varphi_1$ or $\sigma, i \vDash \varphi_2$;
- $\sigma, i \vDash \bigcirc\varphi$ if and only if $\sigma, i+1 \vDash \varphi$;
- $\sigma, i \vDash \varphi_1 \mathcal{U} \varphi_2$ if and only if there exists $j \geq i$ such that $\sigma, j \vDash \varphi_2$ and $\sigma, k \vDash \varphi_1$ for all $i \leq k < j$;

We write $\sigma \vDash \varphi$, and say σ satisfies φ, if $\sigma, 0 \vDash \varphi$. An execution ρ of a transition system $\mathcal{T}$ is said to satisfy an LTL formula φ, written as $\rho \vDash \varphi$, if and only if its trace $\mathrm{Trace}(\rho) \vDash \varphi$. Given a control strategy s for $\mathcal{T}$, if all s-controlled executions of $\mathcal{T}$ satisfy φ, we write $(\mathcal{T}, s) \vDash \varphi$. If such a control strategy s exists, we also say that φ is *realizable* for $\mathcal{T}$.

not full LTL!

Remark 1. For technical reasons, we assume that all LTL formulas have been transformed into positive normal form [6, Chapter 5], where all negations appear only in front of the atomic propositions and only the following operators are allowed $\wedge$, $\vee$, $\bigcirc$, $\mathcal{U}$, and $\mathcal{W}$ (defined by $\varphi \mathcal{W} \psi = (\varphi \mathcal{U} \psi) \vee \Box\varphi$. We further assume that all negations of atomic propositions are replaced by new atomic propositions. !!

Definition 6. Given an abstraction relation α from $\mathcal{T}_1$ to $\mathcal{T}_2$ and a control strategy μ_i for $\mathcal{T}_i$ $(i = 1, 2)$, μ_1 is called *α-implementation* of μ_2, if, for each $n \geq 0$,

the

$$u_n = \mu_1(x_0, x_1, x_2, \cdots, x_n)$$

is chosen according to

$$a_n = \mu_2(q_0, q_1, q_2, \cdots, q_n)$$

in such a way (as guaranteed by Definition 4 for $\mathcal{T}_1 \preceq_\alpha \mathcal{T}_2$) that

$$\alpha(\mathrm{Post}_{\mathcal{T}_1}(x, u_n)) \subseteq \mathrm{Post}_{\mathcal{T}_2}(q_n, a_n)$$

for all $x \in \alpha^{-1}(q_n)$, where $q_n \in \alpha(x_n)$.

We end this section by stating a soundness result for abstractions.

Theorem 1. *Suppose that α is an abstraction from $\mathcal{T}_1$ to $\mathcal{T}_2$, i.e., $\mathcal{T}_1 \preceq_\alpha \mathcal{T}_2$ and and let φ be an LTL formula. If there exists a control strategy μ_2 for $\mathcal{T}_2$ such that $(\mathcal{T}_2, \mu_2) \vDash \varphi$, then there exists a control strategy μ_1, which is an α-implementation of μ_2, for $\mathcal{T}_1$ such that $(\mathcal{T}_1, \mu_1) \vDash \varphi$.*

Proof. Let

$$\mathcal{T}_1 = (Q_1, A_1, R_1, \Pi, L_1)$$

and

$$\mathcal{T}_2 = (Q_2, A_2, R_2, \Pi, L_2).$$

We show that, by Definitions 4 and 6, a μ_1-controlled path of $\mathcal{T}_1$ always leads to a μ_2-controlled path of $\mathcal{T}_2$. Suppose we start with $x_k \in Q_1$ and let q_k be arbitrarily chosen from $\alpha(x_k)$, where $k \geq 0$. Suppose $a_k = \mu_2(q_0, q_1, q_2, \cdots, q_k)$ and $u_k = \mu_1(x_0, x_1, x_2, \cdots, x_k)$. Since $\alpha(\mathrm{Post}_{\mathcal{T}_1}(x_k, u_k)) \subseteq \mathrm{Post}_{\mathcal{T}_2}(q_k, a_k)$, we know that for any $q_{k+1} \in \alpha(x_{k+1})$ and $x_{k+1} \in \mathrm{Post}_{\mathcal{T}_1}(x_k, u_k)$, we have $q_{k+1} \in \mathrm{Post}_{\mathcal{T}_2}(q_k, a_k)$. This implies that (q_k, a_k, q_{k+1}) is a valid transition in $\mathcal{T}_2$ and therefore, by induction, $q_0 q_1 q_2 \cdots$ is a μ_2-controlled path of $\mathcal{T}_2$, if $x_0 x_1 x_2 \cdots$ is a μ_1-controlled path of $\mathcal{T}_1$. Furthermore, by Definitions 4, we have $L_2(q_k) \subseteq L_1(x_k)$ for all $k \geq 0$. Since the trace of $q_0 q_1 q_2 \cdots$ satisfies φ, we know that the trace of $x_0, x_1, x_2 \cdots$ also satisfies φ. □

aren't we supposed to define mu1??

Based on the proof, it is clear that an abstraction relation preserves not only temporal logic specifications but also linear-time properties in general, because we essentially proved that the controlled traces of $\mathcal{T}_1$ are included in the controlled traces of $\mathcal{T}_2$ (in fact, trace inclusion is equivalent to preservation of LT properties [6, Theorem 3.15]).

3. ROBUST DECIDABILITY OF DISCRETE-TIME CONTROL SYNTHESIS

In this section, we investigate robust abstractions of discrete-time nonlinear systems modelled by nonlinear difference equations with inputs. We establish computational procedures for constructing sound and approximately complete robust abstractions for this class of control systems under very mild conditions.

3.1 Perturbed discrete-time control systems as uncertain transition systems

A *discrete-time control system* is modelled by a difference equation of the form

$$x(t+1) = f(x(t), u(t)), \tag{4}$$

where $x(t) \in X \subseteq \mathbb{R}^n$, $u(t) \in U \subseteq \mathbb{R}^m$, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.

A *solution* to (4) is an alternating sequence of states and control inputs of the form

$$x(0)u(0)x(1)u(1)x(2)u(2)\cdots,$$

such that (4) is satisfied.

A *control strategy* for (4) is a partial function

$$\sigma : (x(0), \cdots, x(t)) \mapsto u(t)$$

for all $t = 0, 1, 2, \cdots$, which maps the state history up to time t to the control input $u(t)$ at time t.

Definition 7. The discrete-time control system (4) can be written as a transition system of the form

$$\mathcal{S} = (Q_{\mathcal{S}}, A_{\mathcal{S}}, R_{\mathcal{S}}, \Pi, L_{\mathcal{S}}) \tag{5}$$

by defining ← sink

- $Q_{\mathcal{S}} = X \cup \{X^c\}$;
- $A_{\mathcal{S}} = U$;
- $(x, u, x') \in R_{\mathcal{S}}$ if and only if one of the following holds: (i) $x' = f(x, u)$ and $x, x' \in X$; (ii) $x' = X^c$ and $f(x, u) \notin X$; (iii) $x' = x = X^c$;
- Π is a set of atomic propositions on $Q_{\mathcal{S}}$ and $\mathbf{in} \in \Pi$;
- $L_{\mathcal{S}} : Q_{\mathcal{S}} \to 2^{\Pi}$ is a labelling function satisfying $\mathbf{in} \in L_{\mathcal{S}}(q)$ for $q \neq X^c$ and $\mathbf{in} \notin L_{\mathcal{S}}(X^c)$.

The state X^c and label $\mathbf{in}$ are introduced to precisely encode if an out-of-domain transition takes place.

We now introduce an uncertainty model for system (4).

Definition 8. Consider system (4) subject to uncertainties of the form

$$x(t+1) = f(x(t), u(t)) + w(t), \tag{6}$$

where $w(t) \in \delta\mathbb{B}$ for some $\delta \geq 0$. Define Δ_δ to consist of transitions $(x, u, x') \notin R_{\mathcal{S}}$ such that one of the following holds: (i) $x' \in f(x, u) + \delta\mathbb{B}$ and $x, x' \in X$; (ii) $x' = X^c$ and $f(x, u) + w \notin X$ for some $w \in \delta\mathbb{B}$.

Clearly, $\mathcal{S} \oplus \Delta_\delta$ defined together by Definitions 7 and 8 exactly models (6) as summarized in the following proposition.

Proposition 3. *Each solution of (6) that stays in X is an execution of $\mathcal{S} \oplus \Delta_\delta$. Conversely, each execution of $\mathcal{S} \oplus \Delta_\delta$ that stays in X is also a solution of (6).*

Proof. This is straightforward to verify. Denote

$$\rho = x(0)u(0)x(1)u(1)x(2)u(2)\cdots.$$

If ρ is a solution of (6) such that $x(t) \in X$ for all $t \geq 0$. Then there exists $w(0)w(1)\cdots$ such that $x(t+1) = f(x(t), u(t)) + w(t)$, where $w(t) \in \delta\mathbb{B}$ for all $t \geq 0$, which implies that $(x(t), u(t), x(t+1)) \in R_{\mathcal{S}} \cup \Delta_\delta$. Thus ρ is also an execution of $\mathcal{S} \oplus \Delta_\delta$. Now suppose that ρ is an execution of $\mathcal{S} \oplus \Delta_\delta$ such that $x(t) \in X$ for all $t \geq 0$. Then $x(t+1) = f(x(t), u(t)) + w(t)$, where $w(t) \in \delta\mathbb{B}$ for all $t \geq 0$. This shows that ρ is a solution of (6). □

Because of this proposition, in the sequel, when proving soundness results, we always assume that out-of-domain solutions and paths are taken care of by enforcing the solutions and paths to stay in the domain through a safety specification, i.e., by including $\Box(\mathbf{in})$ in the specification.

3.2 Soundness of robust abstractions for discrete-time control systems

Corollary 1. *Suppose there exists a transition system $\mathcal{T}$ such that $\mathcal{S} \oplus \Delta_\delta \preceq_\alpha \mathcal{T}$, where $\mathcal{S}$ and Δ_δ are defined by Definitions 7 and 8. Let φ be an LTL formula over Π. If there exists a control strategy μ for $\mathcal{T}$ such that $(\mathcal{T}, \mu) \models \varphi$, then there exists a control strategy κ, which is an α-implementation of μ, for $\mathcal{S} \oplus \Delta_\delta$ such that $(\mathcal{S} \oplus \Delta_\delta, \kappa) \models \varphi$.*

Proof. It follows directly from Theorem 1. □

It is interesting to note that $(\mathcal{S} \oplus \Delta_\delta, \kappa) \models \varphi$ implies that solutions of (4) robustly satisfy φ in terms of not only additive disturbances modelled by (6), but also other types of uncertainties such as measurement errors. To illustrate this, consider a scenario where the controller κ is implemented on a system with measurement errors. We assume that this error is bounded, i.e., for each $x(t) \in \mathbb{R}^n$, its measurement is given by

$$\hat{x}(t) = x(t) + e(t), \tag{7}$$

where $e(t) \in \varepsilon\mathbb{B}$ for some $\varepsilon > 0$. To make the control strategy κ for (4) robust to measurement errors like (7), we can simply strengthen the labeling function L of $\mathcal{S}$ as follows. A labelling function $\hat{L} : \mathbb{R}^n \to 2^\Pi$ is said to be *the ε-strengthening* of another labelling function $L : \mathbb{R}^n \to 2^\Pi$, if $\pi \in \hat{L}(x)$ if and only if $\pi \in L(y)$ for all $y \in x + \varepsilon\mathbb{B}$.

The remaining technical results of the paper rely on the following assumption.

Assumption 1. *The function $f : \mathbb{R}^n \times \mathbb{R}^m$ is locally Lipschitz continuous in both arguments. The sets X and U are compact.*

The above assumption on f is very mild and is satisfied as long as the function $f : \mathbb{R}^n \times \mathbb{R}^m$ is differentiable with respect to both variables.

Proposition 4. *Let $\hat{\mathcal{S}} = (Q, A, R, \Pi, \hat{L})$, which is obtained from $\mathcal{S}$ in Definition 7 by replacing L with its ε-strengthening $\hat{L}$. Suppose that the assumptions of Corollary 1 hold with $\hat{\mathcal{S}}$ in place of $\mathcal{S}$. Then $(\mathcal{S}, \kappa) \models \varphi$, subject to measurement errors described in (7), provided that $(L+1)\varepsilon \leq \delta$, where L is the uniform Lipschitz constant for both variables of f on the compact set $(X + \varepsilon\mathbb{B}) \times U$.*

Proof. We have $\hat{\mathcal{S}} \oplus \Delta_\delta \preceq_\alpha \mathcal{T}$. The goal is to show that, despite the measurement errors, κ-controlled traces of $\mathcal{S}$ are a subset of the κ-controlled traces of $(\hat{\mathcal{S}}, \Delta)$ and therefore satisfies φ. Starting from $x(0)$, let $\hat{x}(0)$ be the measurement taken for $x(0)$. Suppose that an action $u(0) = \kappa(\hat{x}(0)) = \mu(q_0)$ is chosen by κ, where $q_0 \in \alpha(\hat{x}(0))$. Let L_1 be the labelling function for $\mathcal{T}$. Then $L_1(q_0) \subseteq \hat{L}(\hat{x}(0))$ by the definition of the robust abstraction. Since $\hat{L}$ is the ε-strengthening of L and $x(0) \in \hat{x}(0) + \varepsilon\mathbb{B}$, it follows that $L_1(q_0) \subseteq \hat{L}(\hat{x}(0)) \subseteq L(x(0))$.

We suppose by induction that $L_1(q_k) \subseteq L(x(k))$ holds for some $k \geq 0$, where $q_k \in \alpha(\hat{x}(k))$ and $\hat{x}(k) \in x(k) + \varepsilon\mathbb{B}$. The action at time k is given by $u(k) = \kappa(\hat{x}(0), \cdots, \hat{x}(k))$, which implements $a_k = \mu(q_0, \cdots, q_k)$ in the sense of Definition 6. The next state under $u(k)$ is given by $x(k+1) = f(x(k), u(k))$, whose measurement is $\hat{x}(k+1) = x(k+1) + e(k+1) \in x(k+1) + \varepsilon\mathbb{B}$. Hence $L_1(q_{k+1}) \subseteq \hat{L}(\hat{x}(k+1))$ implies $L_1(q_{k+1}) \subseteq \hat{L}(\hat{x}(k+1))L(x(k+1))$. Thus, $L(q_k) \subseteq L(x(k))$ for all $k \geq 0$.

We show that (q_k, a_k, q_{k+1}) is a valid transition in $\mathcal{T}$. Note that

$$\begin{aligned}
&\hat{x}(k+1)\\
&= x(k+1) + e(k+1)\\
&= f(x(k), u(k)) + e(k+1)\\
&= f(\hat{x}(k), u(k)) + (f(x(k), u(k)) - f(\hat{x}(k), u(k))) + e(k+1).
\end{aligned}$$

Since f is L-Lipschitz continuous in both arguments on the compact set $(X + \varepsilon\mathbb{B}) \times U$, the above equation shows that

$$\hat{x}(k+1) \in f(\hat{x}(k), u(k)) + (L+1)\varepsilon\mathbb{B} \subseteq f(\hat{x}(k), u(k)) + \delta\mathbb{B},$$

because $(L+1)\varepsilon \leq \delta$. Hence, by the choice of $u(k)$ by κ (which is an α-implementation of μ), we have

$$\begin{aligned}
q_{k+1} &\in \alpha(\hat{x}(k+1))\\
&\subseteq \alpha(f(\hat{x}(k), u(k)) + \delta\mathbb{B})\\
&\subseteq \alpha(\mathrm{Post}_{\hat{\mathcal{S}} \oplus \Delta}(\hat{x}(k), u(k)))\\
&\subseteq \mathrm{Post}_{\mathcal{T}}(q_k, a_k),
\end{aligned}$$

where $\hat{x}(k) \in \alpha^{-1}(q_k)$, which shows that (q_k, a_k, q_{k+1}) is a valid transition in $\mathcal{T}$ and therefore $q_0 q_1 q_2 \cdots$ is a valid path for $\mathcal{T}$. Since the trace of this path satisfies φ and $L_1(q_k) \subseteq L(x(k))$ for all $k \geq 0$, it follows that the trace of $x(0)x(1)x(2)\cdots$ also satisfies φ. □

Remark 2. The soundness result above states that to cope with measurement errors, we only need to choose δ

sufficiently large such that $(L+1)\varepsilon \leq \delta$ and strengthen the labelling function by a factor of ε. This condition simplifies the two robustness margins (γ_1, γ_2) considered in the work [14, 15] and also does not require that the abstraction relation to be non-deterministic in order to be robust with respect to measurement errors as stated in [21, Section VI.6].

3.3 Approximate completeness of robust abstractions for discrete-time control systems

In this section, we show that, under Assumption 1, computing robust abstractions for the discrete-time control system (4) is approximately complete, in the sense that, for arbitrary numbers $0 \leq \delta_1 < \delta_2$, we can find a finite transition system $\mathcal{T}$ such that $\mathcal{S} \oplus \Delta_{\delta_1} \preceq \mathcal{T} \preceq \mathcal{S} \oplus \Delta_{\delta_2}$, where $\mathcal{S}$ and Δ_{δ_i} $(i = 1, 2)$ are defined in Definitions 7 and 8. This result is made precise by the following theorem, which we present as the main result of the paper.

MAIN RESULT

THEOREM 2. *For any numbers $0 \leq \delta_1 < \delta_2$, let Δ_{δ_i} $(i = 1, 2)$ be given by Definition 8 with $\delta = \delta_i$. For any numbers $0 \leq \varepsilon_1 < \varepsilon_2$, let $L_{\mathcal{S}_i}$ $(i = 1, 2)$ be the ε_i-strengthening of $L_{\mathcal{S}}$. Let*

$$\mathcal{S}_i = (Q_{\mathcal{S}}, A_{\mathcal{S}}, R_{\mathcal{S}} \cup \Delta_{\delta_i}, \Pi, L_{\mathcal{S}_i}), \quad i = 1, 2.$$

Then there exists a finite transition system $\mathcal{T}$ such that

$$\mathcal{S}_1 \preceq \mathcal{T} \preceq \mathcal{S}_2. \tag{8}$$

To prove Theorem 2, we need the following lemma on over-approximation of the reachable set of a box in $\mathbb{R}^n$ under a nonlinear map.

no kenb...

LEMMA 1. *Fix any $\delta > 0$, any box (also called an interval or a hyperrectangle) $[x] \subseteq \mathbb{R}^n$, and any $u \in U$. For all $\varepsilon > 0$, there exists a finitely terminated algorithm to compute an over-approximation of the reachable set of $[x]$ under (6), i.e., the set*

$$Reach_{(6)}([x], u) = f([x], u) + \delta\mathbb{B},$$

such that

$$Reach_{(6)}([x], u) \subseteq \widehat{Reach}_{(6)}([x], u) \subseteq Reach_{(6)}([x], u) + \varepsilon\mathbb{B},$$

where $\widehat{Reach}_{(6)}([x], u)$ is the computed over-approximation given as a union of boxes.

PROOF. This is a well-known result in interval analysis, known as outer approximation of the image set of a function. It can be proved, for example, using the results in [11, Chapter 3]. Here we include a proof for completeness. Let $\mathbb{IR}^n$ denote the set of all boxes in $\mathbb{R}^n$. Let $[f_u] : \mathbb{IR}^n \to \mathbb{IR}^m$ be a *convergent inclusion function* [11] of $f(\cdot, u)$, which satisfies the following two conditions:

- $f([y], u) \subseteq [f_u]([y])$ for all $[y] \in \mathbb{IR}^n$;
- $\lim_{w([y])\to 0} w([f_u]([y])) = 0$,

where $w([y])$ is the width of $[y]$, given by $\max_{1\leq i\leq n}\{\overline{y_i} - \underline{y_i}\}$ if we write $[y] = [y_1] \times \cdots \times [y_n] \subseteq \mathbb{R}^n$ and $[y_i] = [\underline{y_i}, \overline{y_i}] \subseteq \mathbb{R}$ for $i = 1, \cdots, n$. Without loss of generality, assume that $\varepsilon < 1$. Because f is L-Lipschitz continuous on $[x]$ for some $L > 0$, we can find an inclusion function such that $w([f_u]([y])) \leq Lw([y])$ for any subintervals of $[x]$. We mince the interval $[x]$ into subintervals such that the largest width of among these subintervals is smaller than $\frac{\varepsilon}{2L}$. For each such interval $[y]$, we evaluate $[f_u]([y])$ and obtain the interval $[z] = [f_u]([y]) + \delta\mathbb{B}$. Let $\mathcal{Y}$ denote the collection of all such intervals[1] and let Y be its union. We claim that

$$Y = \widehat{\text{Reach}}_{(6)}([x], u)$$

satisfies the requirement of this lemma. This is clearly true because, for each interval $[z] = [f_u]([y]) + \delta\mathbb{B}$, we have $f([y]) + \delta\mathbb{B} \subseteq [z]$ and the distance from $[z]$ to the true reachable set $\text{Reach}_{(6)}([x], u)$ is bounded by $w([f_u]([y])) \leq L \cdot w([y]) \leq \frac{\varepsilon}{2}$. The proof for Lemma 1 is also summarized in pseudo code format in Algorithm 1. □

Algorithm 1 Computation of an over-approximation of $\text{Reach}_{(6)}([x], u)$ (Lemma 1)

Input: $[x]$, $\delta, \varepsilon > 0$, the Lipschitz constant L for $f(\cdot, u)$, and a centred convergent inclusion function $[f_u]$ for $f(\cdot, u)$

```
List ← [x]
𝒴 ← ∅
while List ≠ ∅ do
    [y] ← First(List)
    List ← List \ {[x]}
    if w([y]) ≤ ε/(2L) then
        [z] ← [f_u]([y]) + δ𝔹
        𝒴 ← 𝒴 ∪ {[z]}
    else
        {Left[y], Right[y]} = Bisect([y])
        List ← List ∪ {Left[y], Right[y]}
Y ← ∪_{[z]∈𝒴} [z]
return Y = Reach̄_(6)([x], u)
```

PROOF OF THEOREM 2. The proof is constructive and we construct a finite transition system

$$\mathcal{T} = (Q_{\mathcal{T}}, A_{\mathcal{T}}, R_{\mathcal{T}}, \Pi, L_{\mathcal{T}})$$

as follows.

For a positive integer k, let $\mathbb{Z}^k$ denote the k-dimensional integer lattice, i.e., the set of all k-tuples of integers. For parameters $\eta > 0$ and $\mu > 0$ (to be chosen later), define

$$[\mathbb{R}^n]_\eta = \eta\mathbb{Z}^n, \quad [\mathbb{R}^m]_\mu = \mu\mathbb{Z}^m,$$

where $\mu\mathbb{Z}^k = \{\mu z : z \in \mathbb{Z}^k\}$ (for $k = n, m$). Define a relation α from $Q_{\mathcal{S}}$ to $[\mathbb{R}^n]_\eta \cup \{X^c\}$ by

$$\left\{(x, q) : q = \eta\lfloor\frac{x}{\eta}\rfloor, x \in X\right\} \cup \{(X^c, X^c)\},$$

where $\lfloor\cdot\rfloor$ is the floor function (i.e., $\lfloor x\rfloor = (\lfloor x_1\rfloor, \cdots, \lfloor x_n\rfloor)$ and $\lfloor x_i\rfloor$ gives the largest integer less than or equal to x_i). Let $Q_{\mathcal{T}}$ be $\alpha(Q_{\mathcal{S}})$ and $A_{\mathcal{T}} = \left\{a : \exists u \in A_{\mathcal{S}} \text{ s.t. } a = \mu\lfloor\frac{u}{\mu}\rfloor\right\}$ (which are both non-empty by definition and are finite because X and U are compact). Note that this gives a deterministic relation in the sense that $\alpha(x)$ is single-valued for all x. It is straightforward to verify that

$$\alpha^{-1}(\alpha(B)) \subseteq B + \eta\mathbb{B}, \tag{9}$$

for any set $B \subseteq \mathbb{R}^n \cup \{X^c\}$ with the slight abuse of notation that $X^c + x = X^c$ for any $x \in \mathbb{R}^n$.

[1] Such a collection $\mathcal{Y}$ is called a non-regular paving of $\mathbb{R}^n$, which can be regularized [11, Chapter 3] to reduce the number of boxes and hence reduce complexity, but this is not necessary for our purpose.

We next construct $R_{\mathcal{T}}$. For each $q \in Q_{\mathcal{T}}$ and $a \in A_{\mathcal{T}}$, denote by

$$\mathrm{Reach}_{\mathcal{S}_1}(\alpha^{-1}(q), a) = \bigcup_{x \in \alpha^{-1}(q)} \mathrm{Post}_{\mathcal{S}_2}(x, a).$$

We let (q, a, q') be included in $R_{\mathcal{T}}$ if and only if

$$q' \in \alpha(\widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a)),$$

i.e.,

$$\mathrm{Post}_{\mathcal{T}}(q, a) = \alpha(\widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a)), \tag{10}$$

where $\widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a)$ is computed from Lemma 1 by setting $[x] = \overline{\alpha^{-1}(q)}$, $u = a$, and $\delta = \delta_1$. In particular, we set $\widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a) = \widehat{\mathrm{Reach}}_{(6)}([x], u)$, if $\widehat{\mathrm{Reach}}_{(6)}([x], u) \subseteq X$, and

$$\widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a) = \widehat{\mathrm{Reach}}_{(6)}([x], u) \cup \{X^c\},$$

if $\widehat{\mathrm{Reach}}_{(6)}([x], u) \not\subseteq X$.

Then it follows from Lemma 1 that

$$\begin{aligned}\alpha(\bigcup_{x \in \alpha^{-1}(q)} \mathrm{Post}_{\mathcal{S}_1}(x, a)) &\subseteq \alpha(\mathrm{Reach}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a)) \\ &\subseteq \alpha(\widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a)) \\ &= \mathrm{Post}_{\mathcal{T}}(q, a),\end{aligned}$$

which verifies condition (ii) of Definition 4 for $\mathcal{S}_1 \preceq_\alpha \mathcal{T}$.

Consider α^{-1} as a relation from $Q_{\mathcal{T}}$ to $Q_{\mathcal{S}}$. Then for each $x \in Q_{\mathcal{S}}$ and $u \in A_{\mathcal{S}}$, we can choose $a = \mu\lfloor \frac{u}{\mu} \rfloor \in A_{\mathcal{T}}$ such that

$$\begin{aligned}\alpha^{-1}(\bigcup_{q \in \alpha(x)} \mathrm{Post}_{\mathcal{T}}(q, a)) &= \alpha^{-1}(\mathrm{Post}_{\mathcal{T}}(q, a)) \\ &\subseteq \alpha^{-1}(\alpha(\widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a))) \\ &\subseteq \widehat{\mathrm{Reach}}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a) + \eta\mathbb{B} \\ &\subseteq \mathrm{Reach}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a) + (\eta + \varepsilon)\mathbb{B}.\end{aligned}$$

where we used (10), (9), and Lemma 1. We claim that, if we can choose η, μ, and ε sufficiently small such that

$$\delta_1 + L(\eta + \mu) + \eta + \varepsilon \le \delta_2, \tag{11}$$

then

$$\mathrm{Reach}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a) + (\eta + \varepsilon)\mathbb{B} \subseteq \mathrm{Post}_{\mathcal{S}_2}(x, u). \tag{12}$$

Note that $\overline{\alpha^{-1}(q)} \subseteq x + \eta\mathbb{B}$ and $a \in u + \mu\mathbb{B}$. We first assume that $X^c \notin \mathrm{Reach}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a)$. Without loss of generality, we can assume that $\eta \le 1$ and $\mu \le 1$. Because f is Lipschitz continuous in both arguments on the compact set $(X + \mathbb{B}) \times (U + \mathbb{B})$ (we use L to indicate the uniform Lipschitz constant for both variables on this set), it follows that

$$\mathrm{Reach}_{\mathcal{S}_1}(\overline{\alpha^{-1}(q)}, a) \subseteq f(x, u) + [\delta_1 + L(\eta + \mu)]\mathbb{B}.$$

Combining the displayed equations above, we obtain

$$\begin{aligned}\alpha^{-1}(\bigcup_{q \in \alpha(x)} \mathrm{Post}_{\mathcal{T}}(q, a)) &\subseteq f(x, u) + \delta_2\mathbb{B} \\ &= \mathrm{Post}_{\mathcal{S}_2}(x, u),\end{aligned}$$

which verifies condition (ii) of Definition 4 for $\mathcal{T} \preceq_\alpha \mathcal{S}_2$, because $X^c \in \alpha^{-1}(\bigcup_{q \in \alpha(x)} \mathrm{Post}_{\mathcal{T}}(q, a))$ would also imply $X^c \in \mathrm{Post}_{\mathcal{S}_2}(x, u)$.

Now we define $L_{\mathcal{T}}$. For each $q \in Q_{\mathcal{T}}$, define

$$\pi \in L_{\mathcal{T}}(q)$$

if and only if $\pi \in L_{\mathcal{S}}(x)$ for all $x \in q + \frac{\varepsilon_1 + \varepsilon_2}{2}\mathbb{B}$. Choose η sufficiently small such that $\eta + \frac{\varepsilon_1 + \varepsilon_2}{2} < \varepsilon_2$. This is possible because $\varepsilon_2 > \varepsilon_1$. To verify condition (iii) of Definition 4 for $\mathcal{S}_1 \preceq_\alpha \mathcal{T}$ and $\mathcal{T} \preceq_{\alpha^{-1}} \mathcal{S}_2$, we need to check that

$$L_{\mathcal{S}_2}(x) \subseteq L_{\mathcal{T}}(q) \tag{13}$$

and

$$L_{\mathcal{T}}(q) \subseteq L_{\mathcal{S}_1}(x) \tag{14}$$

for all $(x, q) \in \alpha$. Fix any $(x, q) \in \alpha$. If $\pi \in L_{\mathcal{S}_2}(x)$, then $\pi \in L_{\mathcal{S}}(y)$ for all $y \in x + \varepsilon_2\mathbb{B}$. Since $q + \frac{\varepsilon_1 + \varepsilon_2}{2}\mathbb{B} \subseteq x + [\eta + \frac{\varepsilon_1 + \varepsilon_2}{2}]\mathbb{B} \subseteq x + \varepsilon_2\mathbb{B}$, we have $\pi \in L_{\mathcal{S}}(y)$ for all $y \in q + \frac{\varepsilon_1 + \varepsilon_2}{2}\mathbb{B}$ and $\pi \in L_{\mathcal{T}}(q)$. Hence, (13) holds. If $\pi \in L_{\mathcal{T}}(q)$, then $\pi \in L_{\mathcal{S}}(y)$ for all $y \in q + \frac{\varepsilon_1 + \varepsilon_2}{2}\mathbb{B}$ by the definition of $L_{\mathcal{T}}$. Since $x + \varepsilon_1\mathbb{B} \subseteq q + (\eta + \varepsilon_1)\mathbb{B} \subseteq q + \frac{\varepsilon_1 + \varepsilon_2}{2}\mathbb{B}$, we have $\pi \in L_{\mathcal{S}}(y)$ for all $y \in x + \varepsilon_1\mathbb{B}$ and $\pi \in L_{\mathcal{S}_1}(x)$. Hence, (14) holds.

We have verified $\mathcal{S}_1 \preceq \mathcal{T} \preceq \mathcal{S}_2$ by checking all the conditions of Definition 4. The main steps of the proof are also summarized in pseudo code format in Algorithm 2. □

Algorithm 2 Computation of an approximately complete robust abstraction $\mathcal{T}$ for $\mathcal{S}$ (Theorem 2)

Input: $\mathcal{S} = (Q_{\mathcal{S}}, A_{\mathcal{S}}, R_{\mathcal{S}}, \Pi, L_{\mathcal{S}})$, numbers $0 \le \delta_1 < \delta_2$ and $0 \le \varepsilon_1 < \varepsilon_2$

Set $L_{\mathcal{S}_i}$ be the ε_i-strengthening of $L_{\mathcal{S}}$ $(i = 1, 2)$
Set Δ_{δ_i} according to Definition 8 $(i = 1, 2)$
Set $\mathcal{S}_i = (Q_{\mathcal{S}}, A_{\mathcal{S}}, R_{\mathcal{S}} \cup \Delta_{\delta_i}, \Pi, L_{\mathcal{S}_i})$ $(i = 1, 2)$
Choose rational numbers $\eta \in (0, 1)$ and $\varepsilon \in (0, 1)$ such that $\delta_1 + L(\eta + \mu) + \eta + \varepsilon \le \delta_2$ and $\eta + \frac{\varepsilon_1 + \varepsilon_2}{2} < \varepsilon_2$, where L is the uniform Lipschitz constant of f on the compact set $(X + \mathbb{B}) \times (U + \mathbb{B})$
Set $Q_{\mathcal{T}} = \left\{x \in [\mathbb{R}^n]_\eta : \exists x \in Q_{\mathcal{S}} \text{ s.t. } x = \eta\lfloor \frac{u}{\eta} \rfloor\right\} \cup \{X^c\}$
Set $A_{\mathcal{T}} = \left\{a \in [\mathbb{R}^m]_\mu : \exists u \in A_{\mathcal{S}} \text{ s.t. } a = \mu\lfloor \frac{u}{\mu} \rfloor\right\}$
for all $q \in Q_{\mathcal{T}}$ **do**
    $L_{\mathcal{T}}(q) \leftarrow \emptyset$
    for all $\pi \in \Pi$ **do**
        if $\pi \in L_{\mathcal{S}}(x)$ for all $x \in q + \frac{\varepsilon_1 + \varepsilon_2}{2}\mathbb{B}$ **then**
            $L_{\mathcal{T}}(q) \leftarrow L_{\mathcal{T}}(q) \cup \{\pi\}$
$R_{\mathcal{T}} \leftarrow \emptyset$
for all $q \in Q_{\mathcal{T}}$ **do**
    for all $a \in A_{\mathcal{T}}$ **do**
        if $q' \in \alpha(\widehat{\mathrm{Reach}}_{(6)}(\overline{\alpha^{-1}(q)}, a))$ **then**
            $R_{\mathcal{T}} \leftarrow R_{\mathcal{T}} \cup \{(q, a, q')\}$
return $\mathcal{T} = (Q_{\mathcal{T}}, A_{\mathcal{T}}, R_{\mathcal{T}}, \Pi, L_{\mathcal{T}})$

Remark 3. While the disturbance sets are so chosen for simplicity of presentation, they do not have to be of the form $\delta\mathbb{B}$. In fact, if we choose two arbitrary sets W_1 and W_2 in place of $\delta_1\mathbb{B}$ and $\delta_2\mathbb{B}$ in Definition 8 such that there exists $\varepsilon > 0$ such that $W_1 + \varepsilon\mathbb{B} \subseteq W_2$, then a completeness result similar to Theorem 2 can be stated. Furthermore, δ can be a vector in $\mathbb{R}^n$ instead of a scalar, in which case $\delta_i\mathbb{B}$ becomes a hyperrectangle and the condition $0 \le \delta_1 < \delta_2$ is a componentwise inequality.

REMARK 4. In the proof of Theorem 2, we in fact construct a single-valued abstraction relation α. While the main results of the paper are presented for the case where α can be multi-valued, it appears, in view of the proof of Theorem 2, that for practical purposes, α may always be chosen to be deterministic, while still preserving robustness (see also Remark 2).

Finally, we would like to point out that Theorem 2 shows that there exists an *approximately complete* abstraction procedure for discrete-time nonlinear control systems of the form (4) in the sense that, if a specification φ is realizable for $\mathcal{S}_2$ (namely, a δ_2-perturbation of $\mathcal{S}$), then there is a robust abstraction $\mathcal{T}$ of $\mathcal{S}_1$, which is a δ_1-perturbation of $\mathcal{S}$, such that φ is realizable for $\mathcal{T}$ and hence it is also realizable for $\mathcal{S}_1$. Note that $\mathcal{S}_1$ and $\mathcal{S}_2$ can be made arbitrarily close by choosing δ_2 close to δ_1 and ε_2 close to ε_1. Since the proof of above theorem is constructive, we can algorithmically synthesize a control strategy for $\mathcal{S}_1$ by computing $\mathcal{T}$ first and then solving a discrete synthesis problem for $\mathcal{T}$ with the specification φ. We summarize this in the following corollary.

COROLLARY 2. *Let $\mathcal{S}_1$, $\mathcal{S}_2$, and φ be as defined in Theorem 2. There is a decision procedure to answer one of the following two questions:*

(i) there exists a control strategy κ (and one can algorithmically construct it) such that $(\mathcal{S}_1, \kappa) \vDash \varphi$;

(ii) φ is not realizable for $\mathcal{S}_2$.

4. AN EXAMPLE

We use a simple motion planning example to illustrate our results. Consider a vehicle steering problem, where the dynamics of the vehicle are given by the so-called bicycle model [5]. The same example is used for illustration of abstraction-based control design in [21, 22, 27]. The model is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} u_1 \cos(\alpha + x_3)/\cos(\alpha) \\ u_1 \sin(\alpha + x_3)/\cos(\alpha) \\ u_1 \tan(u_2) \end{bmatrix},$$

where $(x_1, x_2, x_3) = (x, y, \theta)$, $(u_1, u_2) = (v, \varphi)$, and $\alpha = \arctan(a \tan(u_2)/b)$. The constant $b = 1$ is the wheel base and $a = 0.5$ is the distance between centre of mass and rear wheels. The states consist of the coordinates of the centre of the mass (x, y) and the heading angle θ. The controls consist of the wheel speed v and the steering angle φ. The variable α is the angle of velocity depending on φ.

Let $X = [7, 10] \times [0, 4.5] \times [-\pi, \pi]$ and $U = [-1, 1] \times [-1, 1]$. Consider a workspace and a specification given by $\varphi = A_I \wedge \Box(\neg A_O) \wedge \Diamond A_G$, where $A_I = [7.6, 0.4, \pi/2]^T$, $A_G = [9, 9.6] \times [0, 0.6] \times [-\pi, \pi]$, $A_O = A_{O1} \cup A_{O2} \cup A_{O3}$, $A_{O1} = [8.2, 8.4] \times [0, 3.6] \times [-\pi, \pi]$, $A_{O2} = [8.4, 9.4] \times [3.4, 3.6] \times [-\pi, \pi]$, $A_{O3} = [9.4, 10] \times [2.4, 2.6] \times [-\pi, \pi]$.

To design a control strategy to realize this specification, we discretize the model using a sampling time step $\tau = 0.3$. We first consider the case with no disturbance, i.e., $\delta = 0$. Using the discretization parameters $\eta = 0.2$ and $\mu = 0.3$, the resulting nominal abstraction consists of 12,880 states and 3,023,040 transitions. The computation time was 7.3s for computing the abstraction and 8.6s for solving the synthesis problem on a 2.2GHz Intel Core i7 processor. To design a robust control strategy, we consider an additive disturbance of size $\delta = 0.05$ on the right-hand side of the discretized system. We compute a robust abstraction by setting $\delta = 0.05$ and $\eta = 0.05$. The resulting robust abstraction consists of 782,691 states and 1,727,548,752 transitions. The computation time was 2,327s for abstraction and 2,289s for synthesis on the same processor. We used a modified version of the toolbox SCOTS [22] with an interval arithmetic extension to solve this example.

A feasible trajectory resulting from the nominal controller is shown in Figure 2, where no disturbance was added. A feasible trajectory resulting from a robust controller is shown in Figure 3 (left), where an additive disturbance of size $\delta = 0.05$ was added. Using the same controller, a simulated trajectory shown in Figure 3 (right) with an additive disturbance of size $\delta = 0.15$ fails to satisfy the specification. Furthermore, Theorem 2 implies that, for any $0.05 \leq \delta_1 < \delta_2$, by further refining the abstraction, we should be able to assert that either the specification is robustly realizable with a disturbance of size δ_1 or the specification is not realizable with a disturbance of size δ_2.

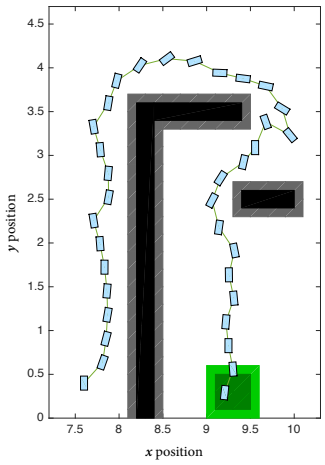

Figure 2: A simulated trajectory from a nominal abstraction that satisfies the specification.

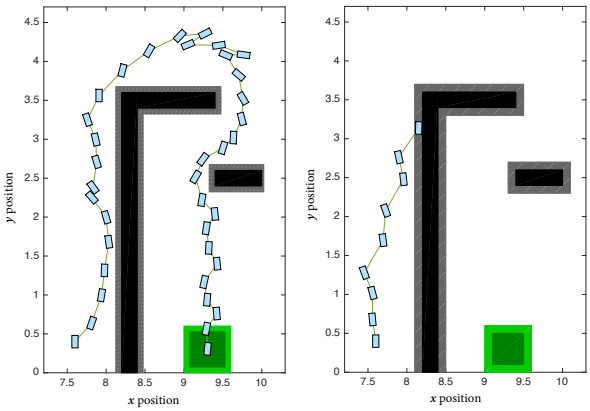

Figure 3: A valid trajectory (left) obtained from a robust abstraction with $\delta = 0.05$ and a failed trajectory (right) with disturbance size $\delta = 0.15$.

5. CONCLUSIONS AND DISCUSSIONS

We proposed a computational framework for designing robust abstractions for control synthesis. It is shown that robust abstractions are not only sound in the sense that they preserve robust satisfaction of linear-time properties, but also approximately complete in the sense that, given a concrete discrete-time control system and an arbitrarily small perturbation of this system, there exists a finite transition system that robustly abstracts the concrete system and is abstracted by the perturbed system at the same time. Consequently, the existence of controllers for a general discrete-time nonlinear control system and linear-time specifications is robustly decidable: if a specification is robustly realizable, there is a decision procedure to find a (potentially less) robust control strategy.

It is interesting to note that the connection between robustness and decidability appeared in different contexts. Recently, the notion of δ-decidability for satisfiability over the reals [8] and δ-reachability analysis [12] have been proposed to turn otherwise undecidable problems into decidable ones. A notion of "robustness implies decidability" was proposed in early work in [7] for verifying bounded properties for polynomial hybrid automaton and in [4] for reachability analysis of several simple models of hybrid systems. Finally, the early work in [2] showed that *robust* stability is decidable for linear systems in the context of output feedback stabilization. In this sense, the current work can serve as an example of "robustness implies decidability" in the context of linear-time logic control synthesis for nonlinear systems.

6. ACKNOWLEDGMENTS

This research was supported in part by NSERC Canada and the University of Waterloo. The author would like to thank Necmiye Ozay and Yinan Li for stimulating discussions on related topics and the anonymous reviewers for helpful comments and suggestions.

7. REFERENCES

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

[2] B. Anderson, N. Bose, and E. Jury. Output feedback stabilization and related problems-solution via decision methods. *IEEE Transactions on Automatic control*, 20(1):53–66, 1975.

[3] D. Angeli et al. A lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

[4] E. Asarin and A. Bouajjani. Perturbed turing machines and hybrid systems. In *Proc. of LICS*, pages 269–278. IEEE, 2001.

[5] K. J. Aström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.

[6] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[7] M. Fränzle. What will be eventually true of polynomial hybrid automata? In *Proc. of TACS*, pages 340–359. Springer, 2001.

[8] S. Gao, J. Avigad, and E. M. Clarke. δ-complete decision procedures for satisfiability over the reals. In *Proc. of IJCAR*, pages 286–300. Springer, 2012.

[9] A. Girard. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5):947–953, 2012.

[10] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans. on Automatic Control*, 55:116–126, 2010.

[11] L. Jaulin. *Applied Interval Analysis*. Springer Science & Business Media, 2001.

[12] S. Kong, S. Gao, W. Chen, and E. Clarke. dreach: δ-reachability analysis for hybrid systems. In *Proc. of TACAS*, pages 200–205. Springer, 2015.

[13] Y. Li, J. Liu, and N. Ozay. Computing finite abstractions with robustness margins via local reachable set over-approximation. In *Proc. ADHS*, 2015.

[14] J. Liu and N. Ozay. Abstraction, discretization, and robustness in temporal logic control of dynamical systems. In *Proc. of HSCC*, pages 293–302, 2014.

[15] J. Liu and N. Ozay. Finite abstractions with robustness margins for temporal logic-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 22:1–15, 2016.

[16] N. Ozay, J. Liu, P. Prabhakar, and R. M. Murray. Computing augmented finite transition systems to synthesize switching protocols for polynomial switched systems. In *Proc. of ACC*, pages 6237–6244, 2013.

[17] A. Pnueli. The temporal logic of programs. In *Proc. of FOCS*, pages 46–57. IEEE, 1977.

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

[19] G. Reissig. Computing abstractions of nonlinear systems. *IEEE Trans. Automatic Control*, 56:2583–2598, 2011.

[20] G. Reissig and M. Rungger. Feedback refinement relations for symbolic controller synthesis. In *Proc. of CDC*, pages 88–94. IEEE, 2014.

[21] G. Reissig, A. Weber, and M. Rungger. Feedback Refinement Relations for the Synthesis of Symbolic Controllers. *IEEE Transactions on Automatic Control, to appear*, 2016.

[22] M. Rungger and M. Zamani. Scots: A tool for the synthesis of symbolic controllers. In *Proc. of HSCC*, pages 99–104. ACM, 2016.

[23] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.

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

[25] Y. Tazaki and J. Imura. Discrete abstractions of nonlinear systems based on error propagation analysis. *IEEE Trans. Automatic Control*, 57:550–564, 2012.

[26] U. Topcu, N. Ozay, J. Liu, and R. M. Murray. On synthesizing robust discrete controllers under modeling uncertainty. In *Proc. of HSCC*, pages 85–94. ACM, 2012.

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