An integrated approach to the symbolic control design of nonlinear systems with infinite states specifications

Alessandro Borri, Giordano Pola, and Maria D. Di Benedetto

***Abstract*— In this paper we address the problem of symbolic control design of nonlinear control systems with infinite states specifications, modelled by differential equations. An algorithm for the design of symbolic controllers is presented, which integrates the construction of the discrete abstractions of the plant and of the specification with the design of the controller. This integrated algorithm reduces the space complexity of the control design computations, as formally discussed in the paper and further illustrated through an illustrative example.**

I. Introduction

Discrete abstractions of continuous and hybrid systems have been the topic of intensive study in the last twenty years from both the control systems and the computer science communities [1]. While physical world processes are often described by differential equations, digital controllers and software/hardware at the implementation layer are usually modelled through discrete/symbolic processes. This mathematical models heterogeneity has posed during the years interesting and challenging theoretical problems that need to be addressed, in order to ensure the formal correctness of control algorithms, in the presence of non–idealities at the implementation layer. One approach to deal with this heterogeneity is to construct symbolic models that are equivalent to the continuous process, so that the mathematical model of the process, of the controller, and of the software and hardware at the implementation layer are of the same nature. Several classes of dynamical and control systems admitting symbolic models were identified during the years. We recall timed automata [2], rectangular hybrid automata [3] and o-minimal hybrid systems [4] in the class of hybrid automata. Control systems were considered further: controllable discrete–time linear systems [5], piecewise-affine systems [6] and multi-affine systems [7]. Many of the aforementioned work are based on the notion of bisimulation equivalence, introduced by Milner and Park [8], [9] in the context of concurrent processes, as a formal equivalence notion to relate continuous and hybrid processes to purely discrete/symbolic models. A new insight in the construction of symbolic models has been recently placed through the notion of approximate bisimulation introduced by Girard and Pappas in [10]. Based on the above notion, some classes of incrementally stable [11] control systems were recently shown to admit symbolic models: we recall stable discrete–time linear control systems [12], nonlinear control systems with and without disturbances [13], [14], nonlinear time–delay systems [15] and switched nonlinear systems [16]. The use of symbolic models in the control design of continuous and hybrid systems has been investigated in the work of [5], [17], [18], among many others. While the work in [5] considers discrete–time linear control systems, the work in [17] considers piecewise affine hybrid systems and finally, the work in [18] considers stabilizable nonlinear control systems. In this paper, we give a further contribution to this research line and in particular, in the direction of the work in [18]. We consider symbolic control design of nonlinear control systems with infinite states specifications, expressed through differential equations: given a plant nonlinear control system and a specification autonomous nonlinear system, we study conditions for the existence of a symbolic controller that implements the behaviour of the specification which can be implemented from the plant, with a precision that can be rendered as small as desired. The symbolic controller is furthermore requested to be non–blocking in order to prevent deadlocks in the interaction between the plant and the symbolic controller. Such control design problem has been solved, being inspired from the so–called *correct–by–design* approach, see e.g. [18], [5]. While being formally correct from the theoretical point of view, this approach, as similar approaches currently available in the literature (see e.g. [5], [17], [18]), results in general in being rather demanding from the computational point of view, because of the large size of the symbolic models needed to be constructed in order to synthesize the symbolic controller. Inspired from the research line in the context of on–the–fly verification and control of timed or untimed transition systems (see e.g. [19], [20]), we approach the design of such controller by means of an "integration" philosophy: instead of computing separately the symbolic models of the plant and of the specification to then synthesize the controller at the symbolic layer, *we integrate each step of this procedure in only one algorithm*. This integrated algorithm reduces the space complexity of the control design computations, as formally discussed in the paper and further illustrated through an illustrative example. For the sake of completeness, a detailed list of the employed notation is included in the Appendix (Section VII).

This work has been partially supported by the Center of Excellence for Research DEWS, University of L'Aquila, Italy.

A. Borri, G. Pola, and M. D. Di Benedetto are with the Department of Electrical and Information Engineering, Center of Excellence DEWS, University of L'Aquila, Poggio di Roio, 67040 L'Aquila, Italy, {giordano.pola,alessandro.borri,mariadomenica.dibenedetto}@univaq.it.

II. Preliminary Definitions

A. Control Systems

The class of control systems that we consider in this paper is formalized in the following definition.

Definition 1: A *control system* is a quintuple:

$$\Sigma = (X, X_0, U, \mathcal{U}, f), \tag{1}$$

where:

- $X \subseteq \mathbb{R}^n$ is the state space;
- $X_0 \subseteq X$ is the set of initial states;
- $U \subseteq \mathbb{R}^m$ is the input space;
- $\mathcal{U}$ is a subset of the set of all locally essentially bounded functions of time from intervals of the form $]a, b[\subseteq \mathbb{R}$ to U with $a < 0$, $b > 0$;
- $f : \mathbb{R}^n \times U \to \mathbb{R}^n$ is a continuous map satisfying the following Lipschitz assumption: for every compact set $K \subset \mathbb{R}^n$, there exists a constant $Z > 0$ such that $\|f(x,u) - f(y,u)\| \leq Z\|x - y\|$ for all $x, y \in K$ and all $u \in U$.

A curve $\xi :]a, b[\to \mathbb{R}^n$ is said to be a *trajectory* of Σ if there exists $u \in \mathcal{U}$ satisfying $\dot{\xi}(t) = f(\xi(t), u(t))$ for almost all $t \in]a, b[$. Although we have defined trajectories over open domains, we shall refer to trajectories $\xi : [0, \tau] \to \mathbb{R}^n$ defined on closed domains $[0, \tau]$, $\tau \in \mathbb{R}^+$ with the understanding of the existence of a trajectory $\xi' :]a, b[\to \mathbb{R}^n$ such that $\xi = \xi'|_{[0,\tau]}$. We also write $\xi_{xu}(\tau)$ to denote the point reached at time τ under the input u from initial condition x; this point is uniquely determined, since the assumptions on f ensure existence and uniqueness of trajectories. The above formulation of control systems can be also used to model autonomous nonlinear systems. With a slight abuse of notation, we denote an autonomous system Σ by means of the tuple (X, X_0, f). A control system Σ is said to be forward complete if every trajectory is defined on an interval of the form $]a, \infty[$. Sufficient and necessary conditions for a system to be forward complete can be found in [21].

B. Systems

We use systems to describe both control systems as well as their symbolic models. For a detailed exposition of systems and their properties we refer to [22].

Definition 2: [22] A system S is a sextuple:

$$S = (X, X_0, U, \longrightarrow, Y, H),$$

consisting of:

- a set of states X;
- a set of initial states $X_0 \subseteq X$;
- a set of inputs U;
- a transition relation $\longrightarrow \subseteq X \times U \times X$;
- an output set Y;
- an output function $H : X \to Y$.

A transition $(x, u, x') \in \longrightarrow$ of system S is denoted by $x \xrightarrow{u} x'$. System S is said to be *countable*, if X and U are countable sets; *symbolic*, if X and U are finite sets; *metric*, if the output set Y is equipped with a metric $d : Y \times Y \to \mathbb{R}_0^+$; *deterministic*, if for any $x \in X$ and $u \in U$ there exists at most one $x' \in X$ such that $(x, u, x') \in \longrightarrow$; *non–blocking*, if for any $x \in X$ there exists $(x, u, x') \in \longrightarrow$; *accessible*, if for any $x \in X$ there exists a finite number of transitions $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \ldots \xrightarrow{u_N} x$ starting from an initial state x_0 in X_0 and ending up in x.

We now introduce some notions which are employed in the further developments.

Definition 3: Given two systems $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow[1]{}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow[2]{}, Y_2, H_2)$, system S_1 is a sub–system of S_2 if $X_1 \subseteq X_2$, $X_{0,1} \subseteq X_{0,2}$, $U_1 \subseteq U_2$, $\xrightarrow[1]{} \subseteq \xrightarrow[2]{}$, $Y_1 \subseteq Y_2$ and $H_1(x) = H_2(x)$ for any $x \in X_1$.

Definition 4: Given a system $S = (X, X_0, U, \longrightarrow, Y, H)$ the non–blocking part of S is a system $Nb(S)$ so that:

(i) $Nb(S)$ is a non–blocking system;
(ii) $Nb(S)$ is a sub–system of S;
(iii) For any non–blocking sub–system S' of S, S' is a sub–system of $Nb(S)$.

Definition 5: Given a system $S = (X, X_0, U, \longrightarrow, Y, H)$ the accessible part of S is a system $Ac(S)$ so that:

(i) $Ac(S)$ is an accessible system;
(ii) $Ac(S)$ is a sub–system of S;
(iii) For any accessible sub–system S' of S, S' is a sub–system of $Ac(S)$.

In this paper we consider simulation and bisimulation relations [8], [9] that are useful when analyzing or synthesizing controllers for deterministic systems [22]. Intuitively, a bisimulation relation between a pair of systems S_1 and S_2 is a relation between the corresponding state sets explaining how a state trajectory s_1 of S_1 can be transformed into a state trajectory s_2 of S_2 and vice versa. While typical bisimulation relations require that s_1 and s_2 are observationally indistinguishable, that is $H_1(s_1) = H_2(s_2)$, we shall relax this by requiring $H_1(s_1)$ to simply be close to $H_2(s_2)$, where closeness is measured with respect to the metric on the output set. A simulation relation is a one-sided version of a bisimulation relation. The following notions have been introduced in [10] and, in a slightly different formulation, in [18].

Definition 6: Let $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow[1]{}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow[2]{}, Y_2, H_2)$ be metric systems with the same output sets $Y_1 = Y_2$ and metric d, and consider a precision $\varepsilon \in \mathbb{R}_0^+$. A relation $\mathcal{R} \subseteq X_1 \times X_2$ is said to be an ε-approximate simulation relation from S_1 to S_2, if the following conditions are satisfied:

(i) for every $x_1 \in X_{0,1}$, there exists $x_2 \in X_{0,2}$ with $(x_1, x_2) \in \mathcal{R}$;
(ii) for every $(x_1, x_2) \in \mathcal{R}$ we have $d(H_1(x_1), H_2(x_2)) \leq \varepsilon$;
(iii) for every $(x_1, x_2) \in \mathcal{R}$ we have that: $x_1 \xrightarrow[1]{u_1} x_1'$ in S_1 implies the existence of $x_2 \xrightarrow[2]{u_2} x_2'$ in S_2 satisfying $(x_1', x_2') \in \mathcal{R}$.

System S_1 is ε-approximately simulated by S_2 or S_2 ε-approximately simulates S_1, denoted by $S_1 \preceq_\varepsilon S_2$, if there exists an ε-approximate simulation relation from S_1 to S_2. When $\varepsilon = 0$, system S_1 is said to be 0–simulated by S_2 or S_2 is said to 0–simulate S_1.

By symmetrizing the notion of approximate simulation, we obtain the notion of approximate bisimulation.

Definition 7: Let $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow[1]{}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow[2]{}, Y_2, H_2)$ be metric systems with the same output sets $Y_1 = Y_2$ and metric d, and consider a precision $\varepsilon \in \mathbb{R}_0^+$. A relation $\mathcal{R} \subseteq X_1 \times X_2$ is said to be an ε-approximate bisimulation relation between S_1 and S_2 if the following conditions are satisfied:

(i) $\mathcal{R}$ is an ε-approximate simulation relation from S_1 to S_2;

(ii) $\mathcal{R}^{-1}$ is an ε-approximate simulation relation from S_2 to S_1.

System S_1 is ε-approximate bisimilar to S_2, denoted by $S_1 \cong_\varepsilon S_2$, if there exists an ε-approximate bisimulation relation $\mathcal{R}$ between S_1 and S_2. When $\varepsilon = 0$, system S_1 is said to be 0–bisimilar or exactly bisimilar to S_2.

We now introduce the notion of approximate composition of systems which is employed to formalize the interconnection between a nonlinear control system representing the plant and a symbolic system representing the symbolic controller.

Definition 8: [18] Given two metric systems $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow[1]{}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow[2]{}, Y_2, H_2)$, with the same output sets $Y_1 = Y_2$ and metric d, and a precision $\varepsilon \in \mathbb{R}_0^+$, the ε–approximate composition of S_1 and S_2 is the system $S_1 \parallel_\varepsilon S_2 := (X, X_0, U, \longrightarrow, Y, H)$, where:

- $X = \{(x_1, x_2) \in X_1 \times X_2 : d(H_1(x_1), H_2(x_2)) \leq \varepsilon\}$;
- $X_0 = X \cap (X_{0,1} \times X_{0,2})$;
- $U = U_1 \times U_2$;
- $(x_1, x_2) \xrightarrow{(u_1,u_2)} (x_1', x_2')$ if $x_1 \xrightarrow[1]{u_1} x_1'$ and $x_2 \xrightarrow[2]{u_2} x_2'$;
- $Y = Y_1$;
- $H : X_1 \times X_2 \to Y$ is given by $H(x_1, x_2) := H_1(x_1)$, for any $(x_1, x_2) \in X$.

The above notion of composition is asymmetric. This is because it models the interaction of systems S_1 and S_2 which play different roles in the composition. As it will be clarified in the next section, we interpret system S_1 as the plant system, i.e. the system to be controlled, and system S_2 as the controller.

III. Problem Statement

Given a control system $\Sigma = (X, X_0, U, \mathcal{U}, f)$ and a sampling time parameter $\tau \in \mathbb{R}^+$, we associate the following system to Σ:

$$S_\tau(\Sigma) := (X, X_0, \mathcal{U}_\tau, \xrightarrow[\tau]{}, Y, H),$$

where:

- $\mathcal{U}_\tau = \{u \in \mathcal{U} |$ the domain of u is $[0, \tau]\}$;
- $x \xrightarrow[\tau]{u} x'$ if there exists a trajectory $\xi : [0, \tau] \to X$ of Σ satisfying $\xi_{xu}(\tau) = x'$;
- $Y = X$;
- $H = 1_X$.

System $S_\tau(\Sigma)$ is metric when we regard $Y = X$ as being equipped with the metric $d(p, q) = \|p - q\|$. The above system can be thought of as the time discretization of the control system Σ. Given the control system Σ, a sampling time $\tau \in \mathbb{R}^+$, a state quantization $\theta \in \mathbb{R}^+$ and an input quantization $\mu \in \mathbb{R}^+$, a symbolic controller for Σ is formalized by means of the system:

$$C := (X_c, X_{c,0}, U_c, \xrightarrow[c]{}, Y_c, H_c),$$

where:

- $X_c = [X]_{2\theta}$;
- $X_{c,0} \subseteq X_c$;
- $U_c = \{u \in \mathcal{U}_\tau |$ the codomain of u is $[U]_{2\mu}\}$;
- $\xrightarrow[c]{} \subseteq X_c \times U_c \times X_c$;
- $Y_c = X_c$;
- $H_c = 1_{X_c}$.

The interpretation of such a symbolic controller is the following. When the state x of the control system Σ is in[1] $\mathcal{B}_{[\theta)}(y)$ for some $y \in X_c$ at time $t = k\tau$ for $k \in \mathbb{N}$, the symbolic controller C provides a control input $u \in U_c$ so that $y \xrightarrow[c]{u} z$ for some $z \in X_c$. The interaction between the control system Σ and the symbolic controller C is formally captured by the approximate composition $S_\tau(\Sigma) \parallel_\theta C$. We denote by $\mathcal{C}^{\tau,\theta,\mu}$ the class of symbolic controllers with sampling time τ, state quantization θ and input quantization μ.

Consider a plant nonlinear control system:

$$P = (X_p, X_{p,0}, U_p, \mathcal{U}_p, f_p), \tag{2}$$

and a specification nonlinear autonomous system $Q = (X_q, X_{q,0}, g_q)$. For the sake of homogeneity in the notation of the plant P and the specification Q, in the following we rephrase the above tuple by means of:

$$Q = (X_q, X_{q,0}, U_q, \mathcal{U}_q, f_q), \tag{3}$$

where $U_q = \{u_q\}$ with $u_q = 0, \mathcal{U}_q = \{\mathbf{u}_q\}$ with $\mathbf{u}_q = \mathbf{0}$, the signal $\mathbf{0}$ being the null function, and $f_q(x, u) = g_q(x) + u$ for any $(x, u) \in X_q \times U_q$. In this paper we consider the following symbolic control design problem:

Problem 1: Given a plant nonlinear control system P as in (2), a specification nonlinear autonomous system Q as in (3) and a desired precision $\varepsilon \in \mathbb{R}^+$, find quantization parameters $\tau, \theta, \mu \in \mathbb{R}^+$ and a symbolic controller $C \in \mathcal{C}^{\tau,\theta,\mu}$ such that:

(i) $\varnothing \neq (S_\tau(P) \parallel_\theta C) \preceq_\varepsilon S_\tau(Q)$;

(ii) $(S_\tau(P) \parallel_\theta C)$ is non–blocking.

The above controller synthesis problem asks for a symbolic controller that implements the behaviour of the specification which is implementable from the plant and which is non–blocking, up to a precision ε that can be chosen as small as desired.

IV. Symbolic Control Design for Infinite States Specification

In this section we provide the solution to Problem 1, inspired by the so–called correct–by design approach, see e.g. [18], [5]. We start by recalling from [11] the notion

[1]The set $\mathcal{B}_{[\theta)}(y)$ is defined in the Appendix.

of incremental input–to–state stability for nonlinear control systems.

Definition 9: A control system Σ is incrementally input–to–state stable (δ-ISS) if it is forward complete and there exist a $\mathcal{KL}$ function β and a $\mathcal{K}_\infty$ function γ such that for any $t \in \mathbb{R}_0^+$, any $x, x' \in \mathbb{R}^n$, and any $u, u' \in \mathcal{U}$ the following condition is satisfied:

$$\|\xi_{xu}(t) - \xi_{x'u'}(t)\| \leq \beta(\|x - x'\|, t) + \gamma(\|u - u'\|_\infty).$$

A characterization of the above incremental stability notion in terms of dissipation inequalities can be found in [11]. Given a δ–ISS nonlinear control system Σ of the form (1), a sampling time $\tau \in \mathbb{R}^+$, a state space quantization $\eta \in \mathbb{R}^+$ and an input space quantization $\mu \in \mathbb{R}^+$, consider the following system:

$$S_{\tau,\eta,\mu}(\Sigma) := (X_{\tau,\eta,\mu}, X_{0,\tau,\eta,\mu}, U_{\tau,\eta,\mu}, \underset{\tau,\eta,\mu}{\longrightarrow}, Y_{\tau,\eta,\mu}, H_{\tau,\eta,\mu}),$$

where:

- $X_{\tau,\eta,\mu} = [X]_{2\eta}$;
- $X_{0,\tau,\eta,\mu} = X_{\tau,\eta,\mu} \cap X_0$;
- $U_{\tau,\eta,\mu} = [U]_{2\mu}$;
- $x \xrightarrow[\tau,\eta,\mu]{u} y$ if $\xi_{xu}(\tau) \in \mathcal{B}_{[\eta)}(y) \cap X$;
- $Y_{\tau,\eta,\mu} = X$;
- $H_{\tau,\eta,\mu} = \imath : X_{\tau,\eta,\mu} \hookrightarrow Y_{\tau,\eta,\mu}$.

System $S_{\tau,\eta,\mu}(\Sigma)$ is countable and becomes symbolic when the state space X and the input space U are bounded. The symbolic system $S_{\tau,\eta,\mu}(\Sigma)$ is basically equivalent to the symbolic systems proposed in [13]. The main difference is that, while the symbolic systems in [13] are not guaranteed to be deterministic, system $S_{\tau,\eta,\mu}(\Sigma)$ is so, as formally stated in the following result:

Proposition 1: System $S_{\tau,\eta,\mu}(\Sigma)$ is deterministic.

Proof: The existence and uniqueness of a trajectory from an initial condition $x \in X_{\tau,\eta,\mu}$ with input $u \in U_{\tau,\eta,\mu}$ guarantees that $\xi_{xu}(\tau)$ is uniquely determined. Moreover, since the collection of sets $\{\mathcal{B}_{[\eta)}(y) \cap X\}_{y \in X_{\tau,\eta,\mu}}$ is a partition of X, there exists at most one state $y \in X_{\tau,\eta,\mu}$ such that $\xi_{xu}(\tau) \in \mathcal{B}_{[\eta)}(y) \cap X$. ∎

We can now give the following result that establishes sufficient conditions for the existence and construction of symbolic systems for nonlinear control systems.

Theorem 1: Consider a δ–ISS nonlinear control system $\Sigma = (X, X_0, U, \mathcal{U}, f)$ and a desired precision $\theta \in \mathbb{R}^+$. For any sampling time $\tau \in \mathbb{R}^+$, state space quantization $\eta \in \mathbb{R}^+$ and input quantization $\mu \in \mathbb{R}^+$ satisfying the following inequality:

$$\beta(\theta, \tau) + \gamma(\mu) + \eta \leq \theta, \tag{4}$$

systems $S_{\tau,\eta,\mu}(\Sigma)$ and $S_\tau(\Sigma)$ are θ–approximately bisimilar.

Proof: The proof of the above result can be given along the lines of Theorem 5.1 in [13]. We include it here for the sake of completeness. Consider the relation $R \subseteq X \times X_{\tau,\eta,\mu}$ defined by $(x, y) \in R$ if and only if $x \in \mathcal{B}_{[\theta)}(y) \cap X$. We start by showing that condition (i) of Definition 6 holds. Consider an initial condition $x_0 \in X_0$. By definition of the set $X_{0,\tau,\eta,\mu}$ there exists $y_0 \in X_{0,\tau,\eta,\mu}$ so that $(x_0, y_0) \in R$. Condition (ii) in Definition 6 is satisfied by the definition of R. Let us now show that condition (iii) in Definition 6 holds. Consider any $(x, y) \in R$. Consider any $u_1 \in \mathcal{U}_\tau$ and the transition $x \xrightarrow[\tau]{u_1} w$ in $S_\tau(\Sigma)$. Then there exists $u_2 \in U_{\tau,\eta,\mu}$ such that:

$$\|u_2 - u_1\|_\infty \leq \mu, \tag{5}$$

and set $z = \xi_{yu_2}(\tau)$. Since $X = \bigcup_{v \in X_{\tau,\eta,\mu}} \mathcal{B}_{[\eta)}(v) \cap X$, there exists $v \in X_{\tau,\eta,\mu}$ such that:

$$z \in \mathcal{B}_{[\eta)}(v), \tag{6}$$

and therefore $y \xrightarrow[\tau,\eta,\mu]{u_2} v$ in $S_{\tau,\eta,\mu}(\Sigma)$. Since Σ is δ–ISS, by the definition of R and by condition (5), the following inequality holds:

$$\|w - z\| \leq \beta(\|x - y\|, \tau) + \gamma(\|u_1 - u_2\|_\infty) \leq \beta(\theta, \tau) + \gamma(\mu),$$

which implies:

$$w \in \mathcal{B}_{\beta(\theta,\tau)+\gamma(\mu)}(z). \tag{7}$$

By combining inclusions in (6) and (7), it is readily seen that $w \in \mathcal{B}_{[\beta(\theta,\tau)+\gamma(\mu)+\eta)}(v)$. By the inequality in (4), $\mathcal{B}_{[\beta(\theta,\tau)+\gamma(\mu)+\eta)}(v) \subseteq \mathcal{B}_{[\theta)}(v)$, which implies $(w, v) \in R$ and hence, condition (iii) in Definition 6 holds. Thus, condition (i) in Definition (7) is satisfied. By using similar arguments it is possible to show condition (ii) of Definition 7. ∎

Consider a plant system P as defined in (2) and a specification system Q as defined in (3). Suppose that P and Q are δ–ISS and choose a precision $\theta_p \in \mathbb{R}^+$ and a precision $\theta_q \in \mathbb{R}^+$, required in the construction of the symbolic systems for P and Q, respectively. Let β_p and γ_p be a $\mathcal{KL}$ function and a $\mathcal{K}_\infty$ function guaranteeing the δ–ISS stability property for P and β_q be a $\mathcal{KL}$ function guaranteeing the δ–ISS stability property for Q. Find quantization parameters $\tau, \eta, \mu \in \mathbb{R}^+$ such that:

$$\begin{aligned} &\beta_p(\theta_p, \tau) + \gamma_p(\mu) + \eta \leq \theta_p, \\ &\beta_q(\theta_q, \tau) + \eta \leq \theta_q. \end{aligned} \tag{8}$$

It is readily seen that parameters $\tau, \eta, \mu \in \mathbb{R}^+$ satisfying the above inequalities always exist. By Theorem 1, $S_{\tau,\eta,\mu}(P)$ is θ_p–approximately bisimilar to $S_\tau(P)$ and $S_{\tau,\eta,0}(Q)$ is θ_q–approximately bisimilar to $S_\tau(Q)$. For the sake of notational simplicity in the further developments we refer to the systems $S_{\tau,\eta,\mu}(P)$ and $S_{\tau,\eta,0}(Q)$ by means of S_p and S_q, respectively.

We now introduce a technical lemma which is employed in the further developments.

Lemma 1: Consider three metric systems $S_i = (X_i, X_{0,i}, U_i, \underset{i}{\longrightarrow}, Y, H_i)$, $i = 1, 2, 3$. The following properties hold:

(i) [10] For all $\varepsilon_1 \in \mathbb{R}_0^+$, if $S_1 \preceq_{\varepsilon_1} S_2$ then $S_1 \preceq_{\varepsilon_2} S_2$, for all $\varepsilon_2 \geq \varepsilon_1$;
(ii) [10] For all $\varepsilon_1, \varepsilon_2 \in \mathbb{R}_0^+$, if $S_1 \preceq_{\varepsilon_1} S_2$ and $S_2 \preceq_{\varepsilon_2} S_3$, then $S_1 \preceq_{\varepsilon_1+\varepsilon_2} S_3$;
(iii) For all $\varepsilon \in \mathbb{R}_0^+$, $S_1 \parallel_\varepsilon S_2 \preceq_\varepsilon S_2$.

Proof: [Proof of (iii)] Denote $S_1 \parallel_\varepsilon S_2 = (X, X_0, U, \longrightarrow, Y, H)$ and define $\mathcal{R} = \{((x_1, x_2), x) \in X \times X_2 : x_2 = x\}$. We start by showing that condition (i) in Definition 6 holds. Consider any initial condition $(x_{0,1}, x_{0,2}) \in X_0$. Since $x_{0,2} \in X_2$, by choosing $x_0 = x_{0,2}$ we have that $((x_{0,1}, x_{0,2}), x_0) \in \mathcal{R}$. We now show that also condition (ii) in Definition 6 holds. Consider any $((x_1, x_2), x) \in \mathcal{R}$. Since $x_2 = x$, then $H_2(x_2) = H_2(x)$, hence by the definition of composition $d(H(x_1, x_2), H_2(x)) = d(H_1(x_1), H_2(x_2)) \leq \varepsilon$. We conclude by showing that condition (iii) in Definition 6 holds. Consider any $((x_1, x_2), x) \in \mathcal{R}$ and any transition $(x_1, x_2) \xrightarrow{(u_1,u_2)} (x_1', x_2')$ in $S_1 \parallel_\varepsilon S_2$. Since $x_2 = x$, choose the transition $x \xrightarrow[2]{u_2} x'$ in S_2 so that $x_2' = x'$. This implies that $((x_1', x_2'), x') \in \mathcal{R}$, which concludes the proof. ■

We are now ready to provide the solution to Problem 1. Define:

$$C^* = S_p \parallel_0 S_q. \tag{9}$$

Theorem 2: Consider the plant system P as in (2), the specification system Q as in (3) and a precision $\varepsilon \in \mathbb{R}^+$. Suppose that P and Q are δ–ISS and choose parameters $\theta_p, \theta_q \in \mathbb{R}^+$ so that:

$$\theta_p + \theta_q \leq \varepsilon. \tag{10}$$

Furthermore choose parameters $\tau, \eta, \mu \in \mathbb{R}^+$ satisfying the inequalities in (8). Then the symbolic controller $Nb(C^*) \in \mathcal{C}^{\tau,\theta,\mu}$ with $\theta = \theta_p$ and C^* defined in (9) with $S_p = S_{\tau,\eta,\mu}(P)$ and $S_q = S_{\tau,\eta,0}(Q)$ solves Problem 1.

Proof: We start by proving condition (i) of Problem 1. By Lemma 1 (iii), we obtain:

$$S_\tau(P) \parallel_{\theta_p} Nb(C^*) \preceq_{\theta_p} Nb(C^*). \tag{11}$$

By the definition of $Nb(C^*)$ it is readily seen that:

$$Nb(C^*) \preceq_0 C^*. \tag{12}$$

By the definition of $C^* = S_p \parallel_0 S_q$ and Lemma 1 (iii), one gets:

$$C^* \preceq_0 S_q. \tag{13}$$

Since S_q is θ_q–approximately bisimilar to $S_\tau(Q)$ then:

$$S_q \preceq_{\theta_q} S_\tau(Q). \tag{14}$$

By combining conditions in (11), (12), (13), (14) and by Lemma 1 (ii) we obtain $S_\tau(P) \parallel_{\theta_p} Nb(C^*) \preceq_{\theta_p+\theta_q} S_\tau(Q)$. Since $\theta_p + \theta_q \leq \varepsilon$ by Lemma 1 (i), condition (i) of Problem 1 is proved. We now prove condition (ii) of Problem 1. Consider any state (p_1, p_2, q) of $S_\tau(P) \parallel_{\theta_p} Nb(C^*)$. Since $Nb(C^*)$ is non–blocking there exists a state (p_2^+, q^+) of $Nb(C^*)$ so that $(p_2, q) \xrightarrow{u} (p_2^+, q^+)$ is a transition of $Nb(C^*)$ for some input $u = (u_2, u_3)$. Since $S_\tau(P)$ and S_p are θ_p–approximately bisimilar, for the transition $p_2 \xrightarrow{u_2} p_2^+$ in S_p there exists a transition $p_1 \xrightarrow{u_1} p_1^+$ in $S_\tau(P)$ so that $d(H_p(p_1^+), H_p(p_2^+)) \leq \theta = \theta_p$. This implies that (p_1^+, p_2^+, q^+) is a state of $S_\tau(P) \parallel_{\theta_p} Nb(C^*)$ and therefore that $(p_1, p_2, q) \xrightarrow{(u_1,u)} (p_1^+, p_2^+, q^+)$ is a transition of $S_\tau(P) \parallel_{\theta_p} Nb(C^*)$, which concludes the proof. ■

V. Integrated Symbolic Control Design

The construction of the controller $Nb(S_p \parallel_0 S_q)$ solving Problem 1 relies upon the basic–steps procedure illustrated in Algorithm 1.

1 Construct the symbolic system S_p, θ_p–approximately bisimilar to $S_\tau(P)$;
2 Construct the symbolic system S_q, θ_q–approximately bisimilar to $S_\tau(Q)$;
3 Construct the composition $S_p \parallel_0 S_q$;
4 Compute the non–blocking part $Nb(S_p \parallel_0 S_q)$ of $S_p \parallel_0 S_q$

Algorithm 1: Construction of $Nb(S_p \parallel_0 S_q)$.

The software implementation of Algorithm 1 requires that:

- The state space X_p and set of input values U_p of P are bounded;
- The state space X_q of Q is bounded.

The above assumptions, while being reasonable in many realistic engineering control problems, are also needed to store the symbolic states of the symbolic systems in a computer machine, whose memory resources are limited by their nature. In this section, we suppose that the plant P and the specification Q satisfy the above assumptions.
Space complexity required in the computation of the controller $Nb(C^*)$ is discussed in the following result.

Proposition 2: Space complexity of Algorithm 1 is

$$O(\max\{card([X_p]_{2\eta}) \cdot card([U_p]_{2\mu}), card([X_q]_{2\eta})\}).$$

Proof: The number of transitions of S_p amounts in the worst case to $card([X_p]_{2\eta}) \cdot card([U_p]_{2\mu})$ since by Proposition 1, system S_p is deterministic. For the same reason, the number of transitions in S_q is given by $card([X_q]_{2\eta})$. By definition of exact composition, the number of transitions in $S_p \parallel_0 S_q$ amounts in the worst case to $(card([X_p]_{2\eta}) \cap card([X_q]_{2\eta})) \cdot card([U_p]_{2\mu})$. Hence, by comparing the above worst case bounds, the result follows. ■

Algorithm 1 is not efficient from the space complexity point of view because: (i) It considers the whole state spaces of the plant P and the specification Q. A more efficient algorithm would consider only the intersection of the accessible parts of P and Q; (ii) For any source state x and target state y it includes all transitions (x, u, y) with any control input u by which state x reaches state y. A more efficient algorithm would consider for any source state x and target state y only one control input u and hence, only one transition; (iii) It first construct the symbolic models S_p and S_q, then the composed system $S_p \parallel_0 S_q$ to finally eliminate blocking states from $S_p \parallel_0 S_q$. A more efficient algorithm would eliminate blocking states as soon as they show up. Inspired from the research line in the context of on–the–fly verification and control of timed or untimed transition systems (see e.g. [19], [20]), we now present an algorithm which *integrates each step of the four sub–algorithms in*

Algorithm 1 in only one algorithm. The procedure that we now present is composed of Algorithm 2 and Algorithm 3. Algorithm 2 is the main one while Algorithm 3 introduces Function **NonBlock**, which is recursively used in Algorithm 2. Given a set $T \subseteq X \times U \times Y$, the set $\mathbf{X}_{source}(T) \subseteq X$ denotes the projection of T onto X, i.e. $\mathbf{X}_{source}(T) = \{x \in X : \exists y \in Y \wedge \exists u \in U \text{ s.t. } (x,u,y) \in T\}$. Given a vector $x \in \mathbb{R}^n$ and a precision $\eta \in \mathbb{R}^+$, the symbol $[x]_{2\eta}$ denotes the unique vector in $[\mathbb{R}^n]_{2\eta}$ such that $x \in \mathcal{B}_{[\eta)}([x]_{2\eta})$. Algorithm 2 proceeds as follows. The set of states X_0 of C^{**} is initialized to be $[X_{p,0} \cap X_{q,0}]_{2\eta}$ (line 2.8) and the set of states to be processed, denoted by $\mathbf{X}_{target}$, is initialized to the set of initial states (line 2.9). The set T of transitions and the set Bad of blocking states of C^{**} are initialized to be the empty–sets (lines 2.10, 2.11). At each basic step, the algorithm processes a (non processed) state (line 2.12), by computing the state $y = [\xi^q_x(\tau)]_{2\eta}$ (line 2.13). If the state y is non–blocking (line 2.14), the algorithm looks for a control input $u \in [U]_{2\mu}$ such that the plant P meets the specification Q, i.e. $z = y$ (line 2.18). If such a control input u exists, then the loop is broken (line 2.20), the transition (x,u,y) is added to the set of transitions T (line 2.24), and the state y is added to the set of the to–be–processed states (line 2.25). If either y is blocking or no inputs are found for the plant P to meet the specification Q, then the state x is declared blocking, and Function **NonBlock**(T,x,Bad) in Algorithm 3 is invoked (line 2.29), in order to remove all blocking states originating from x. Algorithm 2 proceeds with further basic steps, until there are no more states to be processed. When the algorithm terminates, it returns (line 2.32) the symbolic controller C^{**}. Function **NonBlock**(T,x,Bad) extracts the non–blocking part of T. The set $BadBis$ includes the states to be processed and is initialized to contain the only state x (line 3.3). At each basic step, for any $y \in BadBis$, Function **NonBlock** removes from the set T any transition (z,u,y) ending up in y (line 3.7), it adds z in the set $BadBis$ of states to be processed (line 3.8) and moves y to the set Bad of blocking states (lines 3.11, 3.12). Function **NonBlock** terminates when there are no more states to be processed and returns (line 2.14) the updated sets of transitions T and blocking states Bad. Termination of Algorithm 2 is discussed in the following result:

Theorem 3: Algorithm 2 terminates in a finite number of steps.

Proof: Algorithm 2 terminates when there are no more states x in $\mathbf{X}_{target}$ to be processed. For each state x, either line 2.24 or line 2.29 is executed; this ensures by line 2.12 that state x cannot be processed again in future iterations. Furthermore, the set $\mathbf{X}_{target}$ is nondecreasing (see line 2.25) and always contained in the finite set $[X_p]_{2\eta} \cap [X_q]_{2\eta}$. Hence, provided that Algorithm 3 terminates in finite time, the result follows. Regarding termination of Algorithm 3, in the worst case the set Bad ends up to coincide with the accessible states of S_p and S_q (line 3.12) and the set $BadBis$ ends up to be empty (lines 3.11). Hence from line 3.4, finite termination of Algorithm 3 is guaranteed. ■

Formal correctness of Algorithm 2 is guaranteed by the

```
1  Input:
2  Plant: P = (X_p, X_{p,0}, U_p, 𝒰_p, f_p);
3  Specification: Q = (X_q, X_{q,0}, U_q, 𝒰_q, f_q);
4  Precision: ε ∈ ℝ^+;
5  Parameters: θ_p, θ_q ∈ ℝ^+ satisfying (10);
6  Parameters: τ, η, μ ∈ ℝ^+ satisfying (8);
7  Init:
8  X_0 := [X_{p,0} ∩ X_{q,0}]_{2η};
9  X_target = X_0;
10 T := ∅;
11 Bad := ∅;
12 foreach x ∈ X_target \ (X_source(T) ∪ Bad) do
13     compute y = [ξ^q_x(τ)]_{2η};
14     if y ∉ Bad then
15         Flag := 0;
16         foreach u ∈ [U_p]_{2μ} do
17             compute z = [ξ^p_{xu}(τ)]_{2η};
18             if z = y then
19                 Flag := 1;
20                 break foreach u ∈ [U_p]_{2μ};
21             end
22         end
23         if Flag = 1 then
24             T := T ∪ {(x,u,y)};
25             X_target = X_target ∪ {y};
26         end
27     end
28     if Flag = 0 ∨ y ∈ Bad then
29         (T, Bad) :=NonBlock(T, x, Bad);
30     end
31 end
32 output: C** = (X_source(T), X_0 ∩ X_source(T),
33 [U_p]_{2μ}, T, Y_{τ,η,μ}, H_{τ,η,μ})
```

Algorithm 2: Integrated Control Design.

```
1  Function (T, Bad) :=NonBlock(T, x, Bad);
2  Init:
3  BadBis := {x};
4  foreach y ∈ BadBis do
5      foreach z ∈ X_source(T) do
6          if ∃u ∈ [U]_{2μ} such that (z,u,y) ∈ T then
7              T := T \ {(z,u,y)};
8              BadBis := BadBis ∪ {z};
9          end
10     end
11     BadBis := BadBis \ {y};
12     Bad := Bad ∪ {y};
13 end
14 output: (T, Bad)
```

Algorithm 3: Non–blocking Algorithm.

following result.

Theorem 4: Controllers $Nb(C^*)$ and C^{**} are exactly bisimilar.

Proof: (Sketch.) For any state (x_p, x_q) of the accessible part $Ac(Nb(C^*))$ of $Nb(C^*)$ there exists a state x_c of C^{**} so that $x_p = x_q = x_c$ (see lines 2.13, 2.17, 2.18 and 2.24 in Algorithm 2). Consider the relation defined by $((x_p, x_q), x_c) \in \mathcal{R}$ if and only if $x_p = x_c$. It is readily seen that $\mathcal{R}$ is a 0–bisimulation relation between $Nb(C^*)$ and C^{**}. ■

While the controllers $Nb(C^*)$ and C^{**} are exactly bisimilar, the number of states of C^{**} is in general, smaller than the one of $Nb(C^*)$. In fact, it is easy to see that the controller $Nb(C^*)$ may contain spurious states, e.g. states which are not reachable from a quantized initial condition in S_p and a quantized initial condition in S_q, since in general $Ac(Nb(C^*))$ is a (strict) sub–system of $Nb(C^*)$. On the other hand, a straightforward inspection of Algorithm 2 reveals that:

Proposition 3: $Ac(C^{**}) = C^{**}$,

and hence, the aforementioned spurious states of $Nb(C^*)$ are not included in C^{**}. The above remarks suggest the following formal statement:

Theorem 5: C^{**} is the minimal 0–bisimilar system of $Nb(C^*)$.

Proof: Since by Proposition 3 $Ac(C^{**}) = C^{**}$ and since the output function $H_{\tau,\eta,\mu}$ of C^{**} is the natural inclusion from $\mathbf{X}_{source}(T)$ to X, the maximal 0–bisimulation relation $\mathcal{R}^*$ between C^{**} and itself is the identity relation, i.e. $\mathcal{R}^* = \{(x_1, x_2) \in \mathbf{X}_{source}(T) \times \mathbf{X}_{source}(T) : x_1 = x_2\}$. Since $\mathcal{R}^*$ is the identity relation, the quotient of C^{**} induced by $\mathcal{R}^*$, coincides with C^{**}. Finally, since by Theorem 4 systems C^{**} and $Nb(C^*)$ are 0–bisimilar, the result follows. ■

The above result is important because it shows that the controller C^{**} is the system with the smallest number of states which is equivalent by bisimulation to the solution $Nb(C^*)$ of Problem 1. We conclude this section by discussing the space complexity in the construction of C^{**}.

Proposition 4: Space complexity of Algorithm 2 is $O(card([X_p]_{2\eta} \cap [X_q]_{2\eta}))$.

Proof: By lines 2.13, 2.17, 2.18, and 2.24 in Algorithm 2, the triple (x, u, y) is added to the set T of transitions of C^{**}, if (x, u, y) is a transition of S_p and (x, y) is a transition of S_q. Hence, the result follows from determinism of systems S_p and S_q, which is guaranteed by Proposition 1. ■

By comparing Propositions 2 and 4, it is easy to see that the space complexity associated with the computation of C^{**} is smaller than the ones associated with the computation of $Nb(C^*)$. In the following section, we present an example illustrating the benefits from the use of the integrated control design procedure presented in this section.

VI. An illustrative example

Consider the following plant nonlinear control system:

$$P : \begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -1.96 \sin x_1 - 10 x_2 + u, \end{cases}$$

Statistics	$Nb(C^*)$	C^{**}	Ratio
States	327	87	0.26
Transitions	2901	87	0.03
Max memory occupation (integer)	9246981	1151	$1.24 \cdot 10^{-4}$
Time (s)	39950	5292	0.13

TABLE I

and a specification, expressed by the following nonlinear system:

$$Q : \begin{cases} \dot{x}_1 = -2x_1 + x_2^2, \\ \dot{x}_2 = -4 \left(\sin x_2 + x_2^3 \right). \end{cases}$$

For simplicity, we consider the same state space for the plant and the specification, chosen as $X_p = X_q = [-1, 1[\times [-1, 1[$, the same set of initial states, chosen as $X_p^0 = X_q^0 = [-0.5, 0.5[\times [-0.25, 0.25[$, and the same set of input values, chosen as $U = [-1.5, 1.5]$. The plant system P was shown in [13] to be a δ–ISS with functions $\beta_p(r, s) = 10.1 e^{-4.06s} r$ and $\gamma(r) = \sqrt{2.99r}$, while the specification system Q can be shown to be δ–ISS with function $\beta_q(r, s) = \sqrt{2} e^{-s} r$. For a precision $\varepsilon = 0.15$, we can choose $\theta_p = 0.133$, $\theta_q = 0.017$, $\eta = 0.01$, $\tau = 2$ and $\mu = 0.005$ so that inequalities in (8) and (10) are satisfied. The construction of the controller C^{**} has been performed on an Intel Core 2 Duo T5500 1.66GHz laptop with 4 GB RAM. In Table I we report the experimental results obtained in the computation of the controller C^{**} and compare them to the ones associated with the construction of $Nb(C^*)$ as in Algorithm 1: the number of states of C^{**} is 26% times the number of states of $Nb(C^*)$, the number of transitions of C^{**} is 3% times the number of transitions of $Nb(C^*)$; more interestingly, the maximal occupation of memory[2] required in the construction of C^{**} is $1.24 \cdot 10^{-2}$% times the maximal occupation of memory required in the construction of $Nb(C^*)$; finally the time of computation needed in the construction of C^{**} is 13% times the time of computation of $Nb(C^*)$.

VII. Discussion

In this paper we proposed a methodology for the integrated symbolic control design of nonlinear systems with infinite states specifications, modelled by differential equations. Although the focus of the present paper is on infinite states specifications, it is readily seen that the results here shown can be easily applied to finite state specifications which include language specifications, formalized through automata theory. This is important because, as shown in the work of [5], [18], [7], automata theory provides a novel class of specifications which were traditionally not addressed before, in the control design of continuous (nonlinear) systems. Algorithm 2 is presented which integrates the synthesis of the controller with the construction of the symbolic systems of the plant and of the specification. Future work will focus on efficient techniques at the software layer, which can further

[2]The maximal memory occupation is given in terms of the maximal number of data needed in the construction of the controllers. Each transition is weighted as three data and each state as one datum.

reduce space complexity in the implementation of Algorithm 2.

Acknowledgement. The second author would like to thank Paulo Tabuada for having suggested the idea of integration of control algorithms with construction of symbolic systems of the plant and the specification.

APPENDIX: NOTATION

The identity map on a set A is denoted by 1_A. Given two sets A and B, if A is a subset of B we denote by $1_A : A \hookrightarrow B$ or simply by $\imath$ the natural inclusion map taking any $a \in A$ to $\imath(a) = a \in B$. Given a function $f : A \to B$ the symbol $f(A)$ denotes the image of A through f, i.e. $f(A) := \{b \in B : \exists a \in A \text{ s.t. } b = f(a)\}$. We identify a relation $R \subseteq A \times B$ with the map $R : A \times 2^B$ defined by $b \in R(a)$ if and only if $(a, b) \in R$. Given a relation $R \subseteq A \times B$, R^{-1} denotes the inverse relation of R, i.e. $R^{-1} := \{(b, a) \in B \times A : (a, b) \in A \times B\}$.
The symbols $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{R}_0^+$ denote the set of natural, integer, real, positive real, and nonnegative real numbers, respectively. Given a vector $x \in \mathbb{R}^n$, we denote by x_i the i–th element of x and by $\|x\|$ the infinity norm of x, we recall that $\|x\| = \max\{|x_1|, |x_2|, ..., |x_n|\}$, where $|x_i|$ denotes the absolute value of x_i. Given a measurable function $f : \mathbb{R}_0^+ \to \mathbb{R}^n$, the (essential) supremum of f is denoted by $\|f\|_\infty$; we recall that $\|f\|_\infty = (\text{ess}) \sup \{\|f(t)\|, t \geq 0\}$; f is essentially bounded if $\|f\|_\infty < \infty$. Given $x \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}^+$, the symbol $\mathcal{B}_\varepsilon(x)$ denotes the set $\{x \in \mathbb{R}^n : \|x\| \leq \varepsilon\}$ and the symbol $\mathcal{B}_{[\varepsilon)}(x)$ denotes the set $[-\varepsilon + x_1, x_1 + \varepsilon) \times [-\varepsilon + x_2, x_2 + \varepsilon) \times ... \times [-\varepsilon + x_n, x_n + \varepsilon)$ where x_i is the i–th element of x. It is readily seen that if $x \in \mathcal{B}_\varepsilon(y)$ and $y \in \mathcal{B}_{[\theta)}(z)$ then $x \in \mathcal{B}_{[\varepsilon+\theta)}(z)$. For any $A \subseteq \mathbb{R}^n$ and $\mu \in \mathbb{R}^+$, define $[A]_\mu = \{a \in A \mid a_i = k_i\mu, k_i \in \mathbb{Z}, i = 1, 2, ..., n\}$. The set $[A]_\mu$ is used as an approximation of the set A with precision $\mu/2$. A function $f : [a, b] \to \mathbb{R}^n$ is said to be absolutely continuous on $[a, b]$ if for any $\varepsilon \in \mathbb{R}^+$ there exists $\delta \in \mathbb{R}^+$ so that for every $k \in \mathbb{N}$ and for every sequence of points $a \leq a_1 < b_1 < a_2 < b_2 < ... < a_k < b_k \leq b$, if $\sum_{i=1}^k (b_i - a_i) < \delta$ then $\sum_{i=1}^k |f(b_i) - f(a_i)| < \varepsilon$. A function $f :]a, b[\to \mathbb{R}^n$ is said to be locally absolutely continuous if the restriction of f to any compact subset of $]a, b[$ is absolutely continuous. For a given time $\tau \in \mathbb{R}^+$, define f_τ so that $f_\tau(t) = f(t)$, for any $t \in [0, \tau)$, and $f(t) = 0$ elsewhere; f is said to be locally essentially bounded if for any $\tau \in \mathbb{R}^+$, f_τ is essentially bounded. A continuous function $\gamma : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$; γ is said to belong to class $\mathcal{K}_\infty$ if $\gamma \in \mathcal{K}$ and $\gamma(r) \to \infty$ as $r \to \infty$. A continuous function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is said to belong to class $\mathcal{KL}$ if, for each fixed s, the map $\beta(r, s)$ belongs to class $\mathcal{K}_\infty$ with respect to r and, for each fixed r, the map $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to \infty$.

REFERENCES

[1] M. Egerstedt, E. Frazzoli, and G. J. Pappas, *IEEE Transactions of Automatic Control*, vol. 51, no. 6, June 2006, special Issue on Symbolic Methods for Complex Control Systems.

[2] R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[3] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *Journal of Computer and System Sciences*, vol. 57, pp. 94–124, 1998.

[4] G. Lafferriere, G. J. Pappas, and S. Sastry, "O-minimal hybrid systems," *Mathematics of Control, Signals and Systems*, vol. 13, no. 1, pp. 1–21, March 2000.

[5] P. Tabuada and G. Pappas, "Linear Time Logic control of discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1862–1877, 2006.

[6] L. Habets, P. Collins, and J. V. Schuppen, "Reachability and control synthesis for piecewise-affine hybrid systems on simplices," *IEEE Transaction on Automatic Control*, vol. 51, no. 6, pp. 938–948, 2006.

[7] C. Belta and L. Habets, "Controlling a class of nonlinear systems on rectangles," *IEEE Transactions of Automatic Control*, vol. 51, no. 11, pp. 1749–1759, 2006.

[8] R. Milner, *Communication and Concurrency*. Prentice Hall, 1989.

[9] D. Park, "Concurrency and automata on infinite sequences," ser. Lecture Notes in Computer Science, Springer-Verlag, Ed., vol. 104, 1981, pp. 167–183.

[10] A. Girard and G. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 782–798, 2007.

[11] D. Angeli, "A Lyapunov approach to incremental stability properties," *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.

[12] A. Girard, "Approximately bisimilar finite abstractions of stable linear systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Berlin: Springer Verlag, 2007, vol. 4416, pp. 231–244.

[13] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, pp. 2508–2516, October 2008.

[14] G. Pola and P. Tabuada, "Symbolic models for nonlinear control systems: Alternating approximate bisimulations," *SIAM Journal on Control and Optimization*, vol. 48, no. 2, pp. 719–733, 2009.

[15] G. Pola, P. Pepe, M. D. Benedetto, and P. Tabuada, "A symbolic model approach to the digital control of nonlinear timedelay systems," in *48th IEEE Conference on Decision and Control*, Shanghai, P.R. China, December 2009, pp. 2216–2221.

[16] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *IEEE Transactions of Automatic Control*, vol. 55, no. 1, pp. 116–126, January 2010.

[17] B. Yordanov and C. Belta., "Temporal logic control of discrete-time piecewise affine systems," in *48th IEEE Conference on Decision and Control*, Shanghai, P.R. China, December 2009, pp. 3182–3187.

[18] P. Tabuada, "An approximate simulation approach to symbolic control," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1406–1418, 2008.

[19] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis, "Memory-efficient algorithms for the verification of temporal properties," *Formal Methods in System Design*, vol. 1, no. 2-3, pp. 275–288, 1992.

[20] S. Tripakis and K. Altisen, "On-the-fly controller synthesis for discrete and dense-time systems," in *World Congress on Formal Methods in the Development of Computing Systems*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, September 1999, vol. 1708, pp. 233 – 252.

[21] D. Angeli and E. Sontag, "Forward completeness, unboundedness observability, and their lyapunov characterizations," *Systems and Control Letters*, vol. 38, pp. 209–217, 1999.

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