Noname manuscript No.
(will be inserted by the editor)

Finite abstractions for hybrid systems with stable continuous dynamics

Herbert G. Tanner · Jie Fu · Chetan Rawal · Jorge L. Piovesan · Chaouki T. Abdallah

the date of receipt and acceptance should be inserted later

Abstract This paper outlines an abstraction process in which a particular class of hybrid automata with continuous dynamics that have parameterized positive limit sets, are being abstracted into finite transition systems. The limit sets with their corresponding attraction regions define pre- and post-conditions for the continuous dynamics, and determine the transitions in the discrete abstraction. An observable (weak) bisimulation equivalence is established between the two models. The abstraction process described can find application in verification, as well as in planning and symbolic control.

Keywords Abstraction · hybrid systems · symbolic control

1 Introduction

An important question in the context of hybrid systems is whether there is a process of *abstraction*, that would enable the designer to somehow ignore the behavior of the underlying component continuous dynamics of the hybrid system, and capture its behavior of interest in a purely discrete model of computation. Such a discrete model can potentially simplify the solution of problems related to verification, planning and control design for hybrid systems, which in the general case are computationally intractable.

For an abstract model to be of use, it needs to share the properties of interest with the concrete system: if a property is verified on the abstraction, it should hold true

The work of the first author is supported by the National Science Foundation under grant # 0447898. The work of the other authors was also supported by the National Science Foundation (CNS0626380) under the FIND initiative.

Herbert Tanner, Chetan Rawal, and Jie Fu
Mechanical Engineering
University of Delaware
E-mail: btanner@udel.edu

Jorge L. Piovesan and Chaouki T. Abdallah
Electrical and Computer Engineering
University of New Mexico
E-mail: chaouki@ece.unm.edu

for the concrete one; if a controller is designed based on the abstract model, it should be implementable and yield the same outcome on the original system. Therefore, it is critical to identify not only the process that would yield the abstraction, but also the requirements that enable the propagation of properties and designs between the two representations.

Abstraction, as a concept, has been used as a means of obtaining simpler representations of system behavior, which however preserve some properties of interest to the designer. The need for reigning in complexity has become apparent in model checking and verification of general classes of hybrid systems [1, 2], particularly when direct reachability computation is involved [3–6]. Abstraction tools typically overapproximate the reachable regions of the state space [7]. Non-conservative approaches [8] also exist, for analysis from a finite set of initial conditions. There is significant computational complexity associated with the application of reachability computation tools, which is primarily due to the need for predicting what the continuous dynamics of the hybrid system does. In this context, abstraction offers a way to facilitate computations by enabling the analysis to be performed on a smaller system, and allowing the generalization of the conclusions to the larger system. The ability to preserve properties between the two models of different complexity is typically based on the particular equivalence relations established.

There are a number of discrete abstraction approaches reported in literature, and they fall under two main categories. Both create partitions of the continuous state space, but the difference is their starting points. Partition approaches based on simulation, bisimulation [9] or their approximate equivalents [10, 11], focus on the dynamics, and attempt to group together continuous states that evolve in a similar way. The survey [12] demonstrates that in order to obtain bisimulation relations for general hybrid systems, one has to severely restrict either the discrete logic that governs the transitions, or the type of continuous dynamics, or both. Certain undecidability results [12] indicate the limits of bisimulation-based abstraction. Such results motivate less restrictive conditions, provided by simulation relations [13]. In such cases, one may choose to abandon the search for input-output equivalence in the hope of obtaining some property inclusion. The concept of approximate bisimulation [14] characterizes the case where the trajectories of the two related systems under the abstraction map are not required to match exactly, but they are rather allowed to be "close" in a Lyapunov-like sense. Approximately bisimilar finite symbolic models have been constructed [15] based on an incremental stability property of the underlying the hybrid system. The other category of approaches to obtaining discrete abstractions is based on *a priori* partitioning the state space based on properties of each block which may be of interest [16–21]. Among this body of work, [18, 20] make explicit use of a state quantizer.

In equivalence relation-based approaches, partitions are dictated by dynamics — they yield simple models, which may be difficult to construct and may partition the space in non-intuitive ways; in *a priori* partitioning blocks are user-defined, which allows the underlying dynamics to induce too many transitions in the discrete model. The approach in this paper is in-between: partition blocks are defined *a priori* based on logical predicates, but at the same time respecting the asymptotic behavior of the continuous dynamics. The abstraction approach therefore is reminiscent of predicate abstraction (cf. [16]), but at the same time allows a formal link between the concrete and abstract models in the form of a (weak) bisimulation relation, which is the main technical result (Theorem 1) of this paper.

To be able to characterize the type of discrete computation models obtained from the proposed abstraction process, we limit our scope to a specific class of hybrid systems, in which the continuous dynamics has parameterized attractors. Through the parameters, the designer determines the location of these attractors for the continuous dynamics, and thus exercises control over the hybrid system by forcing certain guards to be activated. The switching that occurs between the different continuous dynamics components (modes) is not arbitrary, but meets specific requirements formalized in sets of first-order logic propositions involving continuous states, discrete states, and parameters. Therefore, by resetting these parameters on-line, and without modifying the structure of the component continuous dynamics, the designer not only maintains authority to steer the continuous dynamics, but also controls the discrete transitions between modes. By establishing the weak bisimulation equivalence between the hybrid system and its abstraction, this paper offers a finite, discrete model that can be potentially used for verification, as well as for symbolic control synthesis. In contrast to existing stability-based finite abstraction methods [15], the proposed does not rely on the stability of the overall hybrid system, but rather exploits the *local* stability of individual modes. In this context, overall asymptotic stability —although provable— is irrelevant. The rationale is that in a bottom-up scheme such as this one is that if well designed specialized stand-alone control laws are available, there should be a way to combine them in order to perform a more complex task without having to explicitly analyze the ensuing hybrid system.

Section 2 sets the stage for the technical discussion by providing necessary definitions, introducing notation, and describing the computation models used. In Section 3 the equivalence between the concrete and abstract models is established. In Section 4 the proposed approach is applied to the case of a robotic system that needs to execute a fairly complex task. Concrete and abstract models are derived, and it is shown how the abstract model can match the evolution of the concrete system. Section 5 gives a short summary of the main idea behind the proposed approach and highlights ongoing and future research directions.

2 Preliminary definitions and notation

Let us first start with a general definition for a hybrid system, so that it can later be compared with the special case considered in this paper.

Definition 1 ([22]) A hybrid automaton H is a collection $H = (Q, X, D, f, \text{Init}, E, G, R)$, in which

Q	is a finite set of discrete variables;
$X \subset \mathbb{R}^n$	is a set of continuous variables;
$D : Q \to P(X)$	is a continuous domain, where $P(X)$ is the powerset of X;
$f : Q \times D(Q) \to TX$	is a vector field;
$\text{Init} \subseteq Q \times X$	is the set of initial states;
$E \subseteq Q \times Q$	is a set of edges;
$G : E \to P(X)$	is the guard condition;
$R : E \times X \to P(X)$	is the reset map.

The pair $(q, x) \in Q \times X$ is called the state of **H**.

In Definition 1, the domain $D(q)$ is not required to be (positively) invariant for $f(q,x)$ —this is one of the main differences compared to our model. The notion of positive invariance is understood in terms of the limiting behavior of the solutions (flows) of $\dot{x} = f(q,x)$ as follows:

Definition 2 ([23]) A curve $\sigma_x : \mathbb{R} \to X$, with $x \in X$, is an integral curve of vector field f if $\dot{\sigma}_x(t) = f(\sigma_x(t))$, and $\sigma_x(0) = p$. If $I(f,x)$ is the largest time interval for which an integral curve through x can be defined, then the integral closure $\sigma_x(t) : I \to X$ is a maximal integral curve for f and the flow of the vector field f is defined as the map $\Phi_t : X \to X$; $\Phi_t(x) \mapsto \sigma_x(t)$.

Definition 3 ([24]) Let $\Phi_t(x)$ be the flow of f passing through $x \in X$ at $t = t_0$. Then $z \in X$ is said to be a positive limit point of $\Phi_t(x)$ if there is a sequence $\{t_n\}$, with $t_n \to \infty$ as $n \to \infty$, such that $\Phi_{t_n}(x) \to z$ as $n \to \infty$. The set of all limit points of $\Phi_t(x)$, $x \in X$ is called the positive limit set of Φ_t, denoted L^+.

In the context of this paper, the vector field f is assumed to approach some subset $\mathcal{A}$ of D, and its approach is being quantified by means of a distance:

Definition 4 ([24]) The distance of a point x to a set $\mathcal{A} \subset X$ is denoted $\mathrm{dist}(x, \mathcal{A})$ and is defined as $\mathrm{dist}(x, \mathcal{A}) \triangleq \inf_{y \in \mathcal{A}} \|y - x\|$.

The norm $\|\cdot\|$ is assumed to be the one induced in X by a typical norm on $\mathbb{R}^n$. Positive invariance can now be formally described as follows.

Definition 5 ([24]) A set X is (positively) invariant if $\Phi_{t_0}(x) \in X \Rightarrow \Phi_t(x) \in X$, for all $t \geq t_0$.

It can be shown [24], that for a time-invariant vector field f that evolves in a *compact* set D for all time, the existence of a limit set L^+ is guaranteed, and it is known that L^+ will be nonempty, compact, positively invariant, and *attractive*, in the sense that for every $\epsilon > 0$, there exists a $T > 0$ such that $\mathrm{dist}(x(t), L^+) < \epsilon$. So with reference to Definition 1, let us assume that:

Assumption 1 *The set X is compact.*

We now define a specific class of hybrid systems, in which we can also have an $r < \infty$ dimensional boolean vector $\ell \in \mathcal{L} \subseteq \{\mathbf{0}, \mathbf{1}\}^r$, and where the vector fields are parameterized, with a vector of parameters in a set $\mathcal{P} \subseteq \mathbb{R}^m$. The parameters in f are assumed to determine the shape and position of its limit set [25].

Definition 6 (Hybrid agent) The hybrid agent is a collection $\mathbf{H} = \left\{\mathcal{H}, \mathcal{P}, \mathcal{K}, \mathcal{I}, \mathcal{AP}, f, \overleftarrow{\cdot}, \overrightarrow{\cdot}, s, \Delta\right\}$. In this collection,

$\mathcal{H} \subset X \times \mathcal{L}$	where $X \subset \mathbb{R}^n$ is compact, is the set of hybrid states;
$\mathcal{P}$	is the set of admissible parameter vectors;
$\mathcal{K}$	is a finite set of system modes;
$\mathcal{I} \subseteq \mathcal{K} \times \mathcal{P}$	is the set of control inputs;
$\mathcal{AP}$	is a finite, indexed set of atomic logical propositions;
$f : \mathcal{H} \times \mathcal{P} \times \mathcal{K} \to T\mathcal{X}$	is a family of locally Lipschitz vector fields indexed by $\mathcal{K}$ and parameterized by $\mathcal{P}$;
$\overleftarrow{\cdot} : \mathcal{K} \to 2^{\mathcal{AP}}$	is the set of pre-conditions for each mode;
$\overrightarrow{\cdot} : \mathcal{K} \to 2^{\mathcal{AP}}$	is the set of post-conditions for each mode;
$s : \mathcal{H} \to 2^{\mathcal{P}}$	is the parameter reset map, and
$\Delta \in (\mathcal{H} \times \mathcal{P} \times \mathcal{K})^2$	is the transition relation, also denoted $\to$.

Note that since X is compact, local Lipschitz continuity is sufficient for global existence and uniqueness of the solutions of f.

The "state" (or configuration) in $\mathbf{H}$ is a triplet $(x, \ell, k) \in \mathcal{H} \times \mathcal{K}$ (c.f. $(x, q) \in X \times Q$ in H of Definition 1), and the first two elements in (x, ℓ) will be referred to as the *hybrid state* in the sequel. In the new model, $\mathcal{H}$ can be viewed as an extension of X that includes boolean variables. Trajectories, or *executions* of $\mathbf{H}$, describe the evolution of the variables that belong in X and $\mathcal{K}$, and are interpreted in the same way as for the model of Definition 1 [22]. Note that no initial states are specified in this model. The semantics for the new components of $\mathbf{H}$ can be described in more detail as follows.

Logical variables in $\mathcal{L}$ represent system states which are more appropriately evaluated as true or false (e.g. the light is on, the battery is charged) and where the additional resolution of a continuous domain would be superfluous.

Each element $k \in \mathcal{K}$ can be thought to be associated with a particular closed-loop control law for the continuous dynamics. These controllers may use as parameters elements of the vector $p \in \mathcal{P}$ (control gains, for example, could be included in p). Here, the positive limit sets L^+ for each $f(h, p, k)$ are assumed given, having been designed in the process of closing the control loop in $f(\cdot)$ for each k. The dependence of L^+ on p and k is denoted by writing $L^+(k; p)$. For every $k \in \mathcal{K}$, it is assumed that $L^+(k; p)$ is path connected, uniformly in $p \in \mathcal{P}$; otherwise, if there are $B(k) > 1$ disconnected components of $L^+(k; p)$ they will each have their own attraction region $\mathcal{A}_b^+$,[1] for $b = 1, \ldots, B(k)$, in which case one associates a different symbol in $\mathcal{K}$ for the restriction of $f(h, p, k)$ on each attraction region. By expanding $\mathcal{K}$ in this way, it is ensured that there is a single path connected positive limit set for every continuous dynamics indexed by $k \in \mathcal{K}$.

The set $\mathcal{AP}$ consists of logical statements that involve hybrid states, parameters, and modes, and express in predicate form all environmental and system conditions which are of interest for the system. For example, the pre and post-conditions (denoted Pre and Post, respectively) of each mode (read controller) can be thought of as

[1] The region of attraction of a particular component L_b^+ of the positive limit set L^+ of the flow of vector field f in X can be defined as [26] $\mathcal{A}_b^+(X) \triangleq \{x \in X \mid \lim_{t \to \infty} \operatorname{dist}(\Phi_t(x), L_b^+) = 0\}$.

the guards and limit sets in each domain, respectively. We write $(h,p) \models \overleftarrow{k}$ if the pair (h,p) satisfies all atomic propositions in $\textsc{Pre}(k)$, and similarly $(h,p) \models \overrightarrow{k}$ if (h,p) satisfies the $\textsc{Post}(k)$. In the case of different isolated components for $L^+(k;p)$ mentioned above, the inclusion $x \in \mathcal{A}_b^+ \subset X$ constitute the pre-condition of k in sense that for some valuation of the boolean vector $\ell \in \mathcal{L}$, and for $p \in s\big((x,\ell)\big)$, $(x,\ell,p) \models \overleftarrow{k} \Rightarrow \lim_{t\to\infty} \mathrm{dist}\big(\Phi_t(x), L^+(k;p)\big) = 0$. For $k \in \mathcal{K}$, $\textsc{Post}(k)$ is a set of predicates that define the positive limit set of the continuous dynamics when driven by the control law indexed by k, in the sense that $x \in L^+(k;p) \Rightarrow (x,\ell,p) \models \overrightarrow{k}$ for some valuation of the boolean vector $\ell \in \mathcal{L}$. For a given $k \in \mathcal{K}$ and $p \in \mathcal{P}$, the parameterized vector field $f(h,p,k)$ with $h = (x_1,\ell)$ may result in $\lim_{t\to\infty} \Phi_t(x_1) = x_2$; in this case we write $h_1 \overset{k[p]}{\dashrightarrow} h_2$, where $h_2 = (x_2,\ell)$.

The reset map s does *not* capture discontinuous jumps in continuous states as in Definition 1; in fact, in Definition 6 continuous states in X are *not* supposed to change discontinuously. Rather, s takes the form of a set valued map that determines all the possible values that the parameter vector can take at a particular hybrid state h. It is the parameter vector p, therefore, that can be instantaneously "reset" from one value in $s(h)$ to another value in $s(h)$, thus allowing a transition from a current mode to a different one according to the transition relation Δ.

The transition relation Δ allows a jump from mode k to mode k' on a reset of the parameter vector from p to p', denoted $(h,p,k) \to (h,p',k')$, if and only $(h,p) \models \overrightarrow{k}$ *and* $(h,p') \models \overleftarrow{k'}$. Transitions in $\mathbf{H}$ express possible switches in the controllers of the continuous dynamics. We require that the limit sets of the flows of the vector field driven by k be properly contained in the set of continuous variables that satisfy $\textsc{Pre}(k')$, and we express this formally by assuming a nonzero constant $d > 0$ for which $\forall x \in L^+(k;p)$ we have $\mathrm{dist}\big(x, \{x' \mid (x',\ell,p) \models \overleftarrow{k'}\}\big) = 0$ (x is in $\textsc{Pre}(k')$) and $\mathrm{dist}\big(x, \partial\{x' \mid (x',\ell,p) \models \overleftarrow{k'}\}\big) > d$ (the distance from the boundary of $\textsc{Pre}(k')$ is larger than d).

The inputs $(k,p) \in \mathcal{I}$ (denoted $k[p]$ in the sequel), are instructions given to the machine in the form of a desired continuous dynamics to be activated, with the specific parameterization. Input instructions are processed by the machine when continuous dynamics have reached their limit set. Then the hybrid state h and parameter p at mode k give $(h,p) \models \overrightarrow{k}$. At that point, the machine determines whether it can process the next input by checking whether there is a pair in Δ that matches. A sequence of inputs $\{k_i[p_i]\}_{i=1}^n$ is called *admissible* if all the elements in the sequence can be sequentially executed by the machine.

The discrete computation model which will serve as an abstraction is a finite transition system,[2] defined as follows:

Definition 7 (Labeled transition system) $\mathbf{T} = (\mathcal{Q}, \Sigma, \delta)$ consists of

$\mathcal{Q}$	a finite set of states;
Σ	a finite alphabet;
$\delta \subseteq \mathcal{Q} \times \Sigma \times \mathcal{Q}$	the transition relation.

If $(v,\sigma,v') \in \delta$ we write $v \xrightarrow{\sigma}_T v'$. Let $\Sigma_\tau \subset \Sigma$ and call the transitions $(v,\sigma,v') \in \delta$ for which $\sigma \in \Sigma_\tau$, *silent*. Any input word of the form $u\,\sigma\,w$, where $u,w \in \Sigma_\tau^*$ (the Kleene closure of Σ_τ) and $\sigma \in \Sigma \setminus \Sigma_\tau$ will be called a *composite transition* of $\mathbf{T}$ if there are

[2] This is in fact a semi-automaton, since there is no specification for initial and final states.

$q_1, \ldots, q_n \in Q$, not necessarily distinct, such that $q_1 \xrightarrow{\sigma_1}_T q_2 \xrightarrow{\sigma_2}_T q_3 \cdots q_{n-1} \xrightarrow{\sigma_{n-1}}_T q_n$ and $\sigma_1 \cdots \sigma_{n-1} = u\,\sigma\,w$. In this case we write $q_1 \overset{\sigma}{\rightsquigarrow} q_n$.

3 Stability-based discrete abstractions

Our approach to partitioning the hybrid state space $\mathcal{H}$ for each $k \in \mathcal{K}$ is based on which subsets of $\mathcal{H}$ we can distinguish using the atomic propositions in $\mathcal{AP}$ as observations. Each such subset can be properly defined as an equivalence class composed of elements that have the same image under a binary vector-valued function. To this end, let us first define a map that associates each pair (h, p) to a binary vector in $\{1, 0\}^{|\mathcal{AP}|}$. The desired equivalence relation will be the one naturally induced by this function.

Definition 8 (Valuation map) $V_M\colon \mathcal{H} \times \mathcal{P} \mapsto \mathcal{V} \subseteq \{1, 0\}^{|\mathcal{AP}|}$ is a function that maps pairs (h, p) of hybrid states and parameters of $\mathbf{H}$, to binary vectors v formed by evaluating each atomic proposition $\alpha_i \in \mathcal{AP}_i$, such that $v[i] = 1 \Leftrightarrow (h, p) \models \alpha_i$.

Define a relation $\mathfrak{R}$ on $\mathcal{H} \times \mathcal{V}$ according to which a state $h \in \mathcal{H}$ is related to $v \in \mathcal{V}$ if there exists a $p \in s(h)$ such that $(h, p) \models v$, in which case we write $(h, v) \in \mathfrak{R}$.[3] Let $\Sigma \triangleq \mathcal{K} \cup \{\tau_1, \ldots, \tau_{|\mathcal{K}|}\}$, where τ_i is an additional symbol (label), and define the homomorphism (with respect to concatenation) $\phi : \mathcal{K} \to \Sigma^*$ in which $k \mapsto \tau_k\, k$. The proposed abstraction of the hybrid agent $\mathbf{H}$ in Definition 6 is a finite labeled transition system $\mathbf{T}(\mathbf{H})$, with ϕ mapping inputs of $\mathbf{H}$ to labels in $\mathbf{T}(\mathbf{H})$.

Definition 9 (Induced transition system) A hybrid agent $\mathbf{H}$ induces a finite labeled transition system $\mathbf{T}(\mathbf{H})$ in which

$$
\begin{aligned}
&\mathcal{Q} = V_M(\mathcal{H}, \mathcal{P}) && \text{is a finite set of states;}\\
&\Sigma = \mathcal{K} \cup \{\tau_1, \ldots, \tau_{|\mathcal{K}|}\} && \text{is a finite set of labels;}\\
&\delta \subseteq \mathcal{Q} \times \Sigma \times \mathcal{Q} && \text{is a transition relation denoted } \rightarrow_T,
\end{aligned}
$$

in which for $h \in \mathcal{H}$, $p \in s(h)$, $k \in \mathcal{K}$, and with $V_M(h, p) = v$,

$$
v \xrightarrow{\sigma}_T v' \Leftrightarrow \begin{cases} \exists h' \in \mathcal{H}, p \in s(h') : (h, p) \models \overleftarrow{\sigma}, V_M(h', p) = v', h \overset{\sigma[p]}{\dashrightarrow} h'; & \sigma \in \mathcal{K} \\ \exists\, p' \in s(h) : V_M(h, p') = v', (h, p, k) \rightarrow (h, p', k'); & \sigma = \tau_{k'}, k' \in \mathcal{K}. \end{cases}
$$

Note that each transition in $\mathbf{H}$ on input $k[p]$ is associated with *two* consecutive transitions in $\mathbf{T}(\mathbf{H})$, the first labeled τ_k and the second labeled k. However, with information about p being abstracted away, transitions in $\mathbf{T}(\mathbf{H})$ can be nondeterministic: a τ_k transition from a given state in $\mathbf{T}(\mathbf{H})$, may lead to several different states $q \in \mathcal{Q}$ from which a subsequent k transition can be initiated. Thus, an input sequence in $\mathbf{H}$ does not *uniquely* specify a path in $\mathbf{T}(\mathbf{H})$. Nevertheless, Lemma 1 and Theorem 1 that follow indicate that there is still a lot of structure of $\mathbf{H}$ that is maintained in $\mathbf{T}(\mathbf{H})$ to the point that both verification and design (under appropriate restrictions on how the transitions of $\mathbf{T}(\mathbf{H})$ are defined, which are discussed later) can be performed using the abstraction.

As a first step, it is shown that the partition on $\mathcal{H}$ induced by V_M has the property that if h can take a transition, then all h' in the same equivalence class can also take the same transition:

[3] Alternatively, we can define $\mathfrak{R}$ as an equivalence relation in $\mathcal{H}$ by identifying it with the map ${V_M}^{-1} \circ V_M(\cdot, s(\cdot))$.

Lemma 1 *Suppose that $h, h' \in \mathcal{H}$ are both $\mathfrak{R}$-related to $v \in \mathcal{V}$. If $(h,p,k) \rightarrow (h,p,k')$ for some $k, k' \in \mathcal{K}$, then there exists a $p' \in s(h)$ such that $(h',p',k) \rightarrow (h',p',k')$.*

Proof : Writing $V_M(h,p) = v$ implies that a specific combination of atomic propositions evaluate true at state h when the parameter vector is set to p; without loss of generality, let this set of propositions that are true be $\{\alpha_1, \ldots, \alpha_m\}$. From $(h,p,k) \rightarrow (h,p,k')$ we conclude that $(h,p) \models \overrightarrow{k}$ and $(h,p) \models \overleftarrow{k'}$. It follows that $\overrightarrow{k} \subseteq \{\alpha_1, \ldots, \alpha_m\} \supseteq \overleftarrow{k'}$. Given that h' is also $\mathfrak{R}$-related to v, there exists a p' such that $V_M(h',p') = v$; this means that the same set of atomic propositions that are true when the system is at state h with parameter p, are also true at state h' with parameter p'. Therefore, we must have $(h',p') \models \overrightarrow{k}$ and $(h',p') \models \overleftarrow{k'}$. According to the expression of Δ in Definition 6, $(h',p',k) \rightarrow (h',p',k')$.

Consistency between the concrete system and its abstraction is established in terms of a version of weak bisimulation [27], which is sometimes referred to as *observable bisimulation* [28]. It applies to cases where some of the transitions in a system are observable, and some are not (unobservable transitions are the ones called silent). An external observer is only able to record the sequence of labels that are associated to the observable transitions, and can have no information about how many silent transitions may have been taken in between. Two systems are *observably bisimilar* when the language generated from the observable labels in both machines is the same. Alternatively, one can view observable bisimulation as a game, in which any one of the two systems can match the observable transitions of the other one-to-one, and where any silent transitions do not "count." In an observable bisimulation "game," the system that moves first is allowed to take a composite transition.

Definition 10 (Observable bisimulation [28]) Consider two (labeled) transition systems over the same input alphabet Σ_T, $\mathbf{T}_1 = (\mathcal{Q}_1, \Sigma_T, \rightarrow_1, I_1)$ and $\mathbf{T}_2 = (\mathcal{Q}_2, \Sigma_T, \rightarrow_2, I_2)$, and let $\Sigma_\tau \subset \Sigma_T$ be a set of input symbols that trigger silent transitions. A (total) binary relation $\mathfrak{R}$ on $\mathcal{Q}_1 \times \mathcal{Q}_2$ is an *observable bisimulation* if

1. $q_1 \overset{\sigma}{\rightsquigarrow}_1 q_1' \quad \Rightarrow \quad \exists\, (q_1', q_2') \in \mathfrak{R} : q_2 \overset{\sigma}{\rightsquigarrow}_2 q_2'.$
2. $q_2 \overset{\sigma}{\rightsquigarrow}_2 q_2' \quad \Rightarrow \quad \exists\, (q_1', q_2') \in \mathfrak{R} : q_1 \overset{\sigma}{\rightsquigarrow}_1 q_1'.$

Then $\mathbf{T}_1$ and $\mathbf{T}_2$ are called observably bisimilar and we write $\mathbf{T}_1 \approx \mathbf{T}_2$.

The key result in this paper is that the discrete abstraction obtained for the hybrid agent in the form of the finite label transition system of Definition 9, and the concrete hybrid agent are observably bisimilar.

Theorem 1 *There exists a total observable bisimulation $\mathfrak{R}$ between a hybrid agent $\mathbf{H}$ and its induced finite labeled transition system $\mathbf{T}(\mathbf{H})$ and in the sense that*

1. *If $(h,v) \in \mathfrak{R} \subset \mathcal{H} \times \mathcal{Q}$ and $h \overset{k[p]}{\dashrightarrow} h'$, then $\exists\, v' \in \mathcal{Q}$, $v \overset{k}{\rightsquigarrow} v'$ and $\left(h', v'\right) \in \mathfrak{R}$.*
2. *If $(h,v) \in \mathfrak{R} \subset \mathcal{H} \times \mathcal{Q}$ and $v \overset{k}{\rightsquigarrow} v'$, then $\exists\, h' \in \mathcal{H}$ such that $h \overset{k[p]}{\dashrightarrow} h'$ and $\left(h', v'\right) \in \mathcal{R}$.*

Then we write $\mathbf{T} \approx \mathbf{H}$.

Proof : To show that $\mathfrak{R}$ is total we note first that by construction,

$$\mathcal{Q} = \left\{ V_M(h,p), \forall (h,p) \in \mathcal{H} \times \mathcal{P} \right\} \ .$$

Thus given $v \in \mathcal{Q}$, there *must* exist a hybrid state $h \in \mathcal{H}$ such that $V_M(h,p) = v$. To show the existence along the other direction, namely that for each $h \in \mathcal{H}$ there is some $v \in \mathcal{Q}$ such that $(h,v) \in \mathfrak{R}$, note that for an arbitrary state $h^* \in \mathcal{H}$, $\{(h^*,p) \mid p \in \mathcal{P}\} \subseteq \mathcal{H} \times \mathcal{P}$ implies $\{V_M(h^*,p) \mid p \in \mathcal{P}\} \subseteq V_M(\mathcal{H} \times \mathcal{P}) = \mathcal{Q}$. Meanwhile, for all $p \in \mathcal{P}$, $V_M(h^*,p)$ is a binary vector and $\{V_M(h^*,p)\} \neq \emptyset$. Hence, there must exist at least one state $v \in \mathcal{Q}$ such that $V_M(h^*,p) = v$ for some $p \in \mathcal{P}$, which means $(h^*,v) \in \mathfrak{R}$. Due to the fact that h^* is chosen arbitrarily, we can state that for all $h \in \mathcal{H}$, there exists $v \in \mathcal{Q}$ and $(h,v) \in \mathfrak{R}$.

To prove the first implication, note that $(h,v) \in \mathfrak{R}$ means by definition that for some p' in $s(h)$, the valuation map at $V_M(h,p') = v$. For generality, assume that $p' \neq p$. Since $h \overset{k[p]}{\dashrightarrow} h'$, p must also be in $s(h)$ and $(h,p) \models \overleftarrow{k}$. Therefore we trivially have $(h,p',k) \longrightarrow (h,p,k)$,[4] and thus by Definition 9 there exists a silent transition $v \xrightarrow{\tau_k}_T v''$, where $v'' = V_M(h,p)$. The existence of the evolution $h \overset{k[p]}{\dashrightarrow} h'$, also suggests that $h \in s^{-1}(p)$ and $h' \in s^{-1}(p)$. Let $V_M(h',p) = v'$. With $V_M(h,p) = v''$ and $V_M(h',p) = v'$ by Definition 9 there must be a transition $v'' \xrightarrow{k}_T v'$. We thus have a composite transition $v \xrightarrow{\tau_k}_T v'' \xrightarrow{k}_T v'$, which means that $v \overset{k}{\rightsquigarrow} v'$ with $V_M(h',p) = v' \Leftrightarrow (h',v') \in \mathfrak{R}$.

To establish the second implication, observe that if $v \overset{k}{\rightsquigarrow} v'$, then there must be $q, q' \in \mathcal{Q}$ such that $q \xrightarrow{k}_T q'$. Given that $(v,h) \in \mathfrak{R}$ and that v jumps to q via a series of silent transitions (in which the hybrid state h is preserved), we can invoke Definition 9 to ensure the existence of an evolution $h \overset{k[p]}{\dashrightarrow} h'$ for some $p \in s(h) \cap s(h')$, such that $V_M(h,p) = q$ and $V_M(h',p) = q'$. It remains to show that $(h',v') \in \mathfrak{R}$. By the same token, q' jumps to v' by another series of silent transitions, in which the hybrid state remains at h'. By Definition 9 therefore, just as we have $p \in s(h')$ and $V_M(h',p) = q'$ there should also be a $p' \in s(h')$ such that $V_M(h',p') = v'$, which shows that $(h',v') \in \mathfrak{R}$.

What Theorem 1 implies, is that a controller switching sequence given as a succession of Δ transitions —once interlaced with the corresponding silent transitions that "prepare" the activation of these controllers— yields an admissible input trace in the transition system. Conversely, an acceptable input trace in the transition system, once projected on the set of controller symbols (removing the silent transitions) gives a controller sequence for the hybrid system that is implementable. This type of equivalence is of interest because one can in theory obtain all possible controller sequences in the hybrid agent without actually simulating it directly; these same controller sequences can be obtained as traces of the finite transition system.

4 Case study: fetching a printout

To illustrate the function of the different components of the concrete system and its abstraction, this section traces the execution of a simple task by a wheeled mobile

[4] These type of transitions in **H** correspond to an on-line re-parameterization of the controller already activated.

manipulator, the kinematics of which are given by

$$\underbrace{\begin{aligned}\dot{x} &= v\cos\theta\\ \dot{y} &= v\sin\theta\\ \dot{\theta} &= \omega\end{aligned}}_{\text{base}} \qquad \underbrace{\begin{aligned}\dot{x}_e &= u_1\\ \dot{y}_e &= u_2\\ \dot{z}_e &= u_3\end{aligned}}_{\text{arm}} \quad , \tag{1}$$

in both concrete and abstract representations of the hybrid system describing the behavior of the mobile manipulator. In (1), $(x, y) = q_p$ denotes the planar position of the robot's base, θ is the orientation of the base, and $(x_e, y_e, z_e) = q_m$ are the cartesian coordinates of the onboard arm, relative to some base-fixed coordinate system. The mapping from the base-fixed coordinate system to the global coordinate system is assumed to be expressed by $C(\theta)q_m + (q_p, 0)$, where C is a rotation matrix (the dependence on the current base orientation θ is dropped for brevity, but assumed).

The robot, through the choice of the control inputs v, ω and u_1, u_2, u_3, can be controlled to either reposition its mobile base from configuration A to configuration B within a planar environment, or pick up a small object with its arm while its base is stationary, or place an object held at a specific mode while its base remains stationary. The three different controllers that give rise to each one of the aforementioned behaviors will be represented by three symbols a, b, and c, respectively. The exact expression of these control laws is immaterial; what is known is that all these control laws are guaranteed to achieve their objective, namely reach the target configuration B from every initial configuration A, pick up an object from mode q_o and hold it at some position $q_m = p_m^h$ relative to the base, and place an object held at some other location, respectively.

Let $\mathcal{W}(q_p)$ denote the physical reachable workspace of the onboard arm's gripper, when the mobile base is at location q_p, and denote q_o the initial cartesian coordinates of the object of interest in some global coordinate frame. The task that the robot is called to complete in this example is to fetch a printout: navigate and park in front of the printer at $p_n \in \mathbb{R}^2 \times \mathsf{S}^1$,[5] pick up a stack of papers at the printer's output tray at location $q_o \in \mathcal{W}(p_n) \subset \mathbb{R}^3$, and deliver the stack to the user at location $q_u \in \mathcal{W}(p_u) \subset \mathbb{R}^3$ for some $p_u \in \mathbb{R}^2 \times \mathsf{S}^1$.

The robot, equipped with the three controllers labeled a, b, and c, can be described as a hybrid agent $\mathbf{H} = \{\mathcal{H}, \mathcal{P}, \mathcal{K}, \mathcal{I}, \mathcal{AP}, f, \overleftarrow{\cdot}, \overrightarrow{\cdot}, s, \Delta\}$. The components of the hybrid agent are described in detail as follows.

$\mathcal{H} = X \times \mathcal{L}$	is the hybrid state space, where $X \subset \mathbb{R}^2 \times \mathsf{S}^1 \times \mathbb{R}^3$, and $\mathcal{L} = \{\mathfrak{g}\}$ consists of a single boolean variable $\mathfrak{g}$, that expresses whether the gripper of the robot arm holds something (true) or not (false).
$\mathcal{P} \subset \mathbb{R}^6$	is the range of the parameter vector $p = (p_p, p_o)$, where p_p is used to specify a desired (x, y, θ) configuration for the base, and p_o a desired configuration for the arm.
$\mathcal{K} = \{a, b, c\}$	is the set of discrete modes, in which the system may operate, one for each control law. In mode a the robot base moves from q_p^i to q_p^d. In mode b the arm picks up an object at p_o and holds it at p_m^h. In mode c the arm places the object held, at location p_o, and returns at p_m^h.

[5] The set S^n denotes the surface of an n-dimensional sphere.

$\mathcal{I} = \mathcal{K} \times \mathcal{P}$ is the set containing all possible inputs.

$\mathcal{AP} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ is a set of four atomic propositions, defined as follows: $\alpha_1 \Leftrightarrow q_p = p_p$, $\alpha_2 \Leftrightarrow q_o = C\,p_o + (q_p, 0)$, $\alpha_3 \Leftrightarrow p_o \in \mathcal{W}(q_p)$, and $\alpha_4 \Leftrightarrow \mathfrak{g}$.

f is the vector field of the system, given by (1).

$\overleftarrow{\cdot}$, $\overrightarrow{\cdot}$ the PRE and POST conditions for each controller:

	a	b	c
PRE	$\{\neg\alpha_1\}$	$\{\alpha_2, \alpha_3, \neg\alpha_4\}$	$\{\neg\alpha_2, \alpha_3, \alpha_4\}$
POST	$\{\alpha_1\}$	$\{\neg\alpha_2, \alpha_3, \alpha_4\}$	$\{\alpha_2, \alpha_3, \neg\alpha_4\}$

The sets of atomic propositions are interpreted as conjunctions, *i.e.*, $\{\alpha_2, \alpha_3, \neg\alpha_4\} \Leftrightarrow \alpha_2 \wedge \alpha_3 \wedge (\neg\alpha_4)$.

$s : \mathcal{H} \to 2^{\mathcal{P}}$ is the reset map in which $(q_p, q_m) \mapsto \mathbb{R}^2 \times \mathsf{S}^1 \times \mathcal{W}(q_p)$.

Δ follows generically its description in Definition 6.

Note that in this model there can be no transitions of the form $(h, p, b) \to (h, p', b)$ or $(h, p, c) \to (h, p', c)$, because irrespectively of the choice of p', the PRE and POST of both b and c are incompatible with each other (α_4 does not depend on parameters). This reflects the fact that once the gripper holds something it cannot pick up something new, and if it has just placed the object somewhere it is not possible to place the same object somewhere else without picking it up first.

The valuation map in this case becomes a mapping from a thirteenth-dimensional space, $\mathcal{H} \times \mathcal{P} \subset \mathbb{R}^2 \times \mathsf{S}^1 \times \{\mathbf{0}, \mathbf{1}\} \times \mathbb{R}^6$ into the vertices of a fourth dimensional (unit) hypercube. These vertices are labeled with a binary sequence indicating which atomic propositions α_i are true, and are shown tabulated in a two-dimensional array in Fig. 1(b). Based on the valuation map, the induced labeled transition system $\mathbf{T}(\mathbf{H})$ can be constructed with its components defined as follows.

$\mathcal{Q} = \{q_1 q_2 q_3 q_4 : q_i \in \{0, 1\},\ i = 1, \ldots, 4\}$ is the set of discrete states.

$\Sigma = \{a, b, c, \tau_a, \tau_b, \tau_c\}$ is the set of labels.

δ is the transition relation of Definition 9.

A sequence of inputs in $\mathbf{H}$ enables it to complete the task:

$a[p^1]$ with $p^1 = (p_n, p_m^h)$,

$b[p^2]$ with $p^2 = \Big(p_n, C^{-1}(q_o - (q_p, 0))\Big)$,

$a[p^3]$ with $p^3 = (p_u, q_u)$, and

$c[p^3]$ with no parameter reset (identity).

The sequence of inputs $a[p^1]\,b[p^2]\,a[p^3]\,c[p^3]$ for $\mathbf{H}$, translates under ϕ to a sequence of labels $\tau_a\,a\,\tau_b\,b\,\tau_a\,a\,\tau_c\,c$ in $\mathbf{T}(\mathbf{H})$, where the transitions labeled with τ_i in $\mathbf{T}(\mathbf{H})$ are considered silent. The remaining of the section illustrates the parallel runs on $\mathbf{H}$ and $\mathbf{T}(\mathbf{H})$, and shows what is the effect of the discrete abstraction going from the hybrid to the purely discrete model.

Assume that the initial parameter assignment in $\mathbf{H}$ is some $p^i = (p_p^i, p_m^i)$, and that the system starts at rest with $(q_p, \theta) = (x_0, y_0, \theta_0) = p_p^i \neq p_n \triangleq (0, 0, 0)$, $q_m = p_m^h = p_m^i$, with $\mathfrak{g} = \mathbf{0}$, and $C\,p_m^i + (q_p, 0) \neq q_o$. The valuation map gives for $h^i = (p_p^i, p_m^h, \mathbf{0})$ and p^i the value $v_i = 1010$, and the system satisfies POST(a) (see Fig. 1(b)). The first input is $a[p^1]$, which involves reseting p^i to $p^1 = (p_p^1, p_o^1) = (p_n, p_m^h)$, in accordance to the reset map s. With this assignment the valuation map now reports for (h^i, p^1) the

value $v_1 = 0010$ which means that $(h^i, p^1) \models \textsc{Pre}(a)$. The semantics of Δ then suggest that $\mathbf{H}$ can make a transition $(h^i, p^i, a) \rightarrow (h^i, p^1, a)$, and that h^i starts evolving to $h^1 = (p^1, \mathbf{0})$; we write $h^i \overset{a[p^1]}{\dashrightarrow} h^1$. When the hybrid state of $\mathbf{H}$ becomes h^1, the valuation map reports $V_M(h^1, p^1) = 1000$ again; the difference now is that $(q_p, \theta) = p_n$.

Note that $\big((h^i, p^i, a), (h^i, p^1, a)\big) \in \Delta \Rightarrow 1010 \xrightarrow{\tau_a}_T 0010$, and that $(h^i, p^1) \models \overleftarrow{a}$ with $h^i \overset{a[p^1]}{\dashrightarrow} h^1$ imply $0010 \xrightarrow{a}_T 1010$; Thus $\mathbf{T}(\mathbf{H})$ indeed matches the evolution from h^i to h^1 in $\mathbf{H}$.

Now $(h^1, p^1) \models \textsc{Post}(a)$. At that point the second input is processed, which is $b[p^2]$, reseting the parameter to $p^2 = (p_p^2, p_o^2) = \Big(p_n, C^{-1}(q_o - (q_p, 0))\Big)$. The switch makes V_M jump to 1110 because $q_o = C\, p_o^2 + (q_p, 0) \Leftrightarrow (h^1, p^2) \models \alpha_2$ and in addition $p_o^2 \in \mathcal{W}(p_n)$. Since $(h^1, p^2) \models \textsc{Pre}(b)$, Δ permits $\mathbf{H}$ to take the transition $(h^1, p^1, a) \rightarrow (h^1, p^2, b)$; for the same reason there exists a transition $1010 \xrightarrow{\tau_b}_T 1110$ in $\mathbf{T}(\mathbf{H})$. With $\mathbf{H}$ at mode b, the continuous dynamics first evolve to a state where $q_m = C^{-1}\big(q_o - (q_p, 0)\big)$, before retreating to $q_m = p_m^h$ with the gripper holding the object originally positioned at q_o. When this happens, the valuation map reads 1011, because the object is no longer at p_o^2 and thus α_2 is now false. Now $\mathbf{H}$ is in $\textsc{Post}(b)$ (see Fig. 1(b)), and $h^2 = (p_n, \mathbf{1})$. Since $h^1 = (p_n, \mathbf{0}) \overset{b[p^1]}{\dashrightarrow} (p_n, \mathbf{1}) = h^2$, with $(h^1, p^1) \models \overleftarrow{b}$ and $V_M(h^2, p^1) = 1011$, there exists a transition $1110 \xrightarrow{b}_T 1011$ in $\mathbf{T}(\mathbf{H})$.

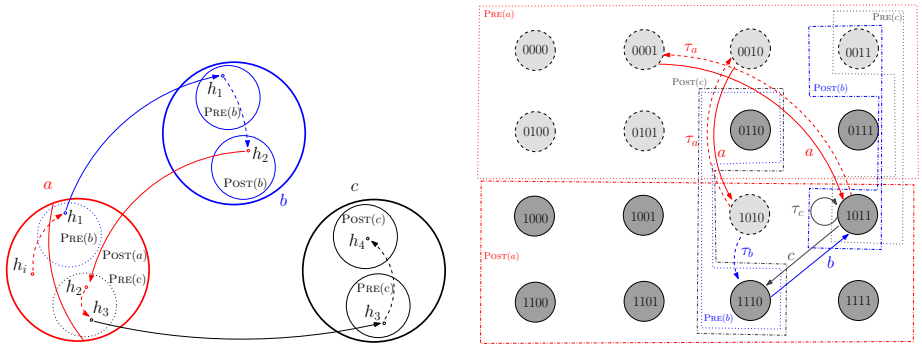
Now the input $a[p^3]$ is processed, where the parameter vector is reset to $p^3 = (p_p^3, p_o^3) = (p_u, q_u)$. This assignment satisfies $(h^2, p^3) \models \textsc{Pre}(a)$ and gives $V_M(h^2, p^3) = 0001$.[6] The transition relation in $\mathbf{H}$ then enables $(h^2, p^2, b) \rightarrow (h^2, p^3, a)$ and the continuous dynamics start to evolve in mode a in a way so that $(h^2, a) \overset{a[p^3]}{\dashrightarrow} (h^3, a)$, where $h^3 = (p_u, p_m^h, \mathbf{1})$. When the hybrid state reaches h^3, then $V_M(h^3, p^3) = 1011$, because in addition to $q_p = p_u$ it is now $q_u \in \mathcal{W}(p_u)$. Since $\big((h^2, p^2, b), (h^2, p^3, a)\big) \in \Delta$, there exists a transition in $\mathbf{T}(\mathbf{H})$ of the form $1011 \xrightarrow{\tau_a}_T 0001$; similarly from $(h^2, a) \overset{a[p^3]}{\dashrightarrow} (h^3, a)$ it follows that there is also a δ transition $0001 \xrightarrow{a}_T 1011$.

Notice now that $(h^3, p^3) \models \textsc{Pre}(c)$. Thus Δ trivially admits $(h^3, p^3, a) \rightarrow (h^3, p^3, c)$ without any additional change in the parameters. At the same time, by definition since $\big((h^3, p^3, a), (h^3, p^3, c)\big) \in \Delta$ it follows that $1011 \xrightarrow{\tau_c}_T 1011 \in \delta$. Now $\mathbf{H}$ is able to process input $c[p^3]$. With that, the robot's gripper reaches out to q_u, releases the printout and returns to $q_m = p_m^h$. With the object at q_u, α_2 becomes true and the valuation map outputs 1110. For $h^4 = (p_u, p_m^h, \mathbf{0})$ we write $h^3 \overset{c[p^3]}{\dashrightarrow} h^4$, which also implies the existence of a δ transition $1011 \xrightarrow{c}_T 1110$.

Thus the evolution of a hybrid system of the form $\mathbf{H}$, is matched by a run in the labeled finite transition system $\mathbf{T}(\mathbf{H})$, through interlacing of observable and unobservable (silent) transitions. The execution of the plan to fetch the printout on the hybrid agent $\mathbf{H}_r$ and its abstraction on the transition system $\mathbf{T}_r$ are shown in Figure 1. Had an additional atomic proposition $\alpha_5 \Leftrightarrow q_o = q_u$ been defined, its satisfaction would signify the completion of the task. For simplicity in the representation of $\mathbf{T}(\mathbf{H})$ in Fig. 1(b), this predicate is not included.

In general, if there is an execution in $\mathbf{H}$, there is always a path in $\mathbf{T}(\mathbf{H})$ that visits the same blocks of the partition induced by V_M (and vice versa); the transition

[6] We assume that the user is not right next to the printer and so $q_u \notin \mathcal{W}(p_n)$, because then there would be no need to send the robot to bring the printout.



(a) Evolution of the hybrid dynamics. (b) Evolution of the discrete dynamics.

Fig. 1 The hybrid agent and its abstraction. Continuous evolution and discrete jumps in the hybrid system are mirrored in the silent and observable transitions of the transition system. Each state in the transition system defines a region on the continuous domain where a specific combination of atomic propositions evaluates true.

sequences in the two machines, however, need not have the same length. In theory, this property of the two models enables one to search in $\mathbf{T}(\mathbf{H})$ for a path that connects an arbitrary initial state to a "forbidden" block. If such a path exists in $\mathbf{T}(\mathbf{H})$ this means that there is a combination of inputs and parameters in $\mathbf{H}$ that can realize it.

5 Conclusions

Hybrid systems within a certain class can be abstracted into finite state transition systems. A finite transition system obtained as an abstraction, is shown to be observably bisimilar to the concrete hybrid dynamics it originated from. This fact ensures that all input strings that the transition system accepts, have a corresponding implementation on the concrete hybrid system, and that whatever behavior the hybrid system can exhibit is also observed as a sequence of transitions between the equivalence classes of the quotient abstract system. The result allows motion planning and behavior design for the hybrid system to be performed on the discrete system, without concerns about the continuous dynamics of the former. In addition, if a property is found to hold for the transition system, it will also hold for the hybrid system. Ongoing work is directed toward using slightly more general discrete models of computation, which will enable the designer to preserve the information about the continuous controller parameterization in the abstraction, and allow planning at the discrete level with the use of this information.

References

1. Gulwani, S., Tiwari, A.: Constraint-based approach for analysis of hybrid systems. In: A. Gupta, S. Malik (eds.) Computer Aided Verification, *LNCS*, vol. 5123, pp. 190–203. Springer-Verlag (2008)
2. Platzer, A., Clarke, E.M.: Computing differential invariants of hybrid systems as fixedpoints. Formal Methods in System Design **35**(1), 98–120 (2009)

3. Tomlin, C., Mitchell, I., Bayen, A., Oishi, M.: Computational techniques for the verification of hybrid systems. Proceedings of the IEEE **91**(7), 986–1001 (2000)
4. Kurzhanski, A., Varaiya, P.: Ellipsoidal techniques for reachability analysis. In: N. Lynch, B. Krogh (eds.) Hybrid Systems : Computation and Control, *LNCS*, vol. 1790, pp. 310–323. Springer-Verlag (2000)
5. Stursberg, O., Krogh, B.: Efficient representation and computation of reachable sets for hybrid systems. In: O. Lynch, A. Pnueli (eds.) Hybrid Systems : Computation and Control, *LNCS*, vol. 2623, pp. 482–497. Springer-Verlag (2003)
6. Girard, A., Cuernic, C.: Zonotope/hyperplane intersection for bybrid systems reachability analysis. In: M. Egerstedt, P. Mishra (eds.) Hybrid Systems : Computation and Control, *LNCS*, vol. 4981, pp. 215–228. Springer-Verlag (2008)
7. Tiwari, A.: Abstractions for hybrid systems. Formal Methods in System Design **32**(1), 57–83 (2008)
8. Lerda, F., Kapinski, J., Clarke, E., Krogh, B.: Verification of supervisory control software using state proximity and merging. In: M. Egerstedt, B. Mishra (eds.) Hybrid Systems : Computation and Control, *LNCS*, vol. 4981, pp. 344–357. Springer-Verlag (2008)
9. Broucke, M.E.: A geometric approach to bisimulation and verification of hybrid systems. In: F.W. Vaandrager, J.H. van Schuppen (eds.) Hybrid Systems: Computation and Control, *Lecture Notes in Computer Science*, vol. 1569, pp. 61–75. Springer-Verlag (1999)
10. Girard, A., Pappas, G.J.: Hierarchical control system design using approximate simulation. Automatica **45**, 566–571 (2009)
11. Tabuada, P.: Approximate simulation relations and finite abstractions of quantized control systems. In: A. Bemporad, A. Bicchi, G. Buttazzo (eds.) Hybrid Systems: Computation and Control, *Lecture Notes in Computer Science*, vol. 4416, pp. 529–542. Springer-Verlag (2007)
12. Alur, R., Henzinger, T., Lafferriere, G., Pappas, G.: Discrete abstractions of hybrid systems. Proceedings of the IEEE **88**(7), 971–984 (2000)
13. Milner, R.: Communication and Concurrency. Prentice Hall (1989)
14. Girard, A., Pappas, G.J.: Approximate metrics for discrete and continuous systems. IEEE Transactions on Automatic Control **53**(5), 782–798 (2007)
15. Girard, A., Pola, G., Tabuada, P.: Approximately bisimilar symbolic models for incrementally stable switched systems. In: M. Egerstedt, B. Mishra (eds.) Hybrid Systems: Computation and Control, *Lecture Notes in Computer Science*, vol. 4981, pp. 201–214. Springer Verlag (2008)
16. Alur, R., Verimag, T.D., Ivancic, F.: Predicate abstractions for reachability analysis of hybrid systems. ACM Transactions on Embedded Computing Systems **5**(1), 152–199 (2006)
17. Koutsoukos, X., Antsaklis, P., Stiver, J., Lemmon, M.: Supervisory control of hybrid systems. Proceedings of the IEEE **88**(7), 1026–1049 (2000)
18. Lunze, J., B.Nixdorf, Schroder, J.: Deterministic discrete-event representations of linear continuous-variable systems. Automatica **35**(3), 395–406 (1999)
19. Raisch, J., O'Young, S.: Discrete approximations and supervisory control of continuous systems. IEEE Transactions on Automatic Control **43**(4), 569–573 (1998)
20. Tazaki, Y., Imura, J.: Finite abstractions of discrete-time linear systems and its application to optimal control. In: Proceedings of the 17th IFAC World Congress, pp. 4656–4661 (2008)
21. Kowalewski, S., Engell, S., Preußig, J., Stursberg, O.: Verification of logic controlles for continuous plants using timed condition/event-system models. Automatica **35**, 505–518 (1999)
22. Lygeros, J., Johansson, K., Simić, S., Sastry, S.: Dynamical properties of hybrid automata. IEEE Transactions on Automatic Control **48**(1), 2–17 (2003)
23. Bullo, F., Lewis, A.D.: Geometric Control of Mechanical Systems. No. 49 in Texts in Applied Mathematics. Springer (2005)
24. Khalil, H.K.: Nonlinear Systems. Prentice Hall (1996)
25. Piovesan, J.L., Tanner, H.G., Abdallah, C.T.: Discrete asymptotic abstractions of hybrid systems. In: Proceedings of 45th IEEE Conference on Decision & Control, pp. 917–922 (2006)
26. Athanasopoulos, K.: Explosions near isolated unstable attractors. Pacific Journal of Mathematics **210**(2), 201–214 (2003)
27. Milner, R.: Communicating and mobile systems: the π calculus. Cambridge University Press (2003)
28. Stirling, C.: Modal and temporal logics for processes. In: F. Moller, G. Birtwistle (eds.) Logics for concurency: structure vs automata. Springer (1996)