Finite abstractions with robustness margins for temporal logic-based control synthesis☆

Jun Liu[a], Necmiye Ozay[b]

[a]Department of Applied Mathematics, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada
[b]Department of Electrical Engineering and Computer Science, University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109-2122, USA

Abstract

This paper introduces a notion of finite abstractions that can be used to synthesize robust controllers for dynamical systems from temporal logic specifications. These finite abstractions, equipped with certain robustness margins, provide a unified approach to various issues commonly encountered in implementing control systems, such as inter-sample behaviors of a sampled-data system, effects of imperfect state measurements and unmodeled dynamics. The main results of this paper demonstrate that the robustness margins can effectively account for the mismatches between a control system and its finite abstractions used for control synthesis. The quantitative nature of the robustness margins also makes it possible to study the trade-offs between the performance of controllers and their robustness against various types of adversaries (e.g., delays, measurement errors, or modeling uncertainties). We use a simple adaptive cruise control (ACC) example to illustrate such robustness-performance trade-offs.

Keywords:
Finite abstractions, control synthesis, robustness, temporal logic, hybrid systems.

1. Introduction

Designing hybrid controllers from high-level specifications using abstraction-based, hierarchical approaches has gained increased popularity over the last few years (see, e.g., [10, 16, 17, 21, 22, 25, 27, 33, 34, 40, 43]). The typical workflow of these approaches are as follows: (i) computation of finite abstractions of the system to be controlled, (ii) discrete synthesis based on the computed abstractions and desired specifications to obtain a discrete strategy, (iii) hybrid implementation of the discrete control strategy to ensure correctness of the overall system. In particular, how to compute finite abstractions of nonlinear control systems has received special attention (see [31, 37] and references therein), as it is the first and most important step in ensuring the overall correctness of such approaches.

One advantage of using abstraction-based methods is that they can provide a feedback solution, as opposed to open-loop trajectory generation strategies [15, 39]. While feedback has the potential to reduce the effects of disturbances and deal with sensing and modeling uncertainties, it remains unclear how to establish robustness of a hybrid feedback controller obtained from abstraction-based methods when the requirements are given in a high-level temporal logic. Motivated by this question, in this paper, we present a unified notion of finite abstractions that can be used to synthesize robust hybrid controllers from high-level specifications. These finite abstractions are equipped with additional robustness margins to account

☆Research supported in part by EU FP7 Grant PCIG13-GA-2013-617377 and by NSF grant CNS-1446298. A preliminary version of this paper was presented in the 17th International Conference on Hybrid Systems: Computation and Control (HSCC) [20].
Email addresses: j.liu@uwaterloo.ca (Jun Liu), necmiye@umich.edu (Necmiye Ozay)

for imperfections in measurements and/or models. More specifically, by focusing on temporal logic specifications and nonlinear control systems, we show that, when the abstractions comply with these margins with respect to a nominal dynamical system, then it is possible to synthesize a hybrid control strategy that remains valid for a family of perturbed dynamical systems (i.e., that can be represented as the nominal dynamical system subject to uncertainty).

The main results of this paper show that it is possible to establish robustness against various issues that are commonly encountered in implementing control systems, namely inter-sample behaviors of a sampled-data system, effects of imperfect state measurements, unmodeled dynamics, jitter and delays within an abstraction-based framework. Such issues have been studied extensively for stability analysis, but they received less attention in the context of temporal logic-based control for dynamical systems. The quantitative nature of the robustness margins also provides explicit trade-offs between the performance of the hybrid controllers being designed and their robustness against various types of adversaries (e.g., delays, measurement errors, or modeling uncertainties).

A preliminary version of this paper appeared in [20]. The current paper differs from [20] in a number of ways. First, we provide a more in-depth discussion of abstractions and robustness margins both for continuous-time and for discrete-time systems. Second, in this paper, we define the robustness margins to be vector-valued parameters. This allows the use of hyper-boxes for computing abstractions and makes the abstractions less conservative as demonstrated in the examples section. Third, we discuss in this paper the trade-offs between robustness and performance of synthesized controllers using a new example on adaptive cruise control design. Fourth, we have added a more detailed explanation of the implementations of discrete control strategies, including a formal definition of continuous implementations of discrete strategies and new block diagrams showing the details of digital implementations. Making the implementation semantics explicit is crucial to talk about the correctness of the closed-loop system. Fifth, we have added a detailed model description for the case with imperfect state measurements. Finally, we have expanded the related work section and added several new remarks to discuss more about relevant work.

The rest of the paper is organized as follows. Preliminaries on temporal logics and control system models are given in Section 2. Finite abstractions with robustness margins are introduced in Section 3. The main results that demonstrate the effectiveness of the new abstraction framework are presented in Section 4. An example on vehicular cruise control is used to illustrate the results in Section 6, highlighting robustness-performance trade-offs.

2. Preliminaries

Notation: $\mathbb{R}^n$ denotes the n-dimensional Euclidean space; given an n-vector $x = (x_1, \cdots, x_n)$ in $\mathbb{R}^n$, let $|x| = (|x_1|, \cdots, |x_n|)$, i.e., the n-vector obtained by taking entry-wise absolute value of x; given two n-vectors $x = (x_1, \cdots, x_n)$ and $y = (y_1, \cdots, y_n)$, $x \leq y$ means $x_i \leq y_i$ for all $i \in \{1, \cdots, n\}$ ($x < y$, $x > y$, and $x \geq y$ are similarly defined); an n-vector x is said to be positive if $x > 0 \in \mathbb{R}^n$; given n-vectors $\delta \geq 0$ and x, let $B_\delta(x) := \{x' \in \mathbb{R}^n : |x' - x| \leq \delta\}$; let $\mathbb{R}^+$ denote the nonnegative real line; given an interval $I \subseteq \mathbb{R}^+$ and $U \subseteq \mathbb{R}^m$, U^I denotes the set of control input signals from I to U; given a function f, $\mathrm{dom}(f)$ denotes its domain; given a scalar $r > 0$, $\mathcal{C}_r$ denotes the space of $\mathbb{R}^n$-valued continuous functions on $[-r, 0]$.

2.1. Linear temporal logics

We use the stutter-invariant fragment of linear temporal logic (denoted by $\mathrm{LTL}_{\setminus\bigcirc}$ [4], which means LTL without the next operator $\bigcirc$) to specify system properties. 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., *always* ($\Box$), *eventually* ($\Diamond$), *until* ($\mathcal{U}$) and *release* ($\mathcal{R}$)).

The syntax of $\mathrm{LTL}_{\setminus\bigcirc}$ over a set of atomic propositions Π is defined inductively follows:

- **true** and **false** are $\text{LTL}_{\setminus\bigcirc}$ formulae;
- an atomic proposition $\pi \in \Pi$ is an $\text{LTL}_{\setminus\bigcirc}$ formula;
- if φ and ψ are $\text{LTL}_{\setminus\bigcirc}$ formulas, then $\neg\varphi$, $\varphi \vee \varphi$, and $\varphi \mathcal{U} \varphi$ are $\text{LTL}_{\setminus\bigcirc}$ formulas,

where atomic propositions are statements on a certain state space X. A labeling function $L : X \to 2^{\Pi}$ maps a state to a set of propositions that hold true for this state.

Negation Normal Form (NNF): All $\text{LTL}_{\setminus\bigcirc}$ formulas can be transformed into negation normal form [8, p. 132], where

- all negations appear only in front of the atomic propositions[1];
- only the logical operators **true**, **false**, $\wedge$, and $\vee$ can appear; and
- only the temporal operators $\mathcal{U}$ and $\mathcal{R}$ can appear, where $\mathcal{R}$ is defined by $\varphi_1 \mathcal{R} \varphi_2 \equiv \neg(\neg\varphi_1 \mathcal{U} \neg\varphi_2)$, called the *dual until* operator.

For syntactic convenience, we can define additional temporal operators $\Box$ and $\Diamond$ by $\Box\varphi \equiv \texttt{false}\mathcal{R}\varphi$ and $\Diamond\varphi \equiv \texttt{true}\mathcal{U}\varphi$.

Linear temporal logic formulas can be interpreted over both continuous-time signals and discrete sequences taking values in the state space X.

Continuous semantics of $\text{LTL}_{\setminus\bigcirc}$: Given a continuous-time signal $\xi \in X^{\mathbb{R}^+}$, we define $\xi, t \vDash \varphi$ with respect to an $\text{LTL}_{\setminus\bigcirc}$ formula φ at time t inductively as follows:

- $\xi, t \vDash \pi$ if and only if $\pi \in L(\xi(t))$;
- $\xi, t \vDash \varphi_1 \vee \varphi_2$ if and only if $\xi, t \vDash \varphi_1$ or $\xi, t \vDash \varphi_2$;
- $\xi, t \vDash \varphi_1 \wedge \varphi_2$ if and only if $\xi, t \vDash \varphi_1$ and $\xi, t \vDash \varphi_2$;
- $\xi, t \vDash \varphi_1 \mathcal{U} \varphi_2$ if and only if there exists $t' \geq 0$ such that $\xi, t + t' \vDash \varphi_2$ and for all $t'' \in [0, t')$, $\xi, t + t'' \vDash \varphi_1$;
- $\xi, t \vDash \varphi_1 \mathcal{R} \varphi_2$ if and only if, for all $t' \geq 0$, at least one of the following holds: $\xi, t + t' \vDash \varphi_2$ or there exists $t'' \in [0, t')$ such that $\xi, t + t'' \vDash \varphi_1$.

We write $\xi \vDash \varphi$ if $\xi, 0 \vDash \varphi$.

Discrete semantics of $\text{LTL}_{\setminus\bigcirc}$: Given a sequence $\rho = \{x_i\}_{i=0}^{\infty}$ in X, we define $\rho, i \vDash \varphi$ with respect to an $\text{LTL}_{\setminus\bigcirc}$ formula φ inductively as follows:

- $\rho, i \vDash \pi$ if and only if $\pi \in L(x_i)$;
- $\rho, i \vDash \varphi_1 \vee \varphi_2$ if and only if $\rho, i \vDash \varphi_1$ or $\rho, i \vDash \varphi_2$;
- $\rho, i \vDash \varphi_1 \wedge \varphi_2$ if and only if $\rho, i \vDash \varphi_1$ and $\rho, i \vDash \varphi_2$;
- $\rho, i \vDash \varphi_1 \mathcal{U} \varphi_2$ if and only if there exists $j \geq i$ such that $\rho, j \vDash \varphi_2$ and $\rho, k \vDash \varphi_1$ for all $k \in [i, j)$;
- $\rho, i \vDash \varphi_1 \mathcal{R} \varphi_2$ if and only if, for all $j \geq i$, at least one of the following holds: $\rho, j \vDash \varphi_2$ or there exists $k \in [i, j)$ such that $\rho, k \vDash \varphi_1$.

Similarly, we write $\rho \vDash \varphi$ if $\rho, 0 \vDash \varphi$.

[1] Hence all negations can be effectively removed by introducing new atomic propositions corresponding to the negations of current ones. We assume this has been done for all $\text{LTL}_{\setminus\bigcirc}$ formulas involved in this paper. This is for technical convenience in proving the main results of the paper, where we can change the labels of states with atomic propositions consistently, rather than having to deal with atomic propositions under negations separately.

2.2. Control systems

A continuous-time control system is a tuple $\mathcal{S} = (X, X_0, U, f, \Pi, L)$ where $X \subseteq \mathbb{R}^n$ is a set of states, $X_0 \subseteq X$ is a set of initial states, $U \subseteq \mathbb{R}^m$ is a set of inputs, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a vector field, Π is a set of atomic propositions and $L : X \to 2^{\Pi}$ is a labeling function. The state evolves according to:

$$\dot{x} = f(x, u). \tag{2.1}$$

We assume that f satisfies the basic conditions [3] such that, given any sufficiently regular control input signal $\mathbf{u} \in U^{[0,T]}$ for some $T \geq 0$ and any initial condition $x_0 \in X_0$, there exists a unique solution x defined on $[0, T]$ satisfying $x(0) = x_0$ and $\dot{x}(s) = f(x(s), \mathbf{u}(s))$ for all $s \in [0, T]$.

A *control strategy* for $\mathcal{S}$ is a partial function of the form:

$$\sigma(x_0, \cdots, x_i) = \mathbf{u}_i \in U^{[0,\Delta_i]},\ \forall i = 0, 1, 2, \cdots, \tag{2.2}$$

where $x_0, \cdots, x_i$ is a finite sequence of sampled states taken at sampling times $\tau_0 = 0, \cdots, \tau_i$ and $\mathbf{u}_i$ is a control input signal with duration Δ_i. The sampling times τ_0, τ_1, τ_2, $\cdots$ satisfy $\tau_{i+1} - \tau_i = \Delta_i$ for all $i \geq 0$. Note that Δ_i is not fixed *a priori* but to be computed as a part of the input signal $\mathbf{u}_i$.

Continuous Synthesis Problem: Given a continuous-time system $\mathcal{S} = (X, X_0, U, f, \Pi, L)$ and an $\mathrm{LTL}_{\setminus\bigcirc}$ specification φ over Π, find a control strategy for the system such that all of its solutions satisfy φ for all initial conditions in X_0.
If there exists such a control strategy, we say that φ is *realizable* for $\mathcal{S}$.

3. Abstractions with robustness margins

In this section, we define finite abstractions with robustness margins for $\mathcal{S}$. These abstractions are finite transition systems induced by an abstraction map.

A *transition system* is a tuple $\mathcal{T} = (Q, Q_0, \mathcal{A}, \to_{\mathcal{T}}, \Pi, L)$, where Q is a (finite or infinite) set of states and Q_0 the initial states; $\mathcal{A}$ is a (finite or infinite) set of actions; $\to_{\mathcal{T}} \subseteq Q \times \mathcal{A} \times Q$ is a transition relation; Π is a set of atomic propositions; $L : Q \to 2^{\Pi}$ is a labeling function. $\mathcal{T}$ is said to be *finite* if the cardinality of Q and $\mathcal{A}$ are finite.

An *abstraction map* $\alpha : X \to 2^Q$ maps states in X into subsets of a finite set Q that effectively introduces a finite covering of X given by $\bigcup_{q \in Q} \alpha^{-1}(q)$.

Definition 3.1. Given the continuous-time control system $\mathcal{S} = (X, X_0, U, f, \Pi, L)$ and a tuple of positive n-vectors $(\eta, \gamma_1, \gamma_2, \delta)$ satisfying $\gamma_i \geq \eta$ $(i = 1, 2)$ and $\delta \geq \eta$, a finite transition system

$$\mathcal{T}_{\mathcal{S}} = (Q, Q_0, \mathcal{A}, \to_{\mathcal{T}_{\mathcal{S}}}, \Pi, \hat{L})$$

is called an $(\eta, \gamma_1, \gamma_2, \delta)$-*abstraction* of $\mathcal{S}$ if there exists an abstraction map $\alpha : X \to 2^Q$ such that

(i) Q and $\mathcal{A}$ are finite subsets of X and $\bigcup_{\tau \in \mathbb{R}^+} U^{[0,\tau]}$, respectively, and $\bigcup_{x \in X_0} \alpha(x) \subseteq Q_0$;

(ii) $|x - q| \leq \eta$ for all $(x, q) \in X \times Q$ such that $q \in \alpha(x)$;

(iii) $(q, \mathbf{u}, q') \in \to_{\mathcal{T}_{\mathcal{S}}}$ if there exists $\xi : [0, \tau] \to \mathbb{R}^n$ such that $|\xi(0) - q| \leq \gamma_1$, $|\xi(\tau) - q'| \leq \gamma_2$, and $\dot{\xi}(s) = f(\xi(s), \mathbf{u}(s))$ and $B_{\gamma_2}(\xi(s)) \subseteq X$ for all $s \in [0, \tau]$, where $\mathrm{dom}(\mathbf{u}) = [0, \tau]$;

(iv) $\hat{L} : Q \to 2^{\Pi}$ is defined by

$$\pi \in \hat{L}(q),\ q \in Q \iff \pi \in L(x),\ \forall x \in B_{\delta}(q). \tag{3.1}$$

We write $\mathcal{S} \preceq_{(\eta, \gamma_1, \gamma_2, \delta)} \mathcal{T}_{\mathcal{S}}$.

Remark 3.1. Without loss of generality, we have assumed that Q is a subset of X; if not, for each $q \in Q$, we can pick a point $x \in X$ to represent q. Since each action in $\mathcal{A}$ is a control input signal of some finite duration and $\mathcal{A}$ is a finite set, there exists a maximum duration for all signals in $\mathcal{A}$, denoted by $\Delta(\mathcal{A})$ or Δ. For example, if we restrict the actions to signals of a fixed duration Δ_s (e.g., due to periodic sampling), we have $\Delta = \Delta_s$. Also, note that by definition only input signals that lead to trajectories that remain in the set X are considered.

Remark 3.2. Intuitively, the abstraction relation defined above is an over-approximation of the nominal concrete model $\mathcal{S}$ by an abstract model $\mathcal{T}_\mathcal{S}$ in the sense that, under a given control signal, the transitions in the abstract model should capture all possible behaviours of the concrete system. As such, the above relation resembles the notion of alternating simulation [1] and is related to symbolic models for control systems using the notion of alternating (bi)simulations [30] (see also [43]). The key difference in the abstraction introduced by Definition 3.1 is that it provides additional robustness parameters to explicitly account for mismatches not only between the abstract model and the concrete nominal model, but also between the nominal concrete model $\mathcal{S}$ and various perturbed models (as will be illustrated by the results in Section 4). More specifically, the parameter η captures the granularity of the approximation, the parameter γ_1 is useful in accounting for imperfect state measurements; γ_2 is useful in dealing with uncertainties/mismatches in the models used for controller synthesis; δ is useful in modifying the annotation of discrete states with atomic propositions such that, effectively, a robust specification [11] is used for discrete synthesis, which in turn guarantees correctness of continuous-time trajectories of $\mathcal{S}$ despite approximation errors.

The following definition (adapted from [27]) defines a refinement relation between two abstractions for $\mathcal{S}$.

Definition 3.2. Given two tuples of positive n-vectors $(\eta, \gamma_1, \gamma_2, \delta)$ and $(\hat{\eta}, \hat{\gamma}_1, \hat{\gamma}_2, \hat{\delta})$, let

$$\mathcal{T}_\mathcal{S} = (Q, Q_0, \mathcal{A}, \rightarrow_{\mathcal{T}_\mathcal{S}}, \Pi, \hat{L}_1)$$

be an $(\eta, \gamma_1, \gamma_2, \delta)$-abstraction of $\mathcal{S}$ and

$$\hat{\mathcal{T}}_\mathcal{S} = (\hat{Q}, \hat{Q}_0, \mathcal{A}, \rightarrow_{\hat{\mathcal{T}}_\mathcal{S}}, \Pi, \hat{L}_2)$$

be an $(\hat{\eta}, \hat{\gamma}_1, \hat{\gamma}_2, \hat{\delta})$-abstraction of $\mathcal{S}$. The transition system $\mathcal{T}_\mathcal{S}$ is said to be a refined abstraction of $\mathcal{S}$ relative to $\hat{\mathcal{T}}_\mathcal{S}$ if there exists a function $r : Q \rightarrow \hat{Q}$ such that the following conditions hold:

(i) $\cup_{q \in Q_0} \{r(q)\} \subseteq \hat{Q}_0$;

(ii) For all $q \in Q$, $\hat{L}_2(r(q)) \subseteq \hat{L}_1(q)$;

(iii) For all $(q, a, q') \in \rightarrow_{\mathcal{T}_\mathcal{S}}$, $(r(q), a, r(q')) \in \rightarrow_{\hat{\mathcal{T}}_\mathcal{S}}$.

We write $\mathcal{S} \preceq \mathcal{T}_\mathcal{S} \preceq \hat{\mathcal{T}}_\mathcal{S}$.

Note that the relations defined by Definitions 3.1 and 3.2 are different. The former defines an abstraction relation between a continuous system $\mathcal{S}$ described by (2.1) and a discrete system described by a finite transition system $\mathcal{T}_\mathcal{S}$. The latter is a refinement relation between two finite transition systems $\mathcal{T}_\mathcal{S}$ and $\hat{\mathcal{T}}_\mathcal{S}$ that are both abstractions of $\mathcal{S}$. The refinement relation essentially says that, for every transition in the refinement $\mathcal{T}_\mathcal{S}$, there is a corresponding transition in the abstract model $\hat{\mathcal{T}}_\mathcal{S}$, which is allowed to have additional transitions (i.e., more non-determinism). One motivation for computing a refinement is to reduce such non-determinism in abstractions.

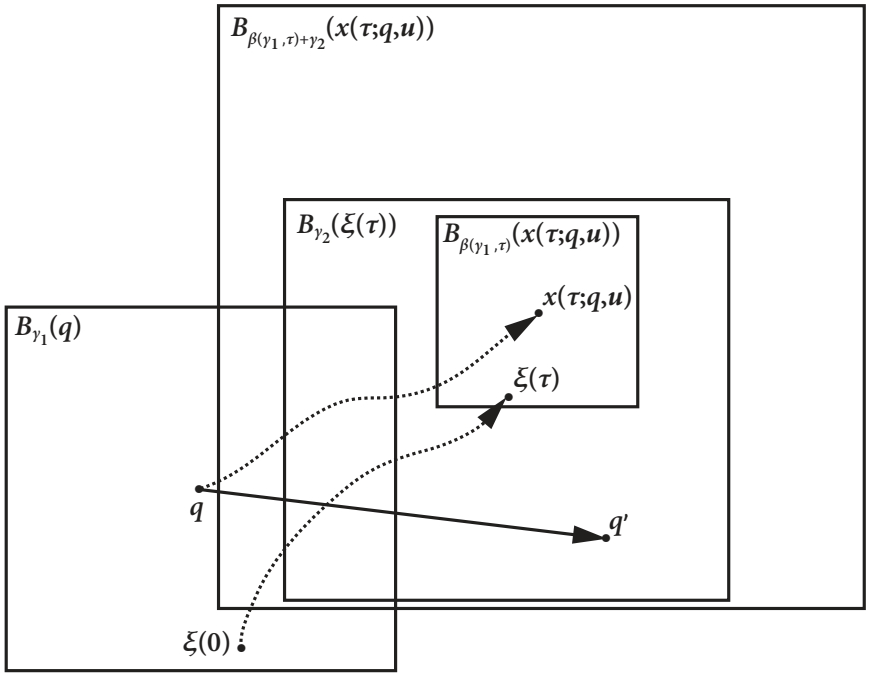

Figure 1: Illustration of Proposition 3.1.

3.1. Computation of transitions

A question that remains is how to compute the transitions in $\rightarrow_{\mathcal{T}_\mathcal{S}}$. Suppose that

$$|x(t;\mathbf{u},\xi) - x(t;\mathbf{u},\zeta)| \leq \beta(|\xi-\zeta|,t), \tag{3.2}$$

for all $\tau \in \mathbb{R}^+$, $\mathbf{u} \in U^{[0,\tau]}$, and $t \in [0,\tau]$, where $x(t;\mathbf{u},\xi)$ and $x(t;\mathbf{u},\zeta)$ denote solutions of $\dot{x} = f(x,u)$ starting from ξ and ζ and with control input $\mathbf{u}$, respectively, and $\beta : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^n$ is a continuous function such that $\beta(x,t) \leq \beta(y,t)$ for all $0 \leq x \leq y$ and $t \geq 0$. The following proposition provides a concrete way of computing transitions in an abstraction $\mathcal{T}_\mathcal{S}$ by simulating a trajectory starting from each of the point in Q and estimating the state evolution under the dynamics of (2.1).

Proposition 3.1. *Suppose (3.2) holds. If $(q,\mathbf{u},q') \in \rightarrow_{\mathcal{T}_\mathcal{S}}$ whenever $(q,\mathbf{u},q') \in Q \times \mathcal{A} \times Q$ and $|q' - x(\tau;\mathbf{u},q)| \leq \beta(\gamma_1,\tau)+\gamma_2$, then $\mathcal{S} \preceq_{(\eta,\gamma_1,\gamma_2,\delta)} \mathcal{T}_\mathcal{S}$, where $dom(\mathbf{u}) = [0,\tau]$.*

Proposition 3.1 essentially says that, for each state in $q \in Q$ and $\mathbf{u} \in \mathcal{A}$, if we add $(q,\mathbf{u},q')$ to $\rightarrow_{\mathcal{T}_\mathcal{S}}$ for each $q' \in Q \cap B_\gamma(x(\tau;\mathbf{u},q))$, where $\gamma = \beta(\gamma_1,\tau)+\gamma_2$, then we obtain an $(\eta,\gamma_1,\gamma_2,\delta)$-abstraction of $\mathcal{S}$ in the sense of Definition 3.1. Figure 1 illustrates how transitions in $\rightarrow_{\mathcal{T}_\mathcal{S}}$ can be computed in Proposition 3.1, where, in view of condition (3.2), $B_{\beta(\gamma_1,\tau)}(x(\tau;q,\mathbf{u}))$ includes $\xi(\tau)$ for all $\xi : [0,\tau] \to \mathbb{R}^n$ such that $|\xi(0)-q| \leq \gamma_1$ and $\dot{\xi}(s) = f(\xi(s),\mathbf{u}(s))$ for all $s \in [0,\tau]$. Hence, $B_{\beta(\gamma_1,\tau)+\gamma_2}(x(\tau;q,\mathbf{u}))$ contains all $q' \in Q$ that is γ_2-close to $\xi(\tau)$ for some ξ defined above; that is, all $q' \in Q$ such that $(q,\mathbf{u},q') \in \rightarrow_{\mathcal{T}_\mathcal{S}}$ as required by Definition 3.1.

Algorithm 1 outlines an abstraction procedure for computing an $(\eta,\gamma_1,\gamma_2,\delta)$-abstraction for a given continuous-time control system $\mathcal{S}$ using Proposition 3.1. Recall that the parameter η captures the granularity of the approximation. If η is given, one can choose Q by gridding the state space X according to η through different state dimensions. Though gridding is not the only option. For instance, when a partition of the state space X is given, η can be chosen to be half the length of the intervals of the smallest hyperbox that can contain each of the cells in the partition. Similarly, $\mathcal{A}$ can be chosen by gridding the input space U

and considering constant signals with a fixed duration τ but it is also possible to consider arbitrary signals of finite length. The parameters γ_1, γ_2, and δ can be chosen in view of the main results in the next section.

Algorithm 1 Computation of an $(\eta, \gamma_1, \gamma_2, \delta)$-abstraction $\mathcal{T}_S$ for a given system $\mathcal{S}$ using Proposition 3.1

Require: $\mathcal{S}, \eta, \gamma_1, \gamma_2, \delta, \beta$
1: Set $\mathcal{A}$ to be a finite subset of $\bigcup_{\tau \in \mathbb{R}^+} U^{[0,\tau]}$
2: Set Q to be a finite subset of X such that $\forall x \in X$, $\exists q \in Q$ that satisfies $|x - q| \leq \eta$
3: Set Q_0 to be the collection of all $q \in Q$ such that $\exists x \in X_0$ that satisfies $|x - q| \leq \eta$
4: Define $\hat{L}$ according to (3.1)
5: $\{\rightarrow_{\mathcal{T}_S}\} \leftarrow \varnothing$
6: **for all** $q \in Q$ **do**
7: **for all** $u \in \mathcal{A}$ **do**
8: $\tau \leftarrow \max\{\mathrm{dom}(u)\}$
9: $x' \leftarrow x(\tau; u, q)$
10: **if** $B_{\beta(\gamma_1,\tau)+\gamma_2}(x') \subseteq X$ and $q' \in B_{\beta(\gamma_1,\tau)+\gamma_2}(x')$ **then**
11: $\{\rightarrow_{\mathcal{T}_S}\} \leftarrow (q,\, u,\, q')$
12: **end if**
13: **end for**
14: **end for**
15: **return** $\{\mathcal{T}_S = (Q, Q_0, \mathcal{A}, \rightarrow_{\mathcal{T}_S}, \Pi, \hat{L})\}$

Remark 3.3. The proof of Proposition 3.1 is omitted as it is similar to the scalar case presented in [20, Proposition 1]. An explicit form of β can usually be obtained using differential inequality techniques. In addition, if the system (2.1) is incrementally asymptotically stable, β can be chosen as a $\mathcal{KL}$-function [2]. The idea of using incremental stability to construct (bi)similar symbolic models was first introduced by [28]. This has later been extended to systems under forward completeness assumptions by [43]. As noted in Remark 3.2, the key difference in the abstraction introduced in by Definition 3.1 is the robustness margins provided by the parameters γ_i $(i = 1, 2)$. In addition, using vector-valued parameters allows the use of hyper-boxes for computing abstractions and also makes the vector-valued estimate provided by (3.2) and hence Proposition 3.1 less conservative, compared with existing results. If $\gamma_1 = \gamma_2 = \eta \in \mathbb{R}^+$, Proposition 3.1 becomes a special case of Theorem 4.1 in [43].

Remark 3.4. To make the computation of transitions less conservative, one could rely on local reachable set computation of (2.1) for computing transitions in $\mathcal{T}_\mathcal{S}$. The idea is to replace a global, analytical bound on reachable sets provided by (3.2) with reachable sets computed using local dynamics and relevant sets of initial conditions (see also Section 4.4 for further discussions and [18] for some preliminary results in this direction).

3.2. Discrete synthesis

The main reason to construct a finite abstraction such as $\mathcal{T}_\mathcal{S}$ in Definition 3.1 is to formulate discrete synthesis problems that can be used to solve the continuous synthesis problem previously defined for $\mathcal{S}$. Given a set of atomic propositions Π on X, an $\mathrm{LTL}_{\setminus\bigcirc}$ formula over Π can be interpreted over executions of $\mathcal{T}_\mathcal{S}$. An *execution* of $\mathcal{T}_\mathcal{S}$ is a sequence of pairs $\rho = (q_0, a_0)(q_1, a_1)(q_2, a_2)\cdots$, where $q_0 \in Q_0$ and $(q_i, a_i, q_{i+1}) \in \rightarrow_{\mathcal{T}_\mathcal{S}}$ for all $i \geq 0$. A *control strategy* for $\mathcal{T}_\mathcal{S}$ is a partial function $\mu : (q_0, \cdots, q_i) \mapsto a_i$ that maps the sequence of states up to q_i to an action a_i. A *μ-controlled execution* of $\mathcal{T}_\mathcal{S}$ is an execution of $\mathcal{T}_\mathcal{S}$, where for each $i \geq 0$, the action a_i is chosen according to the control strategy μ.

We now formulate the discrete synthesis problem as follows.

Discrete Synthesis Problem: Given the transition system $\mathcal{T}_\mathcal{S}$ and an LTL$_{\setminus\bigcirc}$ specification φ, find a control strategy μ such that all μ-controlled executions of $\mathcal{T}_\mathcal{S}$ satisfy φ for all initial conditions in Q_0.

If there exists such a control strategy μ, we say that φ is *realizable* for $\mathcal{T}_\mathcal{S}$. The discrete synthesis problem can be recast as a temporal logic game. While the complexity of solving general LTL games can be of high complexity, efficient solvers exist for solving such games with expressive fragments of LTL (e.g., [5, 9]), together with publicly available toolboxes (e.g., [41]).

Augmented Progress Properties. In view of comments after Proposition 3.1, if τ is sufficiently small compared with γ, then the ball $B_\gamma(x(\tau;\mathbf{u},q))$ will almost always include q itself, which introduces a self-transition $(q,\mathbf{u},q)$ for almost all $q \in Q$. As we shall treat all non-determinism as adversary when solving the discrete synthesis problem, these self-transitions can render the problem unrealizable if the specification involves making progress. In addition to self-transitions, non-determinism can potentially induce spurious cyclic executions in the abstract system that do not exist in the continuous system (2.1). To deal with these issues, we can use augmented finite transition systems [27] to enforce additional progress assumptions when solving the discrete synthesis problem. Such progress assumptions can be captured by the following LTL$_{\setminus\bigcirc}$ formula:

$$\varphi_g \doteq \bigwedge_{\mathbf{u}\in\mathcal{A}} \bigwedge_{G\in\mathcal{G}(\mathbf{u})} \neg\Diamond\Box((\ \vee_{\ q\in G} q)\ \wedge\ \mathbf{u}), \tag{3.3}$$

over an extended set of atomic propositions[2], where each $G \in \mathcal{G}(\mathbf{u})$ represents a progress group. That is, the set $\bigcup_{q\in G} \alpha^{-1}(q)$ does not contain any invariant sets for system (2.1) when a fixed $\mathbf{u}$ is repeatedly executed. Such progress groups can be trivially computed for affine or incrementally stable dynamics. It is also possible to approximate them when the dynamics are polynomial [27]. Appropriately encoding these progress properties is essential for achieving certain specifications (e.g., reachability).

3.3. Continuous implementations of discrete strategies

For the finite abstractions $\mathcal{T}_\mathcal{S}$ to be useful, we need to guarantee two things: (i) every discrete control strategy for $\mathcal{T}_\mathcal{S}$ can be implemented to form a control strategy for the continuous system $\mathcal{S}$; and (ii) if the discrete strategy solves the discrete synthesis problem for $\mathcal{T}_\mathcal{S}$, then the corresponding continuous strategy solves the continuous synthesis problem for $\mathcal{S}$. Establishing these under various scenarios will be the main results of this paper. The following definition and proposition are useful in mapping a discrete strategy to a continuous one.

Given a discrete strategy μ for $\mathcal{T}_\mathcal{S}$, we call a continuous strategy σ for $\mathcal{S}$ an *implementation* of μ if

$$\sigma(x_0,\cdots,x_i) = \mu(q_0,\cdots,q_i) = a_i \in \mathcal{A},\ \forall i = 0,1,2,\cdots,$$

where, for all $i \geq 0$, $q_i \in \alpha(x_i)$, $(q_i,a_i,q_{i+1}) \in \rightarrow_{\mathcal{T}_\mathcal{S}}$, and there exists $\xi : [0,\Delta_i] \to \mathbb{R}^n$ such that $\xi(0) = x_i$, $\xi(\Delta_i) = x_{i+1}$, and $\dot{\xi}(s) = f(\xi(s),a_i(s))$ for all $s \in [0,\Delta_i]$, where $a_i \in \mathcal{U}^{[0,\Delta_i]}$ for some Δ_i.

Proposition 3.2. *If $\mathcal{S} \preceq_{(\eta,\gamma_1,\gamma_2,\delta)} \mathcal{T}_\mathcal{S}$, then every control strategy for $\mathcal{T}_\mathcal{S}$ can be implemented on $\mathcal{S}$.*

Proof. For each control strategy μ for $\mathcal{T}_\mathcal{S}$, we define a control strategy σ for $\mathcal{S}$ as follows: starting from an initial state $x_0 \in X_0$, it follows from (i) of Definition 3.1 that there exists $q_0 \in Q_0$ such that $q_0 \in \alpha(x_0)$. We can define $\sigma(x_0) = \mu(q_0) = a_0 \in \mathcal{A}$. Note that a_0 is indeed a control signal in $\mathcal{U}^{[0,\Delta_0]}$ for some $\Delta_0 \geq 0$. Let x_1 be such that there exists $\xi : [0,\Delta_0] \to \mathbb{R}^n$ such that $\xi(0) = x_0$, $\xi(\Delta_i) = x_1$, and $\dot{\xi}(s) = f(\xi(s),a_0(s))$ for all $s \in [0,\Delta_0]$. Choose any $q_1 \in \alpha(x_1)$. Then (ii) of Definition 3.1 implies that $|x_0 - q_0| \leq \eta \leq \gamma_1$ and $|x_1 - q_1| \leq \eta \leq \gamma_2$. It follows from (iii) of Definition 3.1 that $(q_0,a_0,q_1) \in \rightarrow_{\mathcal{T}_\mathcal{S}}$. Repeating this procedure for all $i \geq 0$, an implementation can be defined. □

[2]To be precise, we define a concatenated labeling function L' as $L'(q,\mathbf{u}) \doteq \hat{L}(q) \cup q \cup \mathbf{u}$ and interpret the LTL formula $\varphi \wedge \varphi_g$ over state-action sequences, where φ is the original specification (see [27] for details).

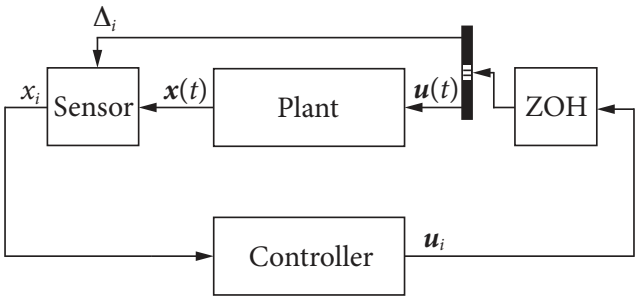

Figure 2: Illustration of a digital implementation of a control strategy: a sensor takes (sampled) state measurements from a plant; as a control strategy of the form (2.2), the controller outputs a new control input signal to the plant based on a sequence of sampled states; state measurements are taken by the sensor at time instants (sampling times) decided by the current control inputs at run time; a control input signal $\mathbf{u}_i$ consists of a constant input value u_i and a duration Δ_i

.

The following proposition states that, if a specification is realizable on an abstract model, then it is also realizable on its refinement.

Proposition 3.3. *Given $\mathcal{S} \preceq \mathcal{T}_\mathcal{S} \preceq \hat{\mathcal{T}}_\mathcal{S}$ and an LTL$_{\setminus\bigcirc}$ formula φ, φ being realizable for $\hat{\mathcal{T}}_\mathcal{S}$ implies that φ is realizable for $\mathcal{T}_\mathcal{S}$.*

Proof. The proof consists of two parts. The first part is to show that each control strategy $\hat{\mu}$ for $\hat{\mathcal{T}}_\mathcal{S}$ can be refined as a control strategy on $\mathcal{T}_\mathcal{S}$. The second part is proving that controlled executions under the refined strategy satisfy the given specification. For each control strategy $\hat{\mu}$ for $\hat{\mathcal{T}}_\mathcal{S}$, we define a control strategy μ for $\mathcal{T}_\mathcal{S}$ as follows: starting from an initial state $q_0 \in \mathcal{Q}$, it follows from (iii) of Definition 3.2 that $r(q_0) \in \hat{Q}_0$. We can define $\mu(q_0) = \hat{\mu}(r(q_0)) = a_0 \in \hat{A} = A$. Let q_1 be such that $(q_0, a_0, q_1) \in \rightarrow_{\mathcal{T}_\mathcal{S}}$ and $\hat{q}_1 = r(q_1)$. Then (iv) of Definition 3.2 implies that $(r(q_0), a_0, r(q_1)) \in \rightarrow_{\hat{\mathcal{T}}_\mathcal{S}}$. Repeating this procedure for all $i \geq 0$ proves the first part. Now let $\hat{\mu}$ be a strategy for $\hat{\mathcal{T}}_\mathcal{S}$ such that all $\hat{\mu}$-controlled executions of $\hat{\mathcal{T}}_\mathcal{S}$ satisfy φ. Let μ be a refined strategy for $\mathcal{T}_\mathcal{S}$ defined above. Then all μ-controlled executions of $\mathcal{T}_\mathcal{S}$ have corresponding $\hat{\mu}$-controlled executions of $\hat{\mathcal{T}}_\mathcal{S}$ under the map $r : Q \to \hat{Q}$. The correctness of these executions with respect to φ follow from (ii) of Definition 3.2 and induction on LTL$_{\setminus\bigcirc}$ formulas in Negation Normal Form (similar to the induction procedure in the proof of Theorem 4.1). □

In practice, a continuous strategy σ for $\mathcal{S}$ is digitally implemented for system (2.1) in the sense that a sampled sequence $x(\tau_0), \cdots, x(\tau_i)$ taken at sampling times $\tau_0 = 0, \cdots, \tau_i$ is regarded as a state sequence $x_0, \cdots, x_i$ in X, where the control input signal required by (2.2) at τ_i is given by $\sigma(x_0, \cdots, x_i)$ for all $i \geq 0$ and also takes some digital form, e.g., zero-order hold (ZOH). Figure 2 gives a schematic illustration of a digital implementation of a control strategy that leads to closed-loop control for the continuous-time system (2.1). Here we consider the case with perfect state measurements and assume that there are no delays in the control loop. Implementations on systems with imperfect state measurements and delays will be discussed in Section 4.2.

4. Main results—Implications of the robustness margins

The main objective of this section is to show that, with the notion of abstraction given in Definition 3.1, we are able to reason about the qualitative properties of solutions of a continuous-time control system in different practical scenarios.

4.1. Continuous correctness by discrete reasoning

When implementing a discrete strategy, perhaps obtained from solving a discrete synthesis problem, we are effectively implementing a hybrid feedback controller such that solutions of a system $\mathcal{S}$ satisfy a given specification.

In general, the existence of a discrete control strategy for the discrete synthesis problem for $\mathcal{T}_\mathcal{S}$ with an $\text{LTL}_{\setminus\bigcirc}$ formula φ *does not* guarantee the existence of a control strategy that solves the continuous synthesis problem for the system $\mathcal{S}$ with the same specification φ. In fact, using discretization-based (or grid point-based), rather than proposition-preserving partition-based, abstractions, we need extra conditions to ensure correctness of continuous executions from discrete reasoning. This motivates condition (3.1) in defining abstractions, which essentially captures the idea of contracting and expanding atomic propositions as used in [10, 22]. This extra condition is needed to account for inter-sample behaviors as pointed out in [20, Example 2].

Let $M = (M_1, \cdots, M_n)$ be an n-vector defined by $M_i = \sup_{x\in X, u\in U} |f_i(x,u)|$ $(1 \le i \le n)$ and Δ be the maximum duration of actions in $\mathcal{A}$.

Theorem 4.1. *If $\mathcal{S} \preceq_{(\eta,\gamma_1,\gamma_2,\delta)} \mathcal{T}_\mathcal{S}$ and $\delta \ge M\Delta/2 + \eta$, then, given any $\text{LTL}_{\setminus\bigcirc}$ formula φ, φ being realizable for $\mathcal{T}_\mathcal{S}$ implies that φ is realizable for $\mathcal{S}$.*

Proof. By Proposition 3.2, to every control strategy μ for $\mathcal{T}_\mathcal{S}$, there corresponds a control strategy σ (which is an implementation of μ) for $\mathcal{S}$ such that, to each σ-controlled solution of $\mathcal{S}$, there corresponds a μ-controlled execution in $\mathcal{T}_\mathcal{S}$. We denote this correspondence by x to ρ, where x is a solution of $\mathcal{S}$ resulting from σ and $\rho = (q_0, \mathbf{u}_0)(q_1, \mathbf{u}_1)(q_2, \mathbf{u}_2)\cdots$. Here each q_i is an abstract state corresponding to $x_i = x(\tau_i)$, for all $i \ge 0$, where $\tau_0 = 0$ and $\tau_{i+1} = \tau_i + \Delta_i$, where Δ_i is the duration of $\mathbf{u}_i$. We have $|x_i - q_i| \le \eta$ for all $i \ge 0$. Furthermore, we have to show that $\rho \vDash \varphi$ implies $x \vDash \varphi$. We prove this by proving a stronger statement: $\rho, i \vDash \varphi$ for $i \ge 0$ implies $x, t \vDash \varphi$ for all $t \in J_i = [\tau_i - \Delta/2, \tau_i + \Delta/2] \cap \mathbb{R}^+$.

The proof is by induction on the structure of an $\text{LTL}_{\setminus\bigcirc}$ formula.

Case $\varphi = \pi$: To show $x, t \vDash \pi$ for all $t \in J_i$, we have to show that $\pi \in L(x(t))$. This follows from $x_i = x(\tau_i)$, $\pi \in \hat{L}(q_i)$, and

$$|x(t) - q_i| \le |x(t) - x(\tau_i)| + |x_i - q_i| \le M\Delta/2 + \eta \le \delta. \tag{4.1}$$

Case $\varphi = \varphi_1 \mathcal{U} \varphi_2$: To show $x, t \vDash \varphi$ for all $t \in J_i$, we need to show that, for each fixed $t \in J_i$, there exists $t' \ge 0$ such that $x, t + t' \vDash \varphi_2$ and for all $t'' \in [0, t')$, $x, t + t'' \vDash \varphi_1$. We have $\rho, i \vDash \varphi$; that is, there exists $j > i$ such that $\rho, j \vDash \varphi_2$ and $\rho, k \vDash \varphi_1$ for all $k \in [i, j)$. It follows from the inductive assumption that $x, s \vDash \varphi_2$ for all $s \in J_j$ and $x, s \vDash \varphi_1$ for all $s \in J_k$ and all $k \in [i, j)$. Take $t' = \max(\tau_j - \Delta/2, t) - t$. Then $t + t' \in J_j$ and hence $x, t + t' \vDash \varphi_2$. In addition, for all $t'' \in [0, t')$, we have $t + t'' \in J_k$ for some $k \in [i, j)$ and hence $x, t + t'' \vDash \varphi_1$. In fact, $\cup_{i \le k \le j-1} J_k = [\tau_i - \Delta/2, \tau_{j-1} + \Delta/2] \cap \mathbb{R}^+ \supseteq [t, \tau_j - \Delta/2) = [t, t + t') \ni t + t''$.

Case $\varphi = \varphi_1 \mathcal{R} \varphi_2$: To show $x(t) \vDash \varphi$ for all $t \in J_i$, we need to show that, for each fixed $t \in J_i$, we have, for all $t' \ge 0$ either $x, t + t' \vDash \varphi_2$ or that there exists $t'' \in [0, t')$ such that $x, t + t'' \vDash \varphi_1$. We have $\rho, i \vDash \varphi$; that is, for all $j \ge i$, either $\rho, j \vDash \varphi_2$ or there exists $k \in [i, j)$ such that $\rho, k \vDash \varphi_1$. Given $t' \ge 0$, let τ_j be such that $t + t' \in J_j$, where $j \ge i$. For this j, we have either $\rho, j \vDash \varphi_2$ or that there exists $k \in [i, j)$ such that $\rho, k \vDash \varphi_1$. It follows from the inductive assumption that either $x, s \vDash \varphi_2$ for all $s \in J_j$ or there exists $k \in [i, j)$ such that $x, s \vDash \varphi_1$ for all $s \in J_k$. If the former holds, since $t + t' \in J_j$, we get $x, t + t' \vDash \varphi_2$. If the latter holds, since $t + t' \ge \tau_j - \Delta/2 > \tau_k - \Delta/2$ and $\tau_k + \Delta/2 \ge \tau_i + \Delta/2 \ge t$, we know $[t, t + t') \cap J_k \ne \emptyset$. Thus, there exists $t'' \in [0, t')$ such that $x, t + t'' \vDash \varphi_1$.

The other cases are straightforward. □

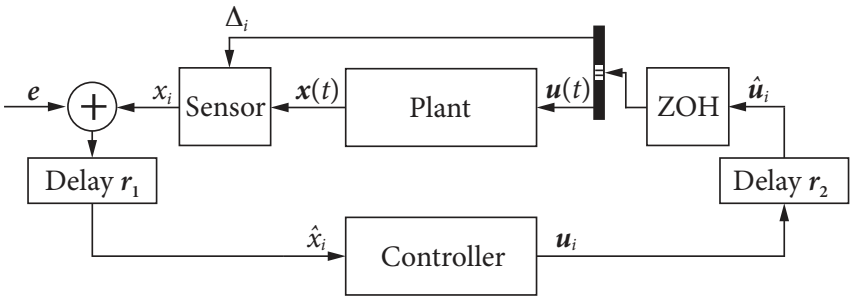

Figure 3: Illustration of a controller that takes delayed (by r_1) and imperfect measurement (subject to measurement errors bounded by ε) from a plant and sends a control input that is received and actuated by the plant after another delay r_2 (measured from when the controller receives the measurement and to when the control input has been actuated by the plant). The total round-trip delay $r_1 + r_2$ is not assumed to be constant, but assumed to be bounded by some constant r. Each computed control input $\mathbf{u}_i$ is a pair consisting of a constant input value u_i and a duration Δ_i. State measurements are taken at time instants decided by the current control inputs at run time. While the plant is waiting for the next control input, it keeps on executing the previous one.

4.2. Imperfect state measurement: bounded errors or delays

In practice, state measurements are not perfect, often subject to measurement noise or quantization. Furthermore, delays are ubiquitous in control systems, for instance, leading to jitter [7]. In this subsection, we consider the robustness of a hybrid controller for the system (2.1) that realizes a temporal logic objective with respect to imperfect state measurements.

Given real constants $r \geq 0$ and $\varepsilon \geq 0$ and the system $\mathcal{S} = (X, X_0, U, f, \Pi, L)$, let $\mathcal{S}_{r,\varepsilon}$ denote a system whose state evolves according to

$$\begin{cases} \dot{x}(t) = f(x, u(t - r_2)), \\ \hat{x}(t) = x(t - r_1) + e(t), \end{cases} \tag{4.2}$$

where $\hat{x}(t)$ is the state measurement received at t, r_1 is the delay for a measurement taken from the plant to reach the controller, r_2 is the delay a control input computed by the controller is received by the plant, and $|e(t)| \leq \varepsilon$ is a time-varying measurement error bounded by ε. Let $r_1 + r_2$ be the round-trip delay, which is assumed to be upper bounded by r. The objective is to design and implement a hybrid controller on system (4.2) with correctness guarantees, based on abstractions with robustness margins of the nominal system (2.1). The details of the problem are further illustrated in Figure 3.

Theorem 4.2. *If $\mathcal{S} \preceq_{(\eta,\gamma_1,\gamma_2,\delta)} \mathcal{T}_\mathcal{S}$ with $\gamma_1 \geq rM + \varepsilon + \eta$, $\gamma_2 \geq \varepsilon + \eta$, and $\delta \geq (r + \Delta)M/2 + (\varepsilon + \eta)$, then, given any LTL$_{\setminus\bigcirc}$ formula φ, φ being realizable for $\mathcal{T}_\mathcal{S}$ implies that φ is realizable for $\mathcal{S}_{r,\varepsilon}$.*

Proof. Let $x_0, x_1, x_2, \cdots$, be the measurements taken at the plant at times $\tau_0, \tau_1, \tau_2, \cdots$; that is $x_i = x(\tau_i)$ for all $i \geq 0$. As shown in Figure 3, we assume it takes time delay $r_{1,i}$ for the hybrid controller to receive a perhaps noisy measurement given by $\hat{x}_i = x(\tau_i) + e_i$ at time $\tau_i + r_{1,i}$. The controller makes a decision and passes on a suggested input $\mathbf{u}_i$ (which includes the duration of $\mathbf{u}_i$ denoted by Δ_i). The plant will receive this input subject to another delay $r_{2,i}$. As $r_{1,i}$ and $r_{2,i}$ can be time varying, we use r_i to indicate the round-trip delay $r_{1,i} + r_{2,i}$ experienced after the ith sample is taken. Note that r_i is upper bounded by the constant r. In other words, after a sample is taken at τ_i, the plant will receive an updated control input at time $\tau_i + r_i$. From this point on, the control input is set to $\mathbf{u}_i$. Between τ_i and $\tau_i + r_i$, the plant will keep executing the previous input $\mathbf{u}_{i-1}$; initially, between τ_0 and $\tau_0 + r_0$, assume this input is set to some initial value. We need to be clear how τ_i's are defined: we set $\tau_0 = 0$ and the rest of the sampling times τ_i ($i \geq 1$) are defined by $\tau_i = \tau_{i-1} + \Delta_{i-1} + r_{i-1}$.

There are two things to prove: (i) every measured state (despite delays and noise) are accounted for in the abstraction, so that the discrete control strategy can be implemented. Put more straightforwardly, every measured state should be expected by the controller so that it can make a decision based on the strategy automaton; (ii) the plant trajectory $x(t)$, $t \geq 0$, should satisfy the desired specification φ.

To show (i), we need to verify that there exists a trajectory ξ of (2.1) under inputs $\mathbf{u}_i$ such that $|\xi(0) - q_i| \leq \gamma_1$ and $|\xi(\Delta_i) - q_{i+1}| \leq \gamma_2$, where q_i and q_{i+1} are such that $|\hat{x}_i - q_i| \leq \eta$ and $|\hat{x}_{i+1} - q_{i+1}| \leq \eta$. We know that $|\hat{x}_i - x(\tau_i)| \leq \varepsilon$, $\tau_{i+1} = \tau_i + \Delta_i + r_i$ for all $i \geq 0$, and $\mathbf{u}_i$ is activated on $[\tau_i + r_i, \tau_i + r_i + \Delta_i]$. Letting $\xi(s) = x(\tau_i + r_i + s)$ for $s \in [0, \Delta_i]$, then $\xi(0) = x(\tau_i + r_i)$ and $\xi(\Delta_i) = x(\tau_i + \Delta_i + r_i)$. It is easy to verify that $|\xi(0) - \hat{q}_i| \leq |x(\tau_i + r_i) - x(\tau_i)| + |x(\tau_i) - \hat{x}_i| + |\hat{x}_i - q_i| \leq rM + \varepsilon + \eta \leq \gamma_1$ and $|\xi(\Delta_i) - q_{i+1}| \leq |x(\tau_{i+1}) - \hat{x}_{i+1}| + |\hat{x}_{i+1} - q_{i+1}| \leq \varepsilon + \eta \leq \gamma_2$.

Let $x(\tau_i) = x_i$. We have $|x_i - q_i| \leq \varepsilon + \eta$ and $\tau_{i+1} - \tau_i = \Delta_i + r_i$ for all $i \geq 0$. Based on $\delta \geq (r + \Delta)M/2 + (\eta + \varepsilon)$, we can prove $x \vDash \varphi$ following the proof of Theorem 4.1 with η replaced by $\eta + \varepsilon$ and Δ_i replaced by $\Delta_i + r_i$. □

4.3. Unmodeled dynamics: bounded disturbance or delays

We can also apply the same methodology to prove the effectiveness of the design in the situation where systems (2.1) contain unmodeled dynamics that can be treated as bounded disturbance in the right-hand side of (2.1).

A general time-delay system can be written as a functional differential equation:

$$\dot{x} = F(x_t, u), \quad t \geq 0, \tag{4.3}$$

where $F : \mathcal{C}_r \times U \to \mathbb{R}^n$ is a functional with control input $u \in U$, and $x_t(s) = x(t + s)$ for all $s \in [-r, 0]$. We assume that, given any initial condition $x_0 \in \mathcal{C}_r$, (4.3) has a unique solution (see, e.g., [14, Chapter 2] for regularity conditions on F that guarantee this).

We can rewrite F such that it has an *ordinary part* and a *functional part*:

$$F(x_t, u) = f(x, u) + g(x_t, u), \tag{4.4}$$

where $f : \mathbb{R}^+ \times U \to \mathbb{R}^n$ and $g : \mathcal{C}_r \times U \to \mathbb{R}^n$. This form can be obtained, for example, from (4.3) by letting $g(x_t, u) := F(x_t, u) - f(x, u)$. The idea is to design controllers for system (4.3), based on the delay-free model (2.1). The results rely on the following assumption:

Assumption (Boundedness). There exists a positive n-vector $D_r > 0$ such that $|g(x_t, u)| \leq D_r$ for all $u \in U$ and all solutions x_t of (4.3) that completely lies in X; that is, $x_t(s) \in X$ for all $s \in [-r, 0]$. □

In most delay models, $D_r \to 0$ as $r \to 0$ for compact sets X and U. We will treat $g(x_t, u)$ as additive disturbances to the right-hand side of (2.1). Therefore, the results also work for general disturbances satisfying a similar boundedness assumption. Similar to that for previous results, we let M be an n-vector such that $|F(x_t, u)| \leq M$ for all $u \in U$ and all solutions x_t of (4.3) that completely lies in X.

Given the system $\mathcal{S} = (X, X_0, U, f, \Pi, L)$ and F defined above satisfying (4.4) and the boundedness assumption, let $\mathcal{S}_F$ denote the system whose state evolves according to (4.3).

Theorem 4.3. *If $\mathcal{S} \preceq_{(\eta, \gamma_1, \gamma_2, \delta)} \mathcal{T}_\mathcal{S}$ with $\gamma_1 \geq \eta$, $\gamma_2 \geq E + \eta$, where $E = (E_1, \cdots, E_n)$ is a n-vector given by $E_i = (e^{K_i \Delta} - 1)(D_r)_i / K_i$ and $K_i > 0$ is the uniform Lipschitz constant of $f_i(\cdot, u)$ on X for all $u \in U$, $(D_r)_i$ is the ith component of D_r, and $\delta \geq \Delta M/2 + \eta$, then, given any LTL$_{\setminus\bigcirc}$ formula φ, φ being realizable for $\mathcal{T}_\mathcal{S}$ implies that φ is realizable for $\mathcal{S}_F$.*

Proof. Let $x_0, x_1, \cdots$, be the measurements taken for the system (4.3) at times $\tau_0, \tau_1, \cdots$; that is $x_k = x(\tau_k)$ for all $k \geq 0$, where $\tau_0 = 0$ and $\tau_{k+1} = \tau_k + \Delta_k$ for all $k \geq 0$ and $\mathbf{u}_k$ is activated on $[\tau_k, \tau_k + \Delta_k]$ for each $k \geq 0$. We need to ensure that the abstraction computed for $\mathcal{S}$ based on (2.1) actually accounts for

all possible behaviors of solutions of (4.3). To do so, we need to verify that there exists a trajectory ξ of (2.1) under inputs $\mathbf{u}_k$ such that $|\xi(0) - q_k| \leq \gamma_1$ and $|\xi(\Delta_k) - q_{k+1}| \leq \gamma_2$, where q_k and q_{k+1} are such that $|x_k - q_k| \leq \eta$ and $|x_{k+1} - q_{k+1}| \leq \eta$. Let ξ be a solution of (2.1) starting from x_k. We have $\xi(0) = \xi(\tau_k)$ and $\dot{\xi}(s) = f(\xi(s), \mathbf{u}_k(s))$ for $s \in [0, \Delta_k]$. Define $y(s) = x(\tau_k + s)$ for $s \in [-r, \Delta_k]$. Then $y(0) = x(\tau_k)$ and $\dot{y}(s) = F(y_s, \mathbf{u}_k(s)) = f(y(s), \mathbf{u}_k(s)) + g(y_s, \mathbf{u}_k(s))$ for $s \in [0, \Delta_k]$. Let $z(s) = y(s) - \xi(s)$ for $s \in [-r, \Delta_k]$. It follows that $|\dot{z}| \leq K|z| + D_r$, where $K = (K_1, \cdots, K_n)$, and $z(0) = 0$. It follows that $|z_i(s)| \leq (e^{K_i s} - 1)(D_r)_i / K_i$ for $s \in [0, \Delta_k]$ and $1 \leq i \leq n$. Therefore, $|\xi(0) - q_k| \leq |z(0)| + |x(\tau_k) - q_k| \leq \eta \leq \gamma_1$ and $|\xi(\Delta_k) - q_{k+1}| \leq |z(\Delta_k)| + |x(\tau_k + \Delta_k) - q_{k+1}| \leq E + \eta \leq \gamma_2$. □

4.4. Discussions

The main results presented earlier in this section provide sufficient conditions on the soundness of abstractions for robust control design from temporal logic specifications, in terms of the abstraction parameters used, system dynamics, and the size and source of uncertainties. We give below a brief account of the advantages, limitations, and trade-offs of the proposed results.

One of the key features of the approach is that we compute abstractions based on a nominal control system model and equip them with robustness margins to account for uncertainties that we may not have specific *a priori* knowledge of at the design time. With the results discussed earlier, we can guarantee that the synthesized controllers are robust against various uncertainties within certain bounds, if such controllers can be founded via discrete synthesis using robust abstractions proposed in this paper. The quantitative nature of the robustness margins also allows us to study potential trade-offs between performance and robustness (see the example in Section 6).

We would like to point out that, due to the conservative nature of the abstractions we have considered in this paper, the unrealizability of a discrete synthesis problem for a given specification does not preclude the possibility of the existence of concrete controllers that realize this specification. In fact, one of the motivations of defining quantitative robustness margins is to be able to discuss trade-offs between robustness and performance, as discussed above and to be illustrated with the example in Section 6 (Figure 4). While adding more transitions in the abstraction will certainly make a synthesized controller (if one can be found) more robust, but it may also render the discrete synthesis unrealizable.

It is worth emphasizing some improvements with respect to our preliminary results on abstractions with robustness margins presented in [20], where scalar parameters were used in the definition of the abstraction. For instance, when non-uniform grids are used for selecting the discrete states of the abstraction, using vector-valued parameters leads to significantly less transitions. Moreover, if the dynamics are faster in one dimension compared to the others, a scalar-valued β function potentially leads to a lot of spurious transition, whereas a vector-valued β function leads to less conservatism. Finally, as opposed to [20], the vector-valued parameters in the abstractions proposed in this paper allow the uncertainties like measurement noise to have a different bound for different state variables. This is important in realistic scenarios where sensors measuring different states might have different accuracies and trying to bound the measurement error with a scalar would lead to unnecessary conservatism.

Future work will investigate how to further reduce the number of spurious transitions in computing robust abstractions. One way to do so is to replace the various global inequalities in Theorems 4.1–4.3 with local versions; that is, the inequalities hold for every single transition. This will potentially require using state-dependent abstraction parameters, considering bounds on dynamics and uncertainties locally, taking into account the size of each individual control actions rather than using a uniform upper bound.

We also plan to study refinement procedures to find a discrete controller if one exists. In particular, we would like to investigate under what assumptions on the system dynamics (e.g., incremental stability) and the type and size of uncertainties we would be able to prove certain completeness of the notion of robust abstractions proposed in this paper.

5. Abstractions with robustness margins for discrete-time control systems

In this section, we show that abstractions with robustness margins can be similarly defined for discrete-time control systems. We start with a few definitions that are parallel to those introduced for continuous-time systems.

A discrete-time control system is a tuple $\mathcal{S}_d = (X, X_0, U, g, \Pi, L)$ where $X \subseteq \mathbb{R}^n$ is a set of states, $X_0 \subseteq X$ is a set of initial states, $U \subseteq \mathbb{R}^m$ is a set of inputs, g is a function from $\mathbb{R}^n \times \mathbb{R}^m$ to $\mathbb{R}^n$, Π is a set of atomic propositions and $L : X \to 2^\Pi$ is a labeling function. The state evolves according to:

$$x^+ = g(x, u), \tag{5.1}$$

where x^+ denotes the next state of x under the above difference equation.

A *control strategy* for $\mathcal{S}_d$ is defined to be a partial function of the form:

$$\sigma(x_0, \cdots, x_i) = u_i \in U, \ \forall i = 0, 1, 2, \cdots. \tag{5.2}$$

A synthesis problem for $\mathcal{S}_d$ can be formulated as follows: Given $\mathcal{S}_d$ and an $\text{LTL}_{\setminus\bigcirc}$ specification φ over Π, find a control strategy for $\mathcal{S}_d$ such that all of its solutions satisfy φ for all initial conditions in X_0. If there exists such a control strategy, we say that φ is *realizable* for $\mathcal{S}_d$.

Definition 5.1. Given the discrete-time control system $\mathcal{S}_d = (X, X_0, U, g, \Pi, L)$ and a tuple of positive n-vectors $(\eta, \gamma_1, \gamma_2, \delta)$ satisfying $\gamma_i \geq \eta$ $(i = 1, 2)$ and $\delta \geq \eta$, a finite transition system

$$\mathcal{T}_{\mathcal{S}_d} = (Q, Q_0, \mathcal{A}, \to_{\mathcal{T}_S}, \Pi, \hat{L})$$

is called an $(\eta, \gamma_1, \gamma_2, \delta)$-*abstraction* of $\mathcal{S}_d$ if there exists an abstraction map $\alpha : X \to 2^Q$ such that

(i) Q and $\mathcal{A}$ are finite subsets of X and U, respectively, and $\bigcup_{x \in X_0} \alpha(x) \subseteq Q_0$;

(ii) $|x - q| \leq \eta$ for all $(x, q) \in X \times Q$ such that $q \in \alpha(x)$;

(iii) $(q, u, q') \in \to_{\mathcal{T}_{\mathcal{S}_d}}$ if there exists ξ and ξ' such that $|\xi - q| \leq \gamma_1$, $|\xi' - q'| \leq \gamma_2$, and $\xi' = g(\xi, u)$,

(iv) $\hat{L} : Q \to 2^\Pi$ is defined by

$$\pi \in \hat{L}(q), \ q \in Q \iff \pi \in L(x), \ \forall x \in B_\delta(q). \tag{5.3}$$

We write $\mathcal{S}_d \preceq_{(\eta, \gamma_1, \gamma_2, \delta)} \mathcal{T}_{\mathcal{S}_d}$.

Given two abstractions with robustness margins for $\mathcal{S}_d$, a refinement relation between them can be defined following exactly Definition 3.2.

Similar to Proposition 3.1, we have the following result for computing transitions of $\mathcal{T}_{\mathcal{S}_d}$. To state it, we replace (3.2) with the following condition:

$$|g(u, \xi) - g(u, \zeta)| \leq \beta(|\xi - \zeta|), \tag{5.4}$$

where $u \in U$ and $\beta : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous function.

Proposition 5.1. *Suppose (5.4) holds. If $(q, u, q') \in \to_{\mathcal{T}_d}$ whenever $(q, u, q') \in Q \times \mathcal{A} \times Q$ and $|q' - g(u, q)| \leq \beta(\gamma_1) + \gamma_2$, then $\mathcal{S}_d \preceq_{(\eta, \gamma_1, \gamma_2, \delta)} \mathcal{T}_{\mathcal{S}_d}$.*

For discrete-time systems, we do not consider delays in this paper, but the following results give robustness with respect to measurement errors and bounded additive disturbances, respectively.

Theorem 5.1. *Suppose that $\mathcal{S}_d$ is to be controlled subject to measurement errors bounded by ε. If $\mathcal{S}_d \preceq_{(\eta,\gamma_1,\gamma_2,\delta)} \mathcal{T}_{\mathcal{S}_d}$ with $\gamma_i \geq \varepsilon + \eta$ $(i = 1, 2)$, and $\delta \geq \varepsilon + \eta$, then, given any LTL$_{\setminus\bigcirc}$ formula φ, φ being realizable for $\mathcal{T}_{\mathcal{S}_d}$ implies that φ is realizable for $\mathcal{S}_d$.*

Theorem 5.2. *Suppose that $\mathcal{S}_d$ is subject to additive disturbances bounded by D. If $\mathcal{S}_d \preceq_{(\eta,\gamma_1,\gamma_2,\delta)} \mathcal{T}_{\mathcal{S}_d}$ with $\gamma_1 \geq \eta$, $\gamma_2 \geq D + \eta$, and $\delta \geq \eta$, then, given any LTL$_{\setminus\bigcirc}$ formula φ, φ being realizable for $\mathcal{T}_{\mathcal{S}_d}$ implies that φ is realizable for $\mathcal{S}_d$.*

The proofs for the above results are similar to that for Theorems 4.3 and 4.2. The only difference is that we do not need to account for inter-sample behaviors. Hence, the conditions are weakened accordingly.

5.1. Justification of time-discretization-based design

There are situations one would like to use a time-discretized model to design controllers for a continuous-time system, for example, when there is already a design methodology proved to be effective for discretized systems. What are the issues that need to be considered to ensure the performance of the resulted controller? This is a standard question in the design of stabilizing controllers (e.g., [12]). Here we consider it in the context of hybrid control for temporal logic objectives.

Let $\mathcal{S}_d = (X, X_0, U, g, \Pi, L)$ be a time-discretized model for $\mathcal{S} = (X, X_0, U, f, \Pi, L)$, which could be an exact model (e.g., available in the case where f is linear) or an approximate model (such as obtained from applying a numerical scheme). For example, $g(x, u)$ can be defined by $g(x, u) = x + \Delta f(x, u)$ as in a forward Euler scheme with a constant step size Δ. We only consider the case of constant step size and write the time-discretized control system as

$$x^+ = g_\Delta(x, u), \tag{5.5}$$

where $x \in X \subseteq \mathbb{R}^n$ and $u \in U \subseteq \mathbb{R}^m$ and g_Δ is a suitable one-step numerical scheme with a constant step size Δ.

Assumption (Consistency). The numerical scheme g_Δ satisfies

$$|x(\Delta; u, x_0) - g_\Delta(x_0, u)| \leq \Delta C(\Delta),$$

for all $x_0 \in X$ and $u \in U$, where $C(\Delta) \to 0$ as $\Delta \to 0$ and $x(\cdot; u, x_0)$ is the solution of $\dot{x} = f(x, u)$ starting from x_0 with a constant control input u. □

For example, for the forward Euler scheme with a fixed step size Δ, the above assumption holds with $C(\Delta) = (e^{K\Delta} - 1)/K$, where K is the uniform Lipschitz constant of $f(\cdot, u)$ on X for all $u \in U$.

Theorem 5.3. *Suppose the consistency assumption holds and that $\mathcal{S}$ is to be controlled with a hybrid controller synthesized using the time-discretized model $\mathcal{S}_d$. If $\mathcal{S}_d \preceq_{(\eta,\gamma_1,\gamma_2,\delta)} \mathcal{T}_{\mathcal{S}_d}$ with $\gamma_1 \geq \eta$, $\gamma_2 \geq \Delta C(\Delta) + \eta$, and $\delta \geq \Delta M/2 + \eta$, then, given any LTL$_{\setminus\bigcirc}$ formula φ, φ being realizable for $\mathcal{T}_{\mathcal{S}_d}$ implies that φ is realizable for $\mathcal{S}$.*

Proof. Let x_0, x_1, x_2, $\cdots$, be the measurements taken for the system (2.1) at times τ_0, τ_1, τ_2, $\cdots$; that is $x_i = x(\tau_i)$ for all $i \geq 0$, where $\tau_0 = 0$ and $\tau_{i+1} = \tau_i + \Delta_i$ for all $i \geq 0$, $\mathbf{u}_i \equiv u_i$ on $[\tau_i, \tau_i + \Delta_i]$ for each $i \geq 0$, and u_i is a control input given by the discrete strategy. We need to show that: (1) every measured state is accounted for in the abstraction (computed from the discretized model), so that the discrete control strategy can be implemented; (2) the plant trajectory $x(t)$, $t \geq 0$, should satisfy the desired specification φ. Let $\{\hat{q}_i\}$ denote a sequence of abstract states corresponding to $\{x_i\}$.

To prove (1): for each i, we need to show that there exists ξ and ξ' such that $|\xi - q_i| \leq \gamma_1$, $|\xi' - q_{i+1}| \leq \gamma_2$, and $\xi' = g_\Delta(\xi, u_i)$. We let $\xi = x_i$ and $\xi' = g_\Delta(x_i, u_i)$. Then $|\xi - q_i| \leq \eta \leq \gamma_1$. Moreover, it follows from the one-step consistency assumption that $|x_{i+1} - g_\Delta(x_i, u_i)| \leq \Delta C(\Delta)$ and $|\xi' - q_{i+1}| \leq |x_{i+1} - g_\Delta(x_i, u_i)| + |x_{i+1} - q_{i+1}| \leq \Delta C(\Delta) + \eta \leq \gamma_2$.

To prove (2): We can prove $x \vDash \varphi$ following the proof of Theorem 4.1. □

6. Example

We consider a simple adaptive cruise control example adapted from [26]. The longitudinal dynamics is given by

$$\begin{aligned} \dot{v} &= u - c_0 - c_1 v^2 \\ \dot{h} &= v_L - v \end{aligned} \tag{6.1}$$

where $v \in [v_{\min}, v_{\max}]$ is the velocity of the controlled car, $u \in [-3a, 2a]$ is the scaled input acceleration, c_i for $i = 1, 2$ are proper constants to account for rolling resistance, air drag and headwind, which are chosen as $c_0 = 0.1$, $c_1 = 0.00016$, $a = 1$, h is the headway (i.e., distance to a lead car) and v_L is the velocity of the lead car, which is taken to be $12m/s$. The set X of states is set to be $X = \{(v, h) \mid 0 \leq v \leq 35, 0 \leq h \leq 300\}$.

We consider a specification of the form

$$\varphi \equiv \Box(h/v \geq 1) \wedge \Diamond\Box((v \leq 35) \ \wedge \ (h/v \geq 1.2)),$$

which enforces a safe time-headway (h/v) of $1s$ and requires eventually reaching and staying within a desired time-headway $h/v \geq 1.2s$ and below a desired speed limit $v \leq 35m/s$. The states of the abstraction are obtained by discretizing the set X with a grid size of 0.5 in v direction and 5 in h direction (i.e., $\eta = (0.5, 5)$); and by discretizing the time with $\tau = 0.5s$. The other parameters $(\gamma_1, \gamma_2, \delta)$ of the abstraction are varied according to the type and level of the uncertainty. These parameters lead to an abstraction with 4332 discrete states (including a state to account for the trajectories leaving the domain X). The computation of an abstraction for this problem with the above mentioned parameters takes about 6 minutes, whereas discrete synthesis takes about 65 seconds on a Macbook Pro with 3GHz Intel Core i7 and 8GB RAM. The computation times are relatively constant with respect to the parameters other than η, which affects the number of discrete states.

To demonstrate the results in Section 4, we assume that the measurements of v and h involve a bounded error in the range $[-\varepsilon, \varepsilon]$ or arrive at the controller with some uncertain delay less than r. We compute all initial conditions in the set X from which it is possible to satisfy the specification for error levels from $\varepsilon = 0$ to $\varepsilon = 0.2$, with increments of 0.04. Similarly, we compute all initial conditions from which it is possible to satisfy the specification for delay levels $r = 0$ to $r = 0.15$, with increments of 0.03. Figure 4 shows some robustness-performance trade-offs. Figure 5 shows a simulation of the system with a controller synthesized using an abstraction that is robust to measurement delays up to $r = 0.12$. This amount of delay corresponds to up to 24% jitter. For a delay level of $r \geq 0.2$ or a measurement error level of $\varepsilon \geq 0.25$, the discrete synthesis problem becomes unrealizable.

As a comparison, we tried to solve the same problem using the approach in [20], where only scalar abstraction parameters were considered. A grid size of 1.5570 in both v and h directions (i.e., $\eta = 0.7785$) leads to an abstraction with 4440 discrete states; roughly the same number of states as before. For this abstraction, even in the nominal case with no uncertainty, the problem was unrealizable. We also tried a grid size of 1 in both v and h directions (i.e., $\eta = 0.5$) leading to an abstraction with 11175 discrete states, however the problem was still unrealizable. These results demonstrate the reduction in conservatism compared to our earlier work in [20] by allowing vector-valued parameters along different continuous state variables.

7. Related work

There are two common ways to construct finite abstractions. One is to partition the state space into a finite number of "proposition-preserving" regions (see, e.g., [27, 40]). This approach has the advantage of resulting in a small number of abstract states (given by elements in the partition) and also do not require any stability assumptions on the system dynamics. However, the fact that the computation of transitions in this type of abstraction relies heavily on the geometry of the vector fields with respect to the partition makes it

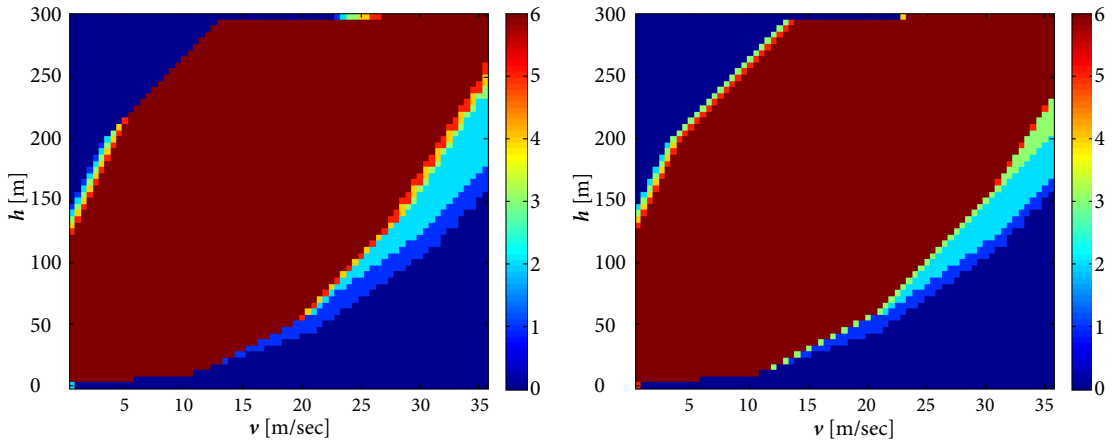

Figure 4: Robustness-performace trade-offs. Abstractions that are robust against sensing delays (left) or measurement errors (right) are computed for six different delay/error levels. The color scale indicates the number of delay/error levels for which it is possible to satisfy the specification from a given initial condition. As can be seen from the plots, as the measurement uncertainty or sensing delay increases, a larger distance h to the lead car is required at initial conditions with high speeds to be able to safely slow down.

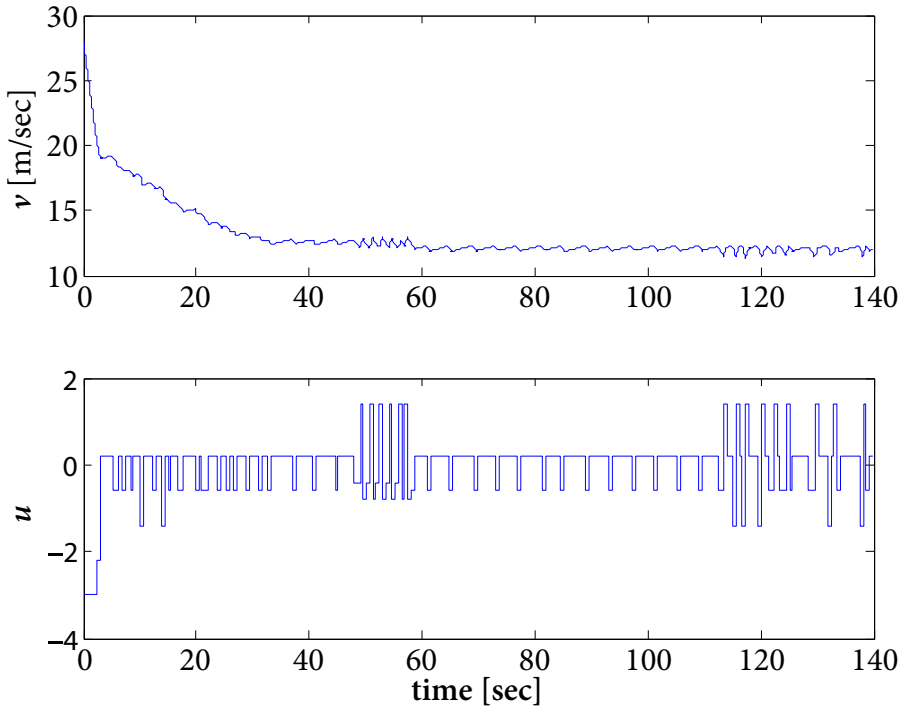

Figure 5: Simulation of the closed-loop system with the synthesized controller, where the initial conditions were $h = 200$ m and $v = 28$ m/sec and the maximum measurement delay was set to be 0.12 s. The simulation was run for 250 control cycles. The upper plot shows the velocity and the lower plot shows the input. Note that due to time-varying delays, the control input is not updated periodically.

difficult to incorporate robustness margins, especially those to deal with imperfect state information except for some special cases [24].

Another approach is to discretize the state space. This has been extensively used for constructing approximate symbolic models for control systems (see, e.g., [25, 29, 30, 33, 43]) based on the notion of approximate (bi)simulation [13]. In these approaches, a finite transition system model is constructed by discretizing the time, the input space, and the continuous state space. Under certain incremental stability assumptions, the resulting finite system can be shown to be approximately bisimilar to the time-discretized model of a continuous-time control system. The stability assumption can be relaxed [43] if one is interested in constructing simulations instead of bisimulations. The advantage of this approach is that it provides a quantitative measure of the fidelity of abstractions using metric transition systems. However in above mentioned papers, the approximation is between the finite abstraction and the time-discretized model of a continuous-time control system and it is unclear how to handle imperfect state information. In this paper we considered a discretization-based approach and addressed these shortcomings. In particular, we introduced abstractions with robustness margins to rigorously reason about the inter-sample behaviors and to account for imperfections in measurements and models.

The type of robustness considered in this paper is related to but distinct from that of [38] and [23]. The focus of [38] and [23] is on the design of discrete controllers for finite transition systems (namely, discrete synthesis) against unmodeled disturbances or transitions, whereas the current paper aims to establish robustness of discrete controllers against imperfect measurements and unmodeled dynamics in the continuous plants. A different quantitative robustness notion for finite transition systems similar to input-output gain has been proposed in [35]. Abstractions that preserve such input-output dynamical stability like robustness notions have recently been introduced in [32, 36]. Our work is complementary to these results as we consider how to preserve satisfaction of temporal logic specifications instead of quantitative input-output gains.

One particular application of our results is the design of delay-robust control protocols from temporal logic specifications (Theorem 4.2). Networked control systems are often considered as a prominent example of control systems where delays are inevitable. The papers [6] and [42] constructed symbolic models to design symbolic controllers for networked control systems. The main difference of our approach lies in that we design robust controllers based on abstractions of a nominal model with additional robustness margins, whereas [6] and [42] constructed symbolic models taking into account specific non-idealities of a networked control system. As a result, the abstractions in [6] and [42] have significantly more discrete states but less conservative, whereas we use more discrete transitions to cope with bounded but possibly unknown delays.

Our work is also related to control of hybrid systems with imperfect state information. The work[19] considered stability of switched systems with limited information under slow switching. Limited information refers to the situation where the state measurements are taken only at sampling times and quantized using a finite alphabet. This is exactly how the hybrid controller is implemented in this paper: it takes measurements at sampling times, maps it to discrete states in the finite abstractions, and looks for appropriate control actions, based on an automaton that represents a discrete control strategy.

8. Conclusions

In this paper we presented a notion of abstractions with robustness margins and showed that, based on these abstractions, it is possible to synthesize provably-correct robust feedback controllers. This allows us to handle various types of imperfections in the models or measurements and to reason about implementation artifacts in a unified fashion. The main insight is to propagate the uncertainty to the discrete level so that the obtained discrete model provides a sound abstraction for a family of uncertain models of a nominal system. Therefore a control strategy for the abstraction can be implemented at the continuous level with correctness guarantees for any member of this family. The idea can be naturally generalized to multi-scale

abstractions, where the abstract states are non-uniformly distributed around the continuous state space. Future work will include investigating such abstractions and combining them with automated refinement procedures to mitigate potential state explosion problem.

References

[1] Alur, R., Henzinger, T. A., Kupferman, O., Vardi, M. Y., 1998. Alternating refinement relations. In: Proc. of CONCUR. pp. 163–178.

[2] Angeli, D., 2002. A Lyapunov approach to incremental stability properties. IEEE Trans. on Automatic Control 47 (3), 410–421.

[3] Angeli, D., Sontag, E. D., 1999. Forward completeness, unboundedness observability, and their lyapunov characterizations. Systems & Control Letters 38 (4), 209–217.

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

[5] Bloem, R., Jobstmann, B., Piterman, N., Pnueli, A., Sa'ar, Y., 2012. Synthesis of reactive (1) designs. J. Comput. System Sci. 78, 911–938.

[6] Borri, A., Pola, G., Di Benedetto, M. D., 2012. A symbolic approach to the design of nonlinear networked control systems. In: Proc. of HSCC. pp. 255–264.

[7] Cervin, A., Lincoln, B., Eker, J., Arzén, K.-E., Buttazzo, G., 2004. The jitter margin and its application in the design of real-time control systems. In: Proceedings of the 10th International Conference on Real-Time and Embedded Computing Systems and Applications. Gothenburg, Sweden, pp. 1–9.

[8] Clarke, E. M., Grumberg, O., Peled, D., 1999. Model Checking. MIT press.

[9] Ehlers, R., 2011. Generalized rabin (1) synthesis with applications to robust system synthesis. In: NASA Formal Methods. Springer, pp. 101–115.

[10] Fainekos, G. E., Girard, A., Kress-Gazit, H., Pappas, G. J., 2009. Temporal logic motion planning for dynamic robots. Automatica 45, 343–352.

[11] Fainekos, G. E., Pappas, G. J., 2009. Robustness of temporal logic specifications for continuous-time signals. Theoretical Computer Science 410 (42), 4262–4291.

[12] Franklin, G. F., Workman, M. L., Powell, D., 1997. Digital Control of Dynamic Systems. Addison-Wesley Longman Publishing Co., Inc.

[13] Girard, A., Pappas, G., 2007. Approximation metrics for discrete and continuous systems. IEEE Trans. on Automatic Control 52, 782–798.

[14] Hale, J. K., 1993. Introduction to Functional Differential Equations. Vol. 99. Springer.

[15] Karaman, S., Frazzoli, E., 2012. Sampling-based algorithms for optimal motion planning with deterministic μ-calculus specifications. In: Proc. of ACC. pp. 735–742.

[16] Kloetzer, M., Belta, C., 2008. A fully automated framework for control of linear systems from temporal logic specifications. IEEE Trans. Automatic Control 53, 287–297.

[17] Kress-Gazit, H., Fainekos, G. E., Pappas, G. J., 2009. Temporal-logic-based reactive mission and motion planning. IEEE Trans. Robotics 25, 1370–1381.

[18] Li, Y., Liu, J., Ozay, N., 2015. Computing finite abstractions with robustness margins via local reachable set over-approximation. In: IFAC Conference on Analysis and Design of Hybrid Systems (ADHS), to appear.

[19] Liberzon, D., 2013. Limited-information control of hybrid systems via reachable set propagation. In: Proc. of HSCC. pp. 11–20.

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

[21] Liu, J., Ozay, N., Topcu, U., Murray, R., 2013. Synthesis of reactive switching protocols from temporal logic specifications. IEEE Trans. on Automatic Control 58 (7), 1771–1785.

[22] Liu, J., Topcu, U., Ozay, N., Murray, R. M., 2012. Reactive controllers for differentially flat systems with temporal logic constraints. In: Proc. of CDC. pp. 7664–7670.

[23] Majumdar, R., Render, E., Tabuada, P., 2011. Robust discrete synthesis against unspecified disturbances. In: Proc. of HSCC. pp. 211–220.

[24] Mickelin, O., Ozay, N., Murray, R. M., 2014. Synthesis of correct-by-construction control protocols for hybrid systems using partial state information. In: Proc. of ACC. pp. 2305–2311.

[25] Mouelhi, S., Girard, A., Gössler, G., 2013. Cosyma: a tool for controller synthesis using multi-scale abstractions. In: Proc. of HSCC. pp. 83–88.

[26] Nilsson, P., Hussien, O., Chen, Y., Balkan, A., Rungger, M., Ames, A., Grizzle, J., Ozay, N., Peng, H., Tabuada, P., 2014. Preliminary results on correct-by-construction control software synthesis for adaptive cruise control. In: Proc. of CDC. pp. 816–823.

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

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

[29] Pola, G., Pepe, P., Di Benedetto, M. D., Tabuada, P., 2010. Symbolic models for nonlinear time-delay systems using approximate bisimulations. Systems & Control Letters 59 (6), 365–373.

[30] Pola, G., Tabuada, P., 2009. Symbolic models for nonlinear control systems: Alternating approximate bisimulations. SIAM J. Control Optim. 48, 719–733.

[31] Reiszig, G., 2011. Computing abstractions of nonlinear systems. IEEE Trans. Automatic Control 56, 2583–2598.

[32] Rungger, M., Tabuada, P., 2014. Abstracting and refining robustness for cyber-physical systems. In: Proc. of HSCC. pp. 223–232.

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

[34] Tabuada, P., Pappas, G. J., 2006. Linear time logic control of discrete-time linear systems. IEEE Trans. Automatic Control 51, 1862–1877.

[35] Tarraf, D., Megretski, A., Dahleh, M. A., 2008. A framework for robust stability of systems over finite alphabets. IEEE Transactions on Automatic Control 53 (5), 1133–1146.

[36] Tarraf, D. C., 2012. A control-oriented notion of finite state approximation. IEEE transactions on automatic control 57 (12), 3197–3202.

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

[38] Topcu, U., Ozay, N., Liu, J., Murray, R. M., 2012. On synthesizing robust discrete controllers under modeling uncertainty. In: Proc. of HSCC. pp. 85–94.

[39] Wolff, E. M., Murray, R. M., 2013. Optimal control of nonlinear systems with temporal logic specifications. In: Proc. of ISRR.

[40] Wongpiromsarn, T., Topcu, U., Murray, R. M., 2012. Receding horizon temporal logic planning. IEEE Trans. Automatic Control 57, 2817–2830.

[41] Wongpiromsarn, T., Topcu, U., Ozay, N., Xu, H., Murray, R. M., 2011. TuLiP: a software toolbox for receding horizon temporal logic planning. In: Proc. of HSCC.

[42] Zamani, M., Mazo Jr, M., Abate, A., 2014. Finite abstractions of networked control systems. In: Proc. of CDC. pp. 95–100.

[43] Zamani, M., Pola, G., Mazo Jr, M., Tabuada, P., 2012. Symbolic models for nonlinear control systems without stability assumptions. IEEE Trans. Automatic Control 57, 1804–1809.