Accurate Abstractions for Controller Synthesis with Non-uniform Disturbances

Yunjun Bai[1] and Kaushik Mallik[2]

[1] SKLCS, Institute of Software Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China, <baiyj@ios.ac.cn>
[2] MPI-SWS, Kaiserslautern, Germany, <kmallik@mpi-sws.org>.

Abstract. Abstraction-Based Controller Synthesis (ABCS) is an emerging field for automatic synthesis of correct-by-design controllers for non-linear dynamical systems in the presence of bounded disturbances. Existing ABCS techniques assume a global (state-independent) and uniform upper bound on disturbances. This can be overly pessimistic, resulting in a failure to find a controller. In this paper, we extend ABCS to accurately compute abstractions for system models with state and input-dependent non-uniform disturbances. This requires a subtle assume-guarantee style computation of the state evolution and the estimation of the error. We empirically show the benefit of our approach with significantly smaller amount of non-determinism in the abstract transitions.

1 Introduction

Abstraction-based controller synthesis (ABCS) is a fully automated controller synthesis technique for non-linear, continuous dynamical systems with respect to temporal control objectives [31, 4, 12, 27, 24, 17]. In ABCS, first, the original non-linear dynamical system, under a sampled-time semantics, is approximated by a simpler finite-state system, called the *abstraction*. Second, using automata-theoretic algorithms for reactive synthesis [21], an appropriate two-player game is solved on the abstraction to compute an *abstract controller*. Finally, the abstract controller is *refined* to a sampled-time controller for the original continuous system. The correctness of the refinement process, and thus correctness of ABCS, depends on an alternating refinement or feedback refinement relation [2, 19, 24, 27] between the original continuous system and the abstraction.

A simple and effective abstraction procedure is to partition the state and the input spaces using disjoint hypercubes, which form the state and input spaces of the finite-state abstraction. For every abstract state and input pair, the (non-deterministic) transitions of the abstraction provides an over-approximation of the possible states that can be reached under the effect of the continuous dynamics in the presence of disturbances. This algorithm has been implemented in several tools [22, 29, 17].

Existing abstraction techniques work on systems whose dynamics can be modeled as the differential inclusion

$$\dot{x} \in f(x, u) + [\![-d, d]\!],$$

where d represents a constant upper-bound on the disturbances which the system may experience. One important shortcoming of such techniques is the use of a global, state-independent, uniform upper bound on the disturbances. In real-world applications, such a global worst-case view often leads to an unnecessarily pessimistic abstraction, which could result in failure of controller synthesis. Consider for example a simple path planning problem for a robot in a space filled with obstacles (Fig. 1). Assume that the disturbance models how slippery the floor is. If it is observed that the floor is very slippery in some part (e.g. the cross-hatched region in Fig. 1) of the state space, even if this part is less critical and obstacle-free, the present state-of-the-art would assume that the floor is very slippery *everywhere*. This pessimism might cause a significant reduction in the size of the controller domain, especially in the critical obstacle-filled parts of the state space.

In this paper, we extend ABCS to system models with *state and input-dependent disturbance*:

$$\dot{x} \in f(x, u) + [\![-d(x, u), d(x, u)]\!].$$

The advantage of this generalization is two fold: First, it increases the accuracy of ABCS and the chances of a successful controller synthesis, even when the existing approaches fail. For example, in Fig. 1 the small purple region is the domain of the controller obtained using one of the existing approaches, whereas the large green region is the domain of the controller obtained using our proposed abstraction method. Second, our generalization enables localized incremental updates of the existing abstraction when the underlying disturbance model either changes locally over time, or is updated during runtime through learning-based exploration of the state space. The problem of exploration has recently gained popularity in the context of machine-learning and robotics applications, as it turns out that in most cases only approximate models of the system and the environment are known apriori, and a more precise model is learnt during runtime [10, 23, 5, 6]. Existing ABCS techniques are not suitable for dynamic changes in the underlying model. Thus, any change in the model involves a complete re-computation of the whole abstraction and the associated controller. We show in a companion paper [3] how the method presented in this paper is instrumental for locally adapting an existing abstraction under changing disturbances. The adaptation algorithm propagates the effect of changes backwards, and selectively re-computes transitions only for those abstract states which might have been affected by this change. We do not discuss this application of adaptive re-computation of abstractions any further in this paper.

Technically, for construction of abstraction, the extension to state- and input-dependent disturbances is non-trivial for the following reason. The key step in the abstraction process is the estimation of a *growth*

bound which is an upper-bound on the maximum perturbation of a nominal trajectory under the influence of disturbance. When the disturbance does not depend on the system's state, the growth bound can be computed independently of the system's state trajectories by solving the initial value problem for an auxiliary linear ordinary differential equation [27, 19]. However, for state-dependent disturbances, the dynamics of the growth bound is non-linear, and is coupled with the state trajectories of the system. We show a new algorithm that jointly computes state trajectories and growth bounds. The joint evolution means that there is a cyclic dependency between the size of the initial hyper-rectangle from which the trajectory starts and the size of the growth bound. A simple numerical integration procedure can be unsound unless the disturbance on the dynamics between two integration points is estimated soundly. We show how the cycle can be broken by locally approximating the disturbance by an upper bound in an assume-guarantee fashion and provide an algorithm to approximate the growth bound to any desired accuracy.

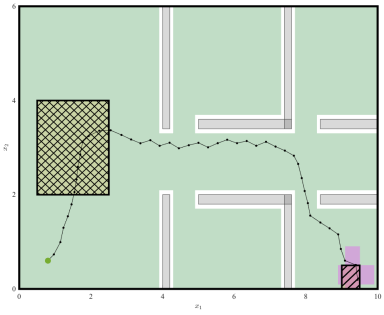

Fig. 1. State space of a vehicle assigned with a reach-avoid task. The goal and the obstacles are indicated using the hatched region in the bottom right and the gray rectangles, respectively. The region in cross-hatch has higher disturbance ($d = 0.1$) than the rest ($d = 0.05$). The purple and the green region represent the controller domains due to a uniform worst case disturbance (existing approaches) and state-dependent disturbances (our algorithm), respectively.

We have implemented our algorithm on top of SCOTS v0.2 [29]. We empirically show, through a number of case studies, that a state-dependent model of disturbances can lead to more precise abstractions. In fact, in our examples, the global upper bound was too conservative to find a controller.

2 Preliminaries

2.1 Notation

We use the notation $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{N}$ to denote the set of reals, the set of positive reals, and the set of natural numbers (excluding 0) respectively. Given a set A and an $n \in \mathbb{N}$, we use A^n to denote the cartesian product $\bigtimes_n A$. Given any vector $x \in A^n$, we use x_i to denote the element in the i-th dimension of x. For any function $f\colon A \to B$, we write dom f to denote the domain of f.

2.2 Systems and Refinement

A *system* $S = (X, U, F)$ consists of a state space X, an input space U, and a transition relation $F \subseteq X \times U \times X$. A system is called *finite* if X and U are finite sets. A *run* of a system is a finite or infinite sequence $\xi = x_0 x_1 \ldots$ such that for every i-th element in the sequence, with $i \geq 0$, there is an $u_i \in U$ such that $(x_i, u_i, x_{i+1}) \in F$. The set of runs of S is denoted $\mathcal{B}(S)$ and the set of runs of S starting from a state $x \in X$ is denoted $\mathcal{B}(S)(x)$. A *controller* $C : X \to U$ is a function mapping states to inputs which restricts the set of behaviors of a system: a run $\xi = x_0 x_1 \ldots$ is *compatible* with C if for every i, we have $(x_i, C(x_i), x_{i+1}) \in F$.

For ABCS to work, we need a suitable notion of abstraction. *Feedback Refinement Relations* (FRR) provide such a notion [2, 27]. Here we omit the details of how an abstraction computed using FRR can be used in ABCS, which can be found in the original paper on FRR-based symbolic control [27].

Definition 1 (Feedback Refinement Relations). *Let $S_i = (X_i, U, F_i)$ for $i \in \{1, 2\}$ be two systems on the same set of inputs. Let $\pi_i : X_i \to 2^U$ be the mapping $\pi_i(x) = \{u \in U \mid \exists x' \in X_i \cdot (x, u, x') \in F_i\}$ giving the set of allowed inputs in state x of S_i. A* feedback refinement relation *from S_1 to S_2 is a relation $Q \subset X_1 \times X_2$ such that for all $x_1 \in X_1$ there is an $x_2 \in X_2$ such that $(x_1, x_2) \in Q$, and for all $(x_1, x_2) \in Q$, the following holds:*

(i) $\pi_2(x_2) \subseteq \pi_1(x_1)$ *and*
(ii) for all $u \in \pi_2(x_2)$ and for all $x_1' \in X_1$ such that $(x_1, u, x_1') \in F_1$, there is an $x_2' \in X_2$ such that $(x_2, u, x_2') \in F_2$ and $(x_1', x_2') \in Q$.

We use $S_1 \preccurlyeq_Q S_2$ to represent that Q from S_1 to S_2 is an FRR. We say S_2 is an *abstraction* of S_1 if there is an FRR Q such that $S_1 \preccurlyeq_Q S_2$. It is well known that if $S_1 \preccurlyeq_Q S_2$ then any controller $C : X_2 \to U$ solving a (suitably projected) control problem on S_2 can be refined to a controller solving the control problem on S_1 [2]. In this paper we focus on computing an abstraction; we refer to the standard reactive synthesis literature [26, 21] for the details of the synthesis algorithms.

2.3 Time-Sampled Control Systems

We consider continuous-time *control systems* $\Sigma = (X, U, d, f)$, where $X = \mathbb{R}^n$ is the state space, $U \subset \mathbb{R}^m$ is the input space, $d : X \times U \to X$ is a function having continuous first order derivative in the first argument which models the upper-bound of the state dependent disturbances, and $f : X \times U \to X$ is the unperturbed system dynamics. The overall dynamics is expressed by the following differential inclusion:

$$\dot{x} \in f(x, u) + [\![-d(x, u), d(x, u)]\!]. \tag{1}$$

Since the set of disturbances $[\![-d(x, u), d(x, u)]\!]$ at any given state $x \in X$ and input $u \in U$ is symmetric about the origin, hence without loss of generality we

assume that $d(x,u)$ is positive for all $x \in X$ and input $u \in U$. We assume that f is explicitly known and is continuously differentiable in x for all $u \in U$. The set $[\![-d(x,u), d(x,u)]\!]$ represents the set of all possible disturbances capturing the unmodeled dynamics and environmental uncertainties. Given an initial state $x_0 \in X$, a constant control input[3] $u \in U$ for time $\tau \in \mathbb{R}_+$ and any interval $I \subset [0,\tau]$, the evolution of Σ in I is given by the continuous trajectory $\xi_u : I \to X$ which satisfies $\dot{\xi}_u(t) \in f(\xi_u(t),u) + [\![-d(\xi_u(t),u), d(\xi_u(t),u)]\!]$. Moreover we define the *nominal trajectory* $\varphi_{x_0 u} : I \to X$, which satisfies the unperturbed differential equation $\dot{\varphi}_{x_0 u}(t) = f(\varphi_{x_0 u}(t), u)$ and $\varphi_{x_0 u}(0) = x_0$.

Given a fixed time sampling parameter $\tau > 0$, we define a system $\Sigma_\tau = (X, U, f_\tau)$ that represents the *sampled-time representation* of a control system Σ, where $f_\tau \subseteq X \times U \times X$ is the transition relation such that for all $(x,u,x') \in X \times U \times X$, we have $(x,u,x') \in f_\tau$ if and only if there is a solution of (1) for the constant input u such that $\xi(0) = x$ and $\xi(\tau) = x'$. The state space of Σ_τ is still infinite; our goal is to obtain a *finite-state* abstraction $\widehat{\Sigma}$ of Σ_τ.

3 Computation of Abstraction

3.1 A Generic Construction

Fix a control system $\Sigma = (X, U, d, f)$ and its sampled-time representation $\Sigma_\tau = (X, U, f_\tau)$ for a fixed sampling time $\tau > 0$. Our goal is to compute a finite-state abstraction $\widehat{\Sigma} = (\widehat{X}, \widehat{U}, \widehat{f})$. We work in the usual setting of ABCS, where an abstraction is obtained by first partitioning the state space and input space of Σ_τ into finitely many hypercubes, and then over-approximating the transitions to obtain the desired finite-state abstraction $\widehat{\Sigma}$.

We consider abstract state spaces defined by hyper-rectangular covers. A *hyper-rectangle* $[\![a,b]\!]$ with $a, b \in (\mathbb{R} \cup \{\pm\infty\})^n$ defines the set $\{x \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i \text{ for } i \in \{1,\ldots,n\}\}$; it is non-empty if $a < b$. For $\eta \in \mathbb{R}^n_{>0}$, we say that a hyper-rectangle $[\![a,b]\!]$ has diameter $\eta \in (\mathbb{R}_{>0} \cup \{\pm\infty\})^n$ if for each $i \in \{1,\ldots,n\}$, we have $|b_i - a_i| = \eta_i$. The *center* of a non-empty hyper-rectangle $[\![a,b]\!]$, written $ctr([\![a,b]\!])$, is the point $(\frac{1}{2}(b_1 - a_1), \frac{1}{2}(b_2 - a_2), \ldots, \frac{1}{2}(b_n - a_n))$.

A set C of hyper-rectangles is called a *cover* of the state space X if every $x \in X$ belongs to some hyper-rectangle in C.

In the following, we make the standard assumption that there is a compact subset $X' \subseteq X$ of the state space that is of interest to the control problem. Given a discretization parameter $\eta \in \mathbb{R}^n_{>0}$, the abstract state space $\widehat{X}$ is a (finite) cover of the compact set X' with hyper-rectangles of diameter η together with a finite number of hyper-rectangles (some are unbounded) which cover $X \setminus X'$.

The abstract transition relation $\widehat{f}$ is defined as an over-approximation of the reachable states in a single time sampling period τ. The over-approximation is formalized by a *growth bound*.

[3] We restrict our notation to piecewise constant control inputs as more general control inputs will be unnecessary for the later part.

Definition 2 (Growth bound). *Let $\Sigma = (X, U, d, f)$ be a control system. Given a fixed sampling time $\tau > 0$ and sets $K \subset X$ and $U' \subset U$, a* growth bound *is a map $\beta_\tau : \mathbb{R}^n \times \mathbb{R}^n_{>0} \times U' \to \mathbb{R}^n_{>0}$ satisfying the following conditions:*

(i) $\beta_\tau(p, r, u) \geq \beta_\tau(p, r', u)$ whenever $r \geq r'$ and $u \in U'$;
(ii) for all $p \in K$ and $u \in U'$, $[0, \tau] \subset dom\, \varphi_{pu}$, and if ξ_u is a trajectory of (1) on $[0, \tau]$ with $\xi_u(0) \in K$ then

$$| \xi_u(\tau) - \varphi_{pu}(\tau) | \leq \beta_\tau(p, | \xi_u(0) - p |, u) \tag{2}$$

holds component-wise. Recall that $\varphi_{pu}(t)$ denotes the nominal trajectory starting from state p.

Def. 2 is an adaptation of the growth bound introduced by Resissig et al. [27]; the difference is that our growth bound also depends on the initial state of the nominal trajectory. The intuition behind this extra requirement is that the deviation of the actual trajectory from the nominal trajectory may not be independent of the path of the state trajectory in our case. This will be clear in the subsequent development.

Given the notion of state space partition and growth bound, we can now define the finite-state abstraction as follows:

Definition 3. *Let $\Sigma = (X, U, d, f)$ be a control system, let $\tau > 0$ be a sampling time, $\eta \in \mathbb{R}^n_{>0}$ a discretization parameter, and $\beta_\tau : \mathbb{R}^n \times \mathbb{R}^n_{>0} \times \widehat{U} \to \mathbb{R}^n_{>0}$ a growth bound. A finite-state abstraction $\widehat{\Sigma} = (\widehat{X}, \widehat{U}, \widehat{f})$ of Σ_τ consists of a finite set $\widehat{X}$ which forms a cover of X with hyper-rectangles of diameter η, a finite set $\widehat{U} \subseteq U$ of inputs, and a transition relation $\widehat{f} \subseteq \widehat{X} \times \widehat{U} \times \widehat{X}$ which satisfies, for all $\widehat{x}, \widehat{x}' \in \widehat{X}$ and $\widehat{u} \in \widehat{U}$, $(\widehat{x}, \widehat{u}, \widehat{x}') \in \widehat{f}$ if and only if*

$$\left(\varphi_{ctr(\widehat{x})\widehat{u}}(\tau) + [\![-\gamma, \gamma]\!]\right) \cap \widehat{x}' \neq \emptyset, \tag{3}$$

where $\gamma = \beta_\tau(ctr(\widehat{x}), \frac{1}{2}\eta, \widehat{u})$.

The following theorem is adapted from [27, Thm. VIII.4].

Theorem 1. *For every control system $\Sigma = (X, U, d, F)$, for every sampling time τ, state discretization η, and growth bound β, we have $\Sigma_\tau \preccurlyeq_Q \widehat{\Sigma}$, through the Feedback Refinement Relation $(x, \widehat{x}) \in Q$ if and only if $x \in \widehat{x}$.*

Thus, given parameters τ and η, in order to compute an abstraction, we have to show how to compute a suitable growth bound.

3.2 Growth Bound Computation

We now present a specific way of computing the growth bound $\beta_\tau(\cdot, \cdot, \cdot)$, which is motivated by the growth bound used in [27] for the case of uniform disturbance.

Given that f is continuously differentiable, we can define a certain upper bound on the speed of deviation of two perturbed trajectories of the system Σ, which is formalized in the following definition. We use $D_j f_i$ to denote the partial derivative with respect to the j-th component of the first argument of f_i.

Definition 4. *For a given control system $\Sigma = (X, U, d, f)$, let $K \subseteq K' \subseteq X$ be sets, where K' is convex, such that for all $u \in U'$ and for all $t \in [0, \tau]$, $\xi_u(0) \in K$ implies $\xi_u(t) \in K'$. Define the parametrized matrix $L : U' \to \mathbb{R}^{n \times n}$ which satisfies*

$$L_{i,j}(u) \geq \begin{cases} \sup_{x \in K'} D_j f_i(x, u) & \text{if } i = j, \\ \sup_{x \in K'} |D_j f_i(x, u)| & \text{otherwise.} \end{cases} \tag{4}$$

In the following, we use the maximum disturbance functions $w(t; p, u, z(t)) := \sup\{d(x', u) \mid x' \in [\varphi_{pu}(t) - z(t), \varphi_{pu}(t) + z(t)]\}$ and $w_{\max}(u) = \sup\{d(x, u) \mid x \in X\}$. Note that we use semicolon in the function argument to separate variables (symbols preceding the semicolon) from parameters (symbols following the semicolon). The following theorem establishes a way of sound computation of the growth bound.

Theorem 2. *Fix a time-sampling parameter $\tau > 0$. Let $\Sigma = (X, U, d, f)$ be a control system, and $z : \mathbb{R}_{>0} \to X$ be a solution to the initial value problem:*

$$\dot{z}(t) = L(u)z(t) + \overline{w}(t; u), \quad z(0) = r \tag{5}$$

where for all $t \in [0, \tau]$, $\overline{w}(t; u) \geq w(t; p, u, z(t))$. Then any trajectory $\xi_u(\cdot)$ of Σ with $|\xi_u(0) - p| \leq r$ is continuable to $[0, \tau]$, and $|\xi_u(t) - \varphi_{pu}(t)| \leq z(t)$ holds for all $t \in [0, \tau]$. In particular, $\beta_\tau(p, r, u) = z(\tau)$ is a growth bound of (1) on K, U' associated with τ.

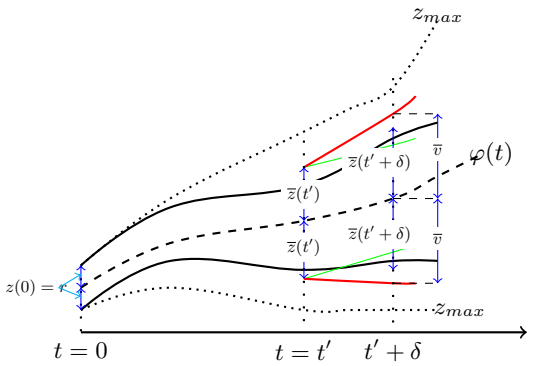

Fig. 2. Growth bound computation. The dashed line shows the nominal trajectory and the solid lines show the ideal growth bound z^*. The dotted lines show the upper bound $z_{\max}$. Having computed $\bar{z}$ at time t', we want to compute $\bar{z}$ at time $t' + \delta$. The green lines show that it is unsound to use $\bar{z}(t')$ to compute a bound on the disturbance in the interval $[t', t' + \delta)$. Instead, we compute $\bar{v}$ using the estimate $\bar{z}(t')$ and the upper bound $w_{\max}$ on the error (the red lines). Using $\bar{v}(t' + \delta)$, we estimate $\bar{z}(t' + \delta)$ soundly.

The proof of Thm. 2 relies on a lemma adopted from [27].

Lemma 1. *Let for any given $\tau > 0$ and $A \subseteq \mathbb{R}^n$, $\xi_i : [0, \tau] \to A$ with $i \in \{1, 2\}$ be two perturbed solutions of a control system with continuous right hand side $f : \mathbb{R}^n \to \mathbb{R}^n$ s.t. the following holds for almost all $t \in [0, \tau]$:*

$$|\dot{\xi}_i(t) - f(\xi_i(t))| \leq w_i(t) \tag{6}$$

where $w_i : [0, \tau] \to \mathbb{R}^n_{>0}$, $i \in \{1, 2\}$ are integrable. Let $L \in \mathbb{R}^{n \times n}$ be a matrix with non-negative off-diagonal entries, and let for all $x, y \in A$ we have

$$x_i \geq y_i \Rightarrow f_i(x) - f_i(y) \leq \sum_{j=1}^{n} L_{i,j}|x_j - y_j|. \tag{7}$$

Let $\rho : [0, \tau] \to \mathbb{R}^n_{>0}$ be absolutely continuous and satisfying

$$\dot{\rho}(t) = L\rho(t) + w_1(t) + w_2(t) \tag{8}$$

for almost all $t \in [0, \tau]$. Then $|\xi_1(0) - \xi_2(0)| \leq \rho(0)$ implies $|\xi_1(t) - \xi_2(t)| \leq \rho(t)$ for every $t \in [0, \tau]$.

Proof (Proof of Thm. 2).

Condition (i) in Def. 2 follows from the same argument in the proof of [27, Thm. VIII.5]. In order to prove Condition (ii), we will apply Lem. 1 to an unperturbed trajectory $\xi_1 = \varphi_{pu}(\cdot)$ (with $w_1(t)$ uniformly set to 0) and a perturbed trajectory $\xi_2 = \xi_u(\cdot)$ (with $w_2(\cdot) = w(\cdot)$) of (1). To be able to do that, we first prove (6) and (7). (6) holds by definition of $\overline{w}(\cdot; p, u, z(\cdot))$ in Thm. 2: At any time instant t, $w(t; p, u, z(t))$ gives the supremum of the disturbances experienced by all the perturbed trajectories $\xi_u(t)$ which started from p. (7) follows by the same argument as in the proof of [27, Thm. VIII.5].

Now note that since both $d(\cdot, u)$ and $\xi(\cdot)$ are continuous functions, hence $d(\xi(\cdot), u)$ is also a continuous function. Then the time varying map $W : t \mapsto [\![-d(\xi(t), u), d(\xi(t), u)]\!]$ can be shown to be upper semi continuous w.r.t. inclusion (in t). So by Filippov's Lemma [9], there exists an integrable map $m : [0, \tau] \to [\![-d_{\max}, d_{\max}]\!]$ such that $\dot{\xi}(t) = f(\xi(t), u) + m(t)$ for any $t \in [0, \tau]$. The rest of the first part of the proof amounts to applying Lem. 1 to derive the inequality (2), and showing that any trajectory $\xi_u : [0, \tau'] \to K'$ is continuable to $[0, \tau]$. These follow the same arguments as in the proof of [27, Thm. VIII.5], and is omitted.

The second part of Thm. 2 follows by observing that for any two $\overline{w}_1(\cdot)$ and $\overline{w}_2(\cdot)$ with $z_1(\cdot)$ and $z_2(\cdot)$ being the associated solutions of (5), if for all $t \in [0, \tau]$, $\overline{w}_1(t) \geq \overline{w}_2(t) \geq w(t; p, u, z(t))$ holds, then since the r.h.s. of (5) and $z(0)$ are positive, hence for all $t \in [0, \tau]$, $z_1(t) \geq z_2(t)$ holds. Hence $z^*(\cdot)$ must be the smallest possible solution of the ODE (5). □

In the following, we write $z_{\max}(\cdot; r, u)$ for the solution of the IVP (5) when $\overline{w}(\cdot; u) = w_{\max}(u)$ and write $z^*(\cdot; p, r, u)$ for the solution when $\overline{w}(\cdot; u) = w(\cdot; p, u, z(\cdot))$. Note that in either case, existence and uniqueness of the solution is guaranteed by the continuity of both the r.h.s. and the first order partial derivative of the r.h.s. w.r.t. $z(t)$; the non-trivial latter case follows from the fact that the function $d(\cdot, u)$ has continuous first order derivative.

Remark 1. For all feasible $\overline{w}(t;u)$, if $z(t)$ is associated with the disturbance $\overline{w}(t;u)$, then $z^*(t;\cdot,\cdot,\cdot) \leq z(t) \leq z_{\max}(t;\cdot,\cdot)$ holds for all t. That is, $z^*(\tau;\cdot,\cdot,\cdot)$ is the tightest growth bound for a fixed $L(u)$ and $z_{\max}(\tau;\cdot,\cdot,\cdot)$ is an upper bound. The growth bound used in [27] is basically $z_{\max}(\tau;\cdot,\cdot)$.

Unfortunately, when $\overline{w}(t;u) = w(t;p,u,z(t))$, (5) is a highly non-linear ODE, and it is difficult to obtain a closed form solution for $z(t)$, if it exists at all. On the other hand, numerical computation of $z(t)$ will be unsound due to the unavoidable time-discretization: At any given time-step k, w will be estimated based on the disturbances seen upto the previous time-step $k-1$, which could be smaller than the actual disturbances seen in the interval $((k-1)h, kh]$, where h is the step size of the numerical solver (see Fig. 2).

For these reasons, we only give an algorithm that approximates the value of $z^*(t;p,r,u)$ by $\overline{z}(t;p,r,u,\delta)$ for a fixed parameter $\delta \in (0,\tau]$, such that the approximation $\overline{z}(\tau;p,r,u,\delta)$ is a sound growth bound of (1) and the approximation error can be made arbitrarily small. (Soundness of $\overline{z}(\tau;p,r,u,\delta)$ as a growth bound essentially prove that the numerical solution that we present later is also sound.) We introduce a conservative over-approximation of $w(t;p,u,z(t))$ that does not depend on $z(t)$, and the conservatism can be made arbitrarily small by tuning δ. This is based on an an assume-guarantee style decomposition of the computation of $w(t;p,u,z(t))$. For any $t \geq 0$, define $t^- := \max\{0, t-\delta\}$. *Assume* that for any given $t \geq 0$, the value of $\overline{z}(t^-;p,r,u,\delta)$ is already known, and in particular, $\overline{z}(0;p,r,u,\delta) = r$. Then we can *guarantee* that the over-approximation of the maximum disturbance seen at time t is $\overline{v}(t;p,r,u,\delta) := \sup\{d(x') \mid \varphi_{pu}(t) - \varepsilon \leq x' \leq \varphi_{pu}(t) + \varepsilon\}$, where $\varepsilon = z_{\max}(\delta^-; \overline{z}(t^-;p,r,u,\delta), u)$ and $\delta^- = \min\{t,\delta\}$. Fig. 2 shows the calculations.

We write $\overline{z}(\cdot;p,r,u,\delta)$ for the solution of the IVP (5) when $\overline{w}(\cdot;u)$ is set to $\overline{v}(\cdot;p,r,u,\delta)$. Note that existence and uniqueness of the solution is guaranteed, because the right hand side is both continuous and has continuous first order partial derivative with respect to $z(t)$: the proof is trivial when we note that $\overline{v}(t;\cdot,\cdot,\cdot,\delta)$ is independent of $\overline{z}(t;\cdot,\cdot,\cdot,\delta)$. For every $\delta > 0$, the solution $\overline{z}(\tau;p,r,u,\delta)$ is a growth bound, and the smaller the value of δ, the tighter is the growth bound, and hence the less conservative (i.e. less non-determinism) is the abstract transition system. This is formalized using the following theorem.

Theorem 3. *Using the notation as in Thm. 2, for all $\delta > 0$ and for all $t \in [0,\tau]$, $\overline{v}(t;p,r,u,\delta) \geq w(t;p,u,z^*(t))$. Moreover, for all $t \in [0,\tau]$ and for all $\varepsilon > 0$, there exists a $\delta > 0$, s.t. $|\overline{z}(t;p,r,u,\delta) - z^*(t;p,r,u)| < \varepsilon$.*

It is easy to see from Thm. 3 that by fixing $t = \tau$, we can always find a small enough $\delta > 0$ to get an arbitrarily tight and sound growth bound $\beta_\tau(p,r,u) = \overline{z}(\tau;p,r,u,\delta)$. Moreover, it can be shown that if $L(u)$ is a non-negative matrix, then a δ corresponding to $t = \tau$ and $\varepsilon > 0$ ensures that $|\overline{z}(t;p,r,u,\delta) - z^*(t;p,r,u)| < \varepsilon$ for all $t \in [0,\tau]$.

3.3 ComputeAbs: Abstraction Algorithm

The abstraction algorithm ComputeAbs iterates over all the abstract state-input pairs $(\widehat{x}, \widehat{u}) \in \widehat{X} \times \widehat{U}$. For each pair, it computes the set of transitions $(\widehat{x}, \widehat{u}, \widehat{x}')$ using an ODE solver in the following way.

First, for all $(\widehat{x}, \widehat{u})$, the disturbance function d is over-approximated using a piecewise constant function $\widehat{d} : \widehat{X} \times \widehat{U} \to \mathbb{R}^n$ defined as

$$\widehat{d}(\widehat{x}, \widehat{u}) := \sup_{x' \in \widehat{x}} \{d(x', \widehat{u})\}. \tag{9}$$

Second, a numerical ODE solver is used to jointly solve the IVP for the dynamics and the growth bound:

$$\dot{x}(t) = f(x(t), u), \quad x(0) = ctr(\widehat{x}) \tag{10}$$

$$\dot{z}(t) = L(u)z(t) + \widehat{w}(t; u), \quad z(0) = \frac{1}{2}\eta \tag{11}$$

where $\widehat{w}(t; u)$ approximates $\overline{w}(t; u)$ as described below.

The numerical algorithm used in our implementation is based on 4th order Runge-Kutta method. The ODE solver uses a fixed step size $h > 0$. For soundness, we need to make sure that the distance between two consecutive intermediate points obtained during the numerical solution of the IVP are not separated by a distance more than the discretization η. The following condition ensures this property:

$$\eta \geq \sup_{x,u} |f(x, u)| \times h. \tag{12}$$

Using the notation used in Thm. 3 we set $\delta = h$. This sets $\delta^- = \delta = h$, $t^- = t - \delta = t - h$ (note that each time point t equals kh for some $k \in \mathbb{N}$) and $\varepsilon = z_{\max}(\delta^-; z^*(t - h; p, r, u), u)$. Finally, we define

$$\widehat{w}(t; u) = \sup_{\widehat{x}'} \{\widehat{d}(\widehat{x}', u) \mid \widehat{x}' \in \widehat{X} \text{ s.t. } \widehat{x}' \cap (x(t) + [\![-r(t) - \varepsilon, r(t) + \varepsilon]\!]) \neq \emptyset\}$$

Remark 2. Notice that the solution $z_{\max}(t; r, u)$ can be written explicitly as $e^{L(u)t}r + \int_0^t e^{L(u)(t-s)} w_{\max}(u) ds$. Since $t = \delta^-$ is fixed, for each $\widehat{u}$, we can precompute the matrix exponential $e^{L(\widehat{u})\delta^-}$ and the above integral for $t = \delta^-$, and compute ε as an affine transformation of $z^*(t - h; p, r, u)$.

Based on the solution $(x(\tau), z(\tau))$ returned by the ODE solver, the abstraction algorithm adds transitions $(\widehat{x}, \widehat{u}, \widehat{x}')$ for every $\widehat{x}, \widehat{x}' \in \widehat{X}$ and every $\widehat{u} \in \widehat{U}$ such that $\widehat{x}' \cap [x(\tau) + -z(\tau), x(\tau) + z(\tau)] \neq \emptyset$.

Related Work on Reach-tube Computation Our growth bound can be seen as a special case of over-approximating the set of reachable states of perturbed nonlinear systems over a bounded time horizon. We compare our setting with related work in reach-tube construction and point out some specifics of our setting. Reach-set (set of reachable states after a fixed time horizon), and reach-tube

(set of reachable states in an interval of time) computation is a very well explored area in the hybrid systems research. The problem of exact computation of reach-set or reach-tube is undecidable for general nonlinear systems [16]. Despite that, many researchers have presented various ways to approximate reach-set and reach-tube computation in the past. We point out few features of our growth bound computation in contrast to the reach-set computation in the existing literature. First, the difficulties: our dynamics is non-linear and in continuous time (unlike, e.g., [11, 28]), we do not assume any stability conditions (unlike [13]) and our disturbances are state- and input-dependent and depend on the inter-sampling behavior of the underlying trajectories (unlike [19, 27]). On the oher hand, in our setting, the reach-set is guaranteed to be convex hyper-rectangles, so that checking intersection with abstract states is efficient (unlike [30, 1, 14]) and the numerical algorithm always terminates (unlike [30]).

4 Experimental results

We implemented ComputeAbs as an extension of SCOTSv0.2 [29], and evaluated the algorithms on 4 benchmark examples to show the effectiveness of our method. All the experiments of this section were performed on an Intel(R) Xeon(R) Processor E7-8857 v2 (3.00GHz) with 16 GB memory.

Since we focus on the abstraction step of ABCS in this paper, no data related to synthesis is presented. However, we emphasize that the number of abstract transitions is a good indicator for the chances of successful synthesis. More transitions indicates a less accurate abstraction making the synthesis harder.

Case Study		SCOTSv0.2	ComputeAbs	relative
Vehicle (3d)	# transitions	3.123×10^7	2.921×10^7	55%
	time (sec.)	16.81	24.64	0.7×
Manipulator(4d)	# transitions	1.244×10^9	1.048×10^9	84%
	time (sec.)	735.65	1012.34	0.7×
Cancer treatment(3d)	# transitions	8.301×10^8	7.263×10^8	88%
	time (sec.)	28.28	78.56	0.4×
Aircraft landing (3d)	# transitions	5.530×10^9	2.047×10^9	37%
	time (sec.)	264.099	528.62	0.5×

Table 1. Experimental results for four different system models : Comparison of SCOTSv0.2 and ComputeAbs in terms of number of transitions (top) and computation time (bottom).

In the following, we provide more information about the model used in each of the experiments. A performance comparison between ComputeAbs and the original SCOTSv0.2 for all the experiments is summarized in Table 1.

4.1 Vehicle Dynamics

The model. We consider the three-dimensional vehicle dynamics

$$\begin{aligned}\dot{x}_1 &= \frac{u_1 \cos(\alpha + x_3)}{\cos \alpha} + [\![-d_1(x_1, x_2), d_1(x_1, x_2)]\!] \\ \dot{x}_2 &= \frac{u_1 \sin(\alpha + x_3)}{\cos \alpha} + [\![-d_2(x_1, x_2), d_2(x_1, x_2)]\!] \\ \dot{x}_3 &= u_1 \tan u_2,\end{aligned}$$

adapted from [27], where the states x_1, x_2 and x_3 respectively represent the two-dimensional coordinates and the orientation of the vehicle as function of time. The control inputs u_1 and u_2 represent the rear wheel velocity and the steering angle, respectively, and α is a parameter given by $\alpha = \tan^{-1}(\tan(u_2)/2)$. The bounds for the state and input spaces are given by $[0, 10] \times [0, 6] \times [-3.5, 3.5]$ and $[-1, 1] \times [-1, 1]$, respectively. The disturbance functions $d_i(\cdot, \cdot)$ for $i \in \{1, 2\}$ are given by

$$d_i(x_1, x_2) = \begin{cases} 0.1 & (x_1, x_2) \in (0.45, 2.55) \times (1.95, 4.05), \\ 0.05 & \text{otherwise}, \end{cases}$$

where we assume that the function d_i are interpolated using a rapidly changing smooth function (e.g. a sigmoid-like curve) in the boundary of $(0.45, 2.55) \times (1.95, 4.05)$. (This is needed as per the assumption on the disturbance function.)

The parameters used for computing the abstract system are given by $\eta = [0.2\ 0.2\ 0.2]^T$, $\omega = [0.3\ 0.3]^T$ and $\tau = 0.3s$, where ω is a parameter used to discretize the input space.

ComputeAbs *vs. SCOTSv0.2.* As SCOTSv0.2 requires a uniform disturbance, we need to use the disturbance $0.1 = \sup\{0.1, 0.05\}$ in the whole state space. In contrary, our ComputeAbs takes advantage of the much smaller disturbance 0.05 appearing in a big part of the state space. This gives a more conservative abstraction with 31 million transitions for SCOTSv0.2, as opposed to a significantly less conservative abstraction with 29 million transitions for ComputeAbs (see Table 1, first line, third and fourth column).

We demonstrate this conservativeness using a controller synthesis example with reachability specification [31] for the vehicle dynamics (Fig. 1). The goal of the controller is to eventually make the vehicle reach the hatched region at the bottom right of the state space, while avoiding the gray obstacles. The cross-hatched region has the disturbances 0.1, and the rest of the state space has disturbance 0.05 (we interpolate the disturbance in the boundary by a smooth function). The purple region and the green region represent the winning domain for the controller synthesis problem when we use SCOTSv0.2 and ComputeAbs for computing the abstraction respectively. The conservativeness of the abstraction computed using SCOTSv0.2 is clear from the fact that the purple region is in this case a proper subset of the green region.

4.2 Robotic Manipulator Arm

We consider the dynamics of a two-degree-of-freedom robotic manipulator arm [7] with continuous friction (function of joint velocity) [20]. Controlling the orientation of such a manipulator has appeared in several papers [8, 25]. Since the friction varies significantly with the joint velocity, a worst case bound on the friction is prohibitive to controller design using ABCS. Hence, an abstraction that is based on non-uniform disturbance is crucial. Moreover, due to wear and tear of the joints, frictions at different parts might change over time.

The dynamics of the manipulator are given by

$$\begin{aligned}\tau_1 - \tau_F(\dot{\theta}_1) &= m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2(2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 \\ &\quad - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g s_{12} + (m_1 + m_2) l_1 g s_1 \\ \tau_2 - \tau_F(\dot{\theta}_2) &= m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g s_{12} + m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2),\end{aligned}$$

where for $i \in \{1, 2\}$, θ_i represent the angular displacements of the two arms, $\tau_i \in [-20, 20]$ represent the motor torques, and $\tau_F : \dot{\theta} \mapsto \gamma_1(\tanh(\gamma_2 r\dot{\theta}) - \tanh(\gamma_3 r\dot{\theta})) + \gamma_4 \tanh(\gamma_5 r\dot{\theta}) + \gamma_6 r\dot{\theta}$ represents the torque produced due to the friction as a function of the angular velocity $\dot{\theta}$. The values of the other parameters are given by $m_1 = 1$, $m_2 = 1$, $l_1 = 0.5$, $l_2 = 0.5$, $g = 9.8$, $\gamma_1 = 0.25$, $\gamma_2 = 100$, $\gamma_3 = 10$, $\gamma_4 = 0.1$, $\gamma_5 = 100$, $\gamma_6 = 0.01$ and $r = 0.08$. A more detailed explanation of the dynamic model and the friction model can be found in [7] and [20], respectively.

For the abstraction, we use the state space $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{4}, \frac{\pi}{4}] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{4}, \frac{\pi}{4}]$ and the discretization parameters $\eta = [\frac{\pi}{20}\ \frac{\pi}{40}\ \frac{\pi}{20}\ \frac{\pi}{40}]^T$, $\omega = [2\ 2]^T$ and $\tau = 3 \cdot 10^{-4}$, where ω is a parameter to discretize the input space.

The state dependent disturbances model the effect of the friction on the overall system which results in complex disturbance functions which we omit due to space constrains.

As can be seen from Table 1, the number of computed transitions is reduced when using ComputeAbs compared to SCOTSv0.2.

4.3 Aircraft Landing Maneuver

We consider the three-dimensional dynamical model

$$\begin{aligned}\dot{x}_1(t) &= \frac{1}{m}(u_1 \cos u_2 - D(u_2, x_1) - mg \sin x_2) + [\![-d_1(x_3), d_1(x_3)]\!] \\ \dot{x}_2(t) &= \frac{1}{m x_1}(u_1 \sin u_2 + L(u_2, x_1) - mg \cos x_2) + [\![-d_2(x_3), d_2(x_3)]\!] \\ \dot{x}_3(t) &= x_1 \sin x_2 + [\![-d_3(x_3), d_3(x_3)]\!]\end{aligned}$$

adapted from [27] where $(x_1, x_2, x_3) \in [58, 70] \times [\frac{-3\pi}{180}, 0] \times [0, 55]$ are the state variables, $(u_1, u_2) \in [0, 32000] \times [0, \frac{8\pi}{180}]$ are the control inputs, $D(u_2, x_1) = (2.7 + 3.08 \cdot (1.25 + 4.2 \cdot u_2)^2) \cdot x_1^2$, $L(u_2, x_1) = (68.6 \cdot (1.25 + 4.2 \cdot u_2)) \cdot x_1^2$,

$m = 60 \cdot 10^3$ and $g = 9.81$. A physical interpretation of variables and parameters can be found in [27].

We consider external state-dependent disturbance bounds

$$d(x_3) = \begin{cases} [0.203\ 0.001\ 0.001]^T & x_3 \in (10 - \frac{14}{334}, 10 + \frac{14}{334}) \\ [0.108\ 0.002\ 0]^T & \text{otherwise,} \end{cases}$$

where we assume that the function d is element-wise interpolated using rapidly changing smooth functions (e.g. sigmoid-like curves) in the boundary of the region $x_3 \in (10 - \frac{14}{334}, 10 + \frac{14}{334})$. The function d can be interpreted as wind turbulence, and are functions of x_3, where $d : x_3 \mapsto [d_1(x_3)\ d_2(x_3)\ d_3(x_3)]^T$. We do not consider any measurement or actuation errors.

The parameters that we use for the abstraction are given by $\eta = (\frac{25}{362}, \frac{3\pi}{80 \cdot 66}, \frac{56}{334})$, $\omega = (32000, \frac{9}{8} \cdot \frac{\pi}{180})$ and $\tau = 0.25s$.

Table 1 shows that the number of computed transitions is also significantly reduced when using ComputeAbs compared to SCOTSv0.2.

4.4 Treatment Model of Prostate Cancer

We consider a dynamical model for the personalized treatment of prostate cancer using hormone-therapy [18]. We consider a perturbed version of the model

$$\begin{aligned} \dot{x}_1 =& \left(\alpha_x \left(k_1 + \frac{(1-k_1)x_3}{x_3 + k_2} \right) - \beta_x \left(k_3 + \frac{(1-k_3)x_3}{x_3 + k_4} \right) - m_1 \left(1 - \frac{x_3}{z_0} \right) \right) x_1 \\ & + d_1(x_1, x_2, x_3) \\ \dot{x}_2 =& m_1 \left(1 - \frac{x_3}{z_0} \right) x_1 + \left(\alpha_y \left(1 - d \cdot \frac{x_3}{z_0} \right) - \beta_y \right) x_2 + d_2(x_1, x_2, x_3) \\ \dot{x}_3 =& \frac{u \cdot z_0 - x_3}{\tau} + d_3(x_1, x_2, x_3) \end{aligned}$$

where $x = (x_1, x_2, x_3) \in [0, 30] \times [0, 30] \times [0, 30]$ are the state variables, $u \in \{0, 1\}$ is the input variable modeling the mode switches and d_1, d_2, d_3 are state dependent disturbances given by

$$d(x) = 10^{-3} \times \begin{cases} [3\ 1\ 1]^T & x \in (20.05, 29.95) \times (20.05, 29.95) \times (20.05, 29.95) \\ [0.5\ 0.5\ 0.5]^T & \text{otherwise.} \end{cases}$$

where we assume that the function d is element-wise interpolated using rapidly changing smooth functions (e.g. sigmoid-like curves) in the boundary of the region $(20.05, 29.95) \times (20.05, 29.95) \times (20.05, 29.95)$.

Table 1 shows that the number of computed transitions is reduced when using ComputeAbs compared to SCOTSv0.2.

5 Conclusion

We have presented an algorithm for computing abstractions in ABCS when the disturbance in the dynamics is state- and input-dependent. Computing the abstraction in a sound but not too pessimistic manner requires a more subtle computation of growth bounds. Our experiments show that considering state-dependent disturbances can lead to more precise abstractions.

References

1. M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *HSCC 13*, pages 173–182. ACM, 2013.
2. R. Alur, T. Henzinger, O. Kupferman, and M. Vardi. Alternating refinement relations. In *CONCUR'97*, LNCS 1466, pages 163–178. Springer, 1998.
3. Y. Bai, K. Mallik, A. Schmuck, D. Zufferey, and R. Majumdar. Incremental abstraction computation for symbolic controller synthesis in a changing environment. In *2019 IEEE 58th CDC*, pages 6261–6268, 2019.
4. C. Belta, B. Yordanov, and E. A. Gol. *Formal Methods for Discrete-Time Dynamical Systems*. Springer, 2017.
5. F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *CDC 16*, pages 4661–4666. IEEE, 2016.
6. F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.
7. J. J. Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2005.
8. J. J. Craig, P. Hsu, and S. S. Sastry. Adaptive control of mechanical manipulators. *The International Journal of Robotics Research*, 6(2):16–28, 1987.
9. A. F. Filippov. On certain questions in the theory of optimal control. *SIAM A: Control*, 1(1):76–84, 1962.
10. J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *arXiv preprint arXiv:1705.01292*, 2017.
11. A. Girard. Reachability of uncertain linear systems using zonotopes. In *HSCC*, pages 291–305. Springer, 2005.
12. A. Girard. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5):947–953, 2012.
13. A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *TAC*, 55(1):116–126, 2010.
14. S. M. Harwood and P. I. Barton. Efficient polyhedral enclosures for the reachable set of nonlinear control systems. *Mathematics of Control, Signals, and Systems*, 28(1):8, 2016.
15. D. Hathaway. Using continuity induction. *The College Mathematics Journal*, 42(3):229–231, 2011.
16. T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of computer and system sciences*, 57(1):94–124, 1998.

17. K. Hsu, R. Majumdar, K. Mallik, and A. Schmuck. Multi-layered abstraction-based controller synthesis for continuous-time systems. In *HSCC 18*, pages 120–129. ACM, 2018.
18. B. Liu, S. Kong, S. Gao, and E. Clarke. Parameter identification using delta-decisions for biological hybrid systems. Technical report, CMU SCS Technical Report, CMU-CS-13-136, 2014.
19. J. Liu and N. Ozay. Finite abstractions with robustness margins for temporal logic-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 22:1–15, 2016.
20. C. Makkar, W. Dixon, W. Sawyer, and G. Hu. A new continuously differentiable friction model for control systems design. In *Intl. Conf. Advanced Intelligent Mechatronics 05*, pages 600–605. IEEE, 2005.
21. O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In *STACS'95*, volume 900 of *LNCS*, pages 229–242. Springer, 1995.
22. M. Mazo Jr., A. Davitian, and P. Tabuada. PESSOA: A tool for embedded controller synthesis. In *CAV 2010*, LNCS 6174, pages 566–569. Springer, 2010.
23. T. M. Moldovan and P. Abbeel. Safe exploration in markov decision processes. *arXiv preprint arXiv:1205.4810*, 2012.
24. P. Nilsson, N. Ozay, and J. Liu. Augmented finite transition systems as abstractions for control synthesis. *Discrete Event Dynamic Systems*, 27(2):301–340, 2017.
25. P. Ouyang, V. Pano, J. Tang, and W. Yue. Nonlinear pd-type control in position domain. In *ICIEA, 2016 IEEE 11th Conference on*, pages 2407–2412. IEEE, 2016.
26. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL 89*, pages 179–190. ACM, 1989.
27. G. Reissig, A. Weber, and M. Rungger. Feedback refinement relations for the synthesis of symbolic controllers. *TAC*, 62(4):1781–1796, 2017.
28. M. Rungger and G. Reissig. Arbitrarily precise abstractions for optimal controller synthesis. In *CDC 17*, pages 1761–1768. IEEE, 2017.
29. M. Rungger and M. Zamani. SCOTS: A tool for the synthesis of symbolic controllers. In *HSCC*, pages 99–104. ACM, 2016.
30. M. Rungger and M. Zamani. Accurate reachability analysis of uncertain nonlinear systems. In *HSCC*, pages 61–70. ACM, 2018.
31. P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer, 2009.

A The proof of Theorem 3

For the first part of Theorem 3 ($\overline{v}(t;p,r,u,\delta) \geq w(t;p,u,z^*(t))$ holds for all $t \in [0,\tau]$), we use the principle of continuity induction:

Lemma 2. *(Principle of Continuity Induction [15]) Let $\psi(x)$ be a predicate with a free real-valued variable x, and let $[a,b]$ be a closed interval. If*

1. *$\psi(a)$ holds,*
2. *for any $x \in [a,b)$, if $\psi(y)$ holds for all $a \leq y \leq x$, then there exists a $\delta > 0$ such that $\psi(y)$ holds for all $y \in [x, x+\delta)$, and*
3. *for any $x \in (a,b]$, if $\psi(y)$ holds for all $a \leq y < x$, then $\psi(x)$ holds,*

then $\psi(x)$ holds for all $x \in [a,b]$.

Define a set $S := \{y \in [0,\tau] \mid \overline{v}(y;p,r,u,\delta) \geq w(y;p,u,z^*(y))\}$. We show that $S = [0,\tau]$ using the principle of continuity induction. We need to prove the following three properties of S:

(a) We have to show that $0 \in S$. At $t = 0$, we have both $z(0) = r$ and $\varepsilon = r$, implying $\overline{v}(y;p,r,u,\delta) = w(y;p,u,z^*(y))$. Hence, $0 \in S$.
(b) Let $x \in [0,\tau)$ s.t. $[0,x] \subset S$. We show that there exists a $\delta' > 0$ s.t. $[x, x+\delta') \subset S$. Let $\delta' \in \min\{\delta, \tau - x\}$. Then for all $t \in [x, x+\delta')$, there is $y = \max\{0, t-\delta\} \in [0,x]$ and $\delta^- = \min\{t,\delta\}$ s.t. $t = y + \delta^-$. Since $y \in S$ by assumption, by the definition of S, we have $\overline{v}(y;p,r,u,\delta) \geq w(y;p,u,z^*(y))$, which implies $\overline{z}(y;p,r,u,\delta) \geq z^*(y;p,r,u)$. Then $\varepsilon = z_{\max}(\delta^-; \overline{z}(y;p,r,u,\delta), u) \geq z^*(\delta^-; \varphi_{pu}(y), z^*(y;p,r,u), u) = z^*(t; \varphi_{pu}(t), r, u)$ (the first inequality follows by observing that $r > r' \Rightarrow z_{max}(\cdot; r, \cdot) > z^*(\cdot; \cdot, r', \cdot)$, and the last equality follows from the definition of $z^*(\cdot;\cdot,\cdot,\cdot)$).
(c) Finally, let $x \in (0,\tau]$ and $[0,x) \subset S$. We have to show that $x \in S$. Let $z = \min\{0, x-\delta\}$. By assumption, $z \in S$. Using similar argument as in (b), it can be shown that $x \in S$.

By the principle of continuity induction, we have that $S = [0,\tau]$, i.e. the first part of Thm. 3 holds.

For the second part, we need to show that given a fixed $s \in [0,\tau]$, and given an $\varepsilon > 0$, we can find a $\delta > 0$ s.t. $\mathcal{M}(\delta) := |\overline{z}(s;p,r,u,\delta) - z^*(s;p,r,u)| < \varepsilon$. The proof is by a continuity argument of $\mathcal{M}(\cdot)$ in the interval $[0,s)$: Since $\mathcal{M}(\cdot)$ is also non-negative and $\mathcal{M}(0) = 0$ (it is easy to verify that $\overline{z}(\cdot;\cdot,\cdot,\cdot,0) = z^*(\cdot;\cdot,\cdot,\cdot)$), so a continuous $\mathcal{M}(\delta)$ will allow us to find a δ to have an arbitrarily small and positive $\mathcal{M}(\delta)$. Following is the proof that $\mathcal{M}(\cdot)$ is indeed continuous.

Since for all $\delta > 0$, $\overline{z}(s;p,r,u,\delta) \geq z^*(s;p,r,u)$ (see Rem. 1), we can drop the $|\cdot|$ and have $\mathcal{M}(\delta) = \overline{z}(s;p,r,u,\delta) - z^*(s;p,r,u)$ instead. Let $s^- := s - \delta$ and $\delta \in [0,s)$, and consider the following derivation: $\mathcal{M}(\delta) = \overline{z}(s;p,r,u,\delta) - z^*(s;p,r,u) = e^{L(u)\delta}[\overline{z}(s^-;p,r,u,\delta) - z^*(s^-;p,r,u)] + \int_0^\delta e^{L(u)(\delta-t)}\left[w_{\max}(u) - w(s^- + t; \varphi_{pu}(s^-), u, z^*(s^- + t))\right] dt$. Since $\overline{z}(\cdot;p,r,u,\delta)$ and $z^*(\cdot;p,r,u)$ are continuous functions, and $e^{L(u)\delta}$ is continuous in δ, it follows that the first part of $\mathcal{M}(\delta)$ is also continuous in δ. Define $\mathcal{I}(\delta,t) := e^{L(u)(\delta-t)}\left[w_{\max}(u) - w(s^- + t; \varphi_{pu}(s^-), u, z^*(s^- + t))\right]$. Note that, since $\varphi_{pu}(\cdot)$, $z^*(\cdot)$ and $d(\cdot,u)$ are continuous functions, hence for a fixed t, $w(s^- + t; \varphi_{pu}(s^-), u, z^*(s^- + t))$ is a continuous function of δ. In addition, for a fixed t, $e^{L(u)(\delta-t)}$ is also a continuous function of δ. So $\mathcal{I}(\cdot,t)$ is also a continuous function.

Now in order to show the continuity of $\int_0^\delta \mathcal{I}(\delta,t)dt$ in δ, we first fix a $\mu > 0$, and then show that there exists a $\nu > 0$, s.t. for any two $\delta, \delta' \in [0,s)$ with $|\delta - \delta'| < \nu$, $\left|\int_0^\delta \mathcal{I}(\delta,t)dt - \int_0^{\delta'} \mathcal{I}(\delta',t)dt\right| < \mu$. W.l.o.g. let us assume that $\delta > \delta'$ (otherwise interchange the role of δ and δ'). Let $k > 0$ be a constant s.t. $k < \mu/s$. Now since $\mathcal{I}(\cdot,t)$ is continuous for a fixed t and $\mathcal{I}(\delta,\cdot)$ is bounded and continuous over $[0,s]$ for a fixed δ, hence given k there exists a $l > 0$ s.t. for all $t \in [0,s]$, whenever $|\delta - \delta'| < l$, $|\mathcal{I}(\delta,t) - \mathcal{I}(\delta',t)| < k$. Let $\nu < \min\{l, (\mu - ks)/(e^{L(u)s} w_{\max}(u))\}$,

$$
\begin{aligned}
&\left| \int_0^{\delta} \mathcal{I}(\delta, t)dt - \int_0^{\delta'} \mathcal{I}(\delta', t)dt \right| \\
&= \left| \int_0^{\delta'} \mathcal{I}(\delta, t)dt + \int_{\delta'}^{\delta} \mathcal{I}(\delta, t)dt - \int_0^{\delta'} \mathcal{I}(\delta', t)dt \right| \\
&\leq \left| \int_0^{\delta'} \mathcal{I}(\delta, t)dt - \int_0^{\delta'} \mathcal{I}(\delta', t)dt \right| + \left| \int_{\delta'}^{\delta} \mathcal{I}(\delta, t)dt \right| \\
&\leq \int_0^{\delta'} \left| \mathcal{I}(\delta, t)dt - \mathcal{I}(\delta', t) \right| dt + \int_{\delta'}^{\delta} \left| \mathcal{I}(\delta, t) \right| dt \\
&< \int_0^{\delta'} k dt + \int_{\delta'}^{\delta} \left| e^{L(u)s} w_{\max}(u) \right| dt \\
&\leq \int_0^{s} k dt + \int_{\delta'}^{\delta} e^{L(u)s} w_{\max}(u) dt \\
&< ks + e^{L(u)s} w_{\max}(u) \nu \\
&< ks + e^{L(u)s} w_{\max}(u) \cdot \frac{\mu - ks}{e^{L(u)s} w_{\max}(u)} \;=\; \mu.
\end{aligned}
$$

Fig. 3. Derivation of continuity of $\int_0^{\delta} \mathcal{I}(\delta, t)$

and note that $\nu > 0$ and $|\delta - \delta'| < \nu$ also implies $|\mathcal{I}(\delta, t) - \mathcal{I}(\delta', t)| < k$. Assume $|\delta - \delta'| < \nu$. Fig. 3 shows the continuity of $\int_0^{\delta} \mathcal{I}(\delta, t)$.