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).

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:

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

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 $K \subseteq \mathbb{R}^n$, 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 $\Sigma$ if there exists $u \in \mathcal{U}$ satisfying $\xi(t) = f(\xi(t), u(t))$ 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 $\xi_u(\tau)$ to denote the point reached at time $\tau$ 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 $\Sigma$ by means of the tuple $(X, X_0, f)$. A control system $\Sigma$ 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:

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

consisting of:

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

A transition $(x, u, x') \in \rightarrow$ 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 $(x, u, x') \in \rightarrow$; non-blocking, if for any $x \in X$ there exists $(x, u, x') \in \rightarrow$; 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 $S_1 = (X_1, X_{0,1}, U_1, \rightarrow, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \rightarrow, Y_2, H_2)$, system $S_1$ is a sub–system of $S_2$ if $X_1 \subseteq X_2, X_{0,1} \subseteq X_{0,2}, U_1 \subseteq U_2, H_1(x) = H_2(x)$ for any $x \in X_1$.

Definition 4: Given a system $S = (X, X_0, U, \rightarrow, Y, H)$ 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 $S = (X, X_0, U, \rightarrow, Y, H)$ 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 $S_1 = (X_1, X_{0,1}, U_1, \rightarrow, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \rightarrow, Y_2, H_2)$ 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 $\varepsilon$–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:

$$x_1 \xrightarrow{u_1} x_1' \text{ in } S_1 \text{ implies the existence of } x_2 \xrightarrow{u_2} x_2' \text{ in } S_2 \text{ satisfying } (x_1', x_2') \in \mathcal{R}.$$

System $S_1$ is $\varepsilon$–approximately simulated by $S_2$ or $S_2$ $\varepsilon$–approximately simulates $S_1$, denoted by $S_1 \preceq_{\varepsilon} S_2$, if there exists an $\varepsilon$–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 
\[ S_1 = (X_1, X_{0,1}, U_1, Y_1, H_1) \] 
and 
\[ S_2 = (X_2, X_{0,2}, U_2, Y_2, H_2) \]
be metric systems with the same output sets \( Y_1 = Y_2 \) and metric \( d \), and consider a precision \( \varepsilon \in \mathbb{R}^+ \). A relation \( R \subseteq X_1 \times X_2 \) is said to be an \( \varepsilon \)-approximation bisimulation relation between \( S_1 \) and \( S_2 \) if the following conditions are satisfied:

(i) \( R \) is an \( \varepsilon \)-approximate simulation relation from \( S_1 \) to \( S_2 \);

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

System \( S_1 \) is \( \varepsilon \)-approximate bisimilar to \( S_2 \), denoted by \( S_1 \cong_{\varepsilon} S_2 \), if there exists an \( \varepsilon \)-approximate bisimulation relation \( R \) 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 \( S_1 = (X_1, X_{0,1}, U_1, Y_1, H_1) \) and \( S_2 = (X_2, X_{0,2}, U_2, Y_2, H_2) \), with the same output sets \( Y_1 = Y_2 \) and metric \( d \), and a precision \( \varepsilon \in \mathbb{R}^+ \), the \( \varepsilon \)-approximate composition of \( S_1 \) and \( S_2 \) is the system \( S_1 \parallel S_2 := (X, X_0, U, \varepsilon, Y, H) \), 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 \( x_1 \xrightarrow{u_1} x'_1 \) and \( x_2 \xrightarrow{u_2} x'_2 \);
- \( Y = Y_1 \);
- \( H : X_1 \times X_2 \to Y \) 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 \( \Sigma \):

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

where:

- \( \mathcal{U} = \{u \in U \mid \text{ the domain of } u \text{ is } [0, \tau] \} \);
- \( x \xrightarrow{\tau} x' \) if there exists a trajectory \( \xi : [0, \tau] \to X \) of \( \Sigma \) satisfying \( \xi_{x_0}(\tau) = x' \);
- \( Y = X \);
- \( H = 1_X \).

System \( S_{\tau}(\Sigma) \) is metric when we regard \( Y = X \) as being equipped with the metric \( d(p, q) = ||p - q|| \).

### 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 notation

\[ B_{[\theta]}(y) \text{ is defined in the Appendix.} \]
of incremental input–to–state stability for nonlinear control systems.

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

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

A characterization of the above incremental stability notion in terms of dissipation inequalities can be found in [11]. Given a $\delta$-ISS nonlinear control system $\Sigma$ 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:

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

where:

- $X_{\tau,\eta,\mu} = [X]_{2^\mu};$
- $X_{0,\tau,\eta,\mu} = X_{\tau,\eta,\mu} \cap X_0;$
- $U_{\tau,\eta,\mu} = [U]_{2^\mu};$
- $x \xrightarrow{\tau,\eta,\mu} y$ if $\xi_{ux}(\tau) \in B\{y\}(y) \cap X;$
- $Y_{\tau,\eta,\mu} = X;$
- $H_{\tau,\eta,\mu} = 1 : X_{\tau,\eta,\mu} \rightarrow Y_{\tau,\eta,\mu}.$

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 $\xi_{ux}(\tau)$ is uniquely determined. Moreover, since the collection of sets $\{B\{y\}(y) \cap X\}_{y \in X_{\tau,\eta,\mu}}$ is a partition of $X$, there exists at most one state $y \in Y_{\tau,\eta,\mu}$ such that $\xi_{ux}(\tau) \in B\{y\}(y) \cap X.$

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 $\delta$-ISS nonlinear control system $\Sigma = (X, X_0, U, f)$ 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:

$$\beta(\theta, \tau) + \gamma(\mu) + \eta \leq \theta,$$

systems $S_{\tau,\eta,\mu}(\Sigma)$ and $S_\tau(\Sigma)$ are $\delta$–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 $x \in B\{y\}(y) \cap X$. 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 $u_1 \in U$, and the transition $x \xrightarrow{\tau_1} w$ in $S_\tau(\Sigma)$. Then there exists $u_2 \in U_{\tau,\eta,\mu}$ such that:

$$\|u_2 - u_1\|_\infty \leq \mu,$$

and set $z = \xi_{u_2}(\tau)$. Since $X = \bigcup_{v \in X_{\tau,\eta,\mu}} B\{v\}(v) \cap X$, there exists $v \in X_{\tau,\eta,\mu}$ such that:

$$z \in B\{v\}(v),$$

and therefore $y \xrightarrow{\tau,\eta,\mu} v$ in $S_{\tau,\eta,\mu}(\Sigma)$. Since $\Sigma$ is $\delta$-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:

$$w \in B_{\beta(\theta, \tau) + \gamma(\mu)}(z).$$

By combining inclusions in (6) and (7), it is readily seen that $w \in B_{\beta(\theta, \tau) + \gamma(\mu) + \eta}(v)$. By the inequality in (4), $B_{\beta(\theta, \tau) + \gamma(\mu) + \eta}(v) \subseteq B\{v\}(v)$, 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 $\delta$–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 $\beta_p$ and $\gamma_p$ be a $KL$ function and a $K\infty$ function guaranteeing the $\delta$–ISS stability property for $P$ and $\beta_q$ be a $KL$ function guaranteeing the $\delta$–ISS stability property for $Q$. Find quantization parameters $\tau_p, \eta, \mu \in \mathbb{R}^+$ such that:

$$\beta_p(\theta_p, \tau) + \gamma_p(\mu) + \eta \leq \theta_p,$$

$$\beta_q(\theta_p, \tau) + \eta \leq \theta_q.'
Proof: [Proof of (iii)] Denote \( S_1 \parallel \varepsilon \parallel \varepsilon \parallel S_2 = (X, X_0, U, \longrightarrow, Y, H) \) 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 \((x_{01}, x_{02}) \in \mathcal{X}_0\). Since \( x_{02} \in X_2 \), by choosing \( x_0 = x_{02} \) we have that \((x_{01}, x_{02}, x_0) \in \mathcal{R}\). 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 \( (x_1, x_2) \xrightarrow{u_{i1},u_{i2}} (x'_1, x'_2) \) in \( S_1 \parallel S_2 \). Since \( x_2 = x \), choose the transition \( x \xrightarrow{u_2} x' \) in \( S_2 \) so that \( x'_1 = x' \). This implies that \((x'_1, x'_2, x') \in \mathcal{R}\), which concludes the proof. \( \square \)

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

Define:

\[
C^* = S_p \parallel S_q. \tag{9}
\]

**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 \( \delta \)-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 \( Nb(C^*) \in C^* \mu \eta \theta \) with \( \theta = \theta_p \) and \( C^* \) defined in (9) with \( S_p = S_{r, \eta, \mu, (P)} \) and \( S_q = S_{r, \eta, \mu, (Q)} \) solves Problem 1.

**Proof:** We start by proving condition (i) of Problem 1.

By Lemma 1 (iii), we obtain:

\[
S_\tau(P) \parallel \theta_p \parallel Nb(C^*) \parallel \theta_q \parallel Nb(C^*). \tag{11}
\]

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

\[
Nb(C^*) \leq Nb(C^*). \tag{12}
\]

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

\[
C^* \leq C^*. \tag{13}
\]

Since \( S_q \) is \( \theta_q \)-approximately bisimilar to \( S_\tau(Q) \) then:

\[
S_q \leq \theta_q, S_\tau(Q). \tag{14}
\]

By combining conditions in (11), (12), (13), (14) and by Lemma 1 (ii) we obtain \( S_\tau(P) \parallel \theta_p \parallel Nb(C^*) \parallel \theta_q \parallel S_\tau(Q) \). 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 \((p_1, p_2, q) \in S_\tau(P) \parallel \theta_p \parallel Nb(C^*) \) and \( Nb(C^*) \) is non–blocking there exists a state \((p_2^*, q^*)\) of \( Nb(C^*) \) so that \((p_2, q) \xrightarrow{u_2} (p_2^*, q^*) \) is a transition of \( Nb(C^*) \) for some input \( u = (u_2, u_3) \). Since \( S_\tau(P) \) and \( S_q \) are \( \theta_q \)-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 \( d(H_2(p_1^*), H_2(p_2^*)) \leq \theta = \theta_p \). This implies that \((p_1^*, p_2^*, q^*)\) is a state of \( S_\tau(P) \parallel \theta_p \parallel Nb(C^*) \) and therefore that \((p_1, p_2, q) \xrightarrow{u_1} (p_1^*, p_2^*, q^*) \) is a transition of \( S_\tau(P) \parallel \theta_p \parallel Nb(C^*) \), which concludes the proof. \( \square \)

\( V. \) **INTEGRATED SYMBOLIC CONTROL DESIGN**

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

1. Construct the symbolic system \( S_p \), \( \theta_p \)-approximately bisimilar to \( S_\tau(P) \).
2. Construct the symbolic system \( S_q \), \( \theta_q \)-approximately bisimilar to \( S_\tau(Q) \).
3. Construct the composition \( S_p \parallel S_q \).
4. Compute the non–blocking part \( Nb(S_p \parallel S_q) \) of \( S_p \parallel S_q \).

**Algorithm 1:** Construction of \( Nb(S_p \parallel S_q) \).

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

\[
O(\max\{\text{card}(X_\tau)|_{20}, \text{card}(U_\tau)|_{20}, \text{card}(X_q)|_{20}\}).
\]

**Proof:** The number of transitions of \( S_p \) amounts in the worst case to \( \text{card}(X_\tau)|_{20} \cdot \text{card}(U_\tau)|_{20} \) since by Proposition 1, system \( S_q \) is deterministic. For the same reason, the number of transitions in \( S_q \) is given by \( \text{card}(X_q)|_{20} \). By definition of exact composition, the number of transitions in \( S_p \parallel S_q \) amounts in the worst case to \( (\text{card}(X_\tau)|_{20} \cap \text{card}(X_q)|_{20}) \cdot \text{card}(U_q)|_{20} \). 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 \( y \). 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 \( S_p \parallel S_q \) to finally eliminate blocking states from \( S_p \parallel S_q \). 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 \( X_{\text{source}}(T) \subseteq X \) denotes the projection of \( T \) onto \( X \), i.e. \( X_{\text{source}}(T) = \{ x \in X : \exists y \in Y \land \exists u \in U \text{ s.t. } (x, u, y) \in T \} \). 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 \( x \in B_{[x]}([x]_{2\eta}) \).

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 \( X_{\text{target}} \), is initialized to the set of initial states (line 2.9). The set \( T \) of transitions and the set \( \text{Bad} \) 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 \( y = [\xi_{\text{u}}(\tau)]_{2\eta} \) (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 (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\((T, x, \text{Bad})\) 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\((T, x, \text{Bad})\) extracts the non–blocking part of \( T \). The set \( \text{BadBis} \) includes the states to be processed and is initialized to contain the only state \( x \) (line 3.3). At each basic step, for any \( y \in \text{BadBis} \), 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 \( \text{BadBis} \) of states to be processed (line 3.8) and moves \( y \) to the set \( \text{Bad} \) 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 \( \text{Bad} \). 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 \( X_{\text{target}} \) 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 \( X_{\text{target}} \) is nondecreasing (see line 2.25) and always contained in the finite set \([X_{p,0}]_{2\eta} \cap [X_{q,0}]_{2\eta}\). Hence, provided that Algorithm 3 terminates in finite time, the result follows. Regarding termination of Algorithm 3, in the worst case the set \( \text{Bad} \) ends up to coincide with the accessible states of \( S_p \) and \( S_q \) (line 3.12) and the set \( \text{BadBis} \) 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

---

**Algorithm 2:** Integrated Control Design.

```
1 Input: 
2 Plant: \( P = (X_p, X_{p,0}, U_p, U_q, f_p) \); 
3 Specification: \( Q = (X_q, X_{q,0}, U_q, U_q, f_q) \); 
4 Precision: \( \varepsilon \in \mathbb{R}^+ \); 
5 Parameters: \( \theta_p, \theta_q \in \mathbb{R}^+ \) satisfying (10); 
6 Parameters: \( \tau, \eta, \mu \in \mathbb{R}^+ \) satisfying (8); 
7 Init: 
8 \( X_0 := [X_{p,0} \cap X_{q,0}]_{2\eta} \); 
9 \( X_{\text{target}} = X_0 \); 
10 \( T := \emptyset \); 
11 \( \text{Bad} := \emptyset \); 
12 foreach \( x \in X_{\text{target}} \) \( \cup \) \( X_{\text{source}}(T) \) \cup \( \text{Bad} \) do 
13 compute \( y = [\xi_{\text{u}}(\tau)]_{2\eta} \); 
14 if \( y \notin \text{Bad} \) then 
15 \( \text{Flag} := 0 \); 
16 foreach \( u \in [U]_{2\mu} \) do 
17 compute \( z = [\xi_{\text{u}}(\tau)]_{2\eta} \); 
18 if \( z = y \) then 
19 \( \text{Flag} := 1 \); 
20 break \( u \in [U]_{2\mu} \); 
21 end 
22 end 
23 if \( \text{Flag} = 1 \) then 
24 \( T := T \cup \{(x, u, y)\} \); 
25 \( X_{\text{target}} = X_{\text{target}} \cup \{y\} \); 
26 end 
27 end 
28 if \( \text{Flag} = 0 \lor y \in \text{Bad} \) then 
29 \( (T, \text{Bad}) := \text{NonBlock}(T, x, \text{Bad}) \); 
30 end 
31 end 
32 output: \( C^{**} = (X_{\text{source}}(T), X_0 \cap X_{\text{source}}(T), [U]_{2\mu}, T, X_{\tau,\eta,\mu} \subset H_{\tau,\eta,\mu}) \)
```

---

**Algorithm 3:** Non–blocking Algorithm.

```
1 Function \( (T, \text{Bad}) := \text{NonBlock}(T, x, \text{Bad}) \); 
2 Init: 
3 \( \text{BadBis} := \{x\} \); 
4 foreach \( y \in \text{BadBis} \) do 
5 \( \text{foreach } z \in X_{\text{source}}(T) \text{ do} \) 
6 if \( \exists u \in [U]_{2\mu} \text{ such that } (z, u, y) \in T \) then 
7 \( T := T \setminus \{(z, u, y)\} \); 
8 \( \text{BadBis} := \text{BadBis} \cup \{z\} \); 
9 end 
10 end 
11 \( \text{BadBis} := \text{BadBis} \setminus \{y\} \); 
12 \( \text{Bad} := \text{Bad} \cup \{y}\); 
13 end 
14 output: \( (T, \text{Bad}) \)
```
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 $(x_p, x_q, x_c) \in R$ if and only if $x_p = x_c$. It is readily seen that $R$ 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 $H_{r, q, \mu}$ of $C^{**}$ is the natural inclusion from $X_{source}(T)$ to $X$, the maximal 0–bisimulation relation $R^*$ between $C^{**}$ and itself is the identity relation, i.e. $R^* = \{ (x_1, x_2) \in X_{source}(T) \times X_{source}(T) : x_1 = x_2 \}$. Since $R^*$ is the identity relation, the quotient of $C^{**}$ induced by $R^*$, 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 $O(\text{card}[X_p] | X_q|)$.

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:

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

and a specification, expressed by the following nonlinear system:

$$Q: \begin{cases} \dot{x}_1 = -2x_1 + x_2^2, \\ \dot{x}_2 = -4 \sin x_2 + x_2^2 \end{cases}.$$
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 → B or simply by the natural inclusion map taking any a ∈ A to (a) = a ∈ B. Given a function f : A → B the symbol f(A) denotes the image of A through f, i.e. f(A) := {b ∈ B : ∃a ∈ A s.t. b = f(a)}. We identify a relation R ⊆ A × B with the map R : A × B^2 defined by b ∈ R(a) if and only if (a, b) ∈ R. Given a relation R ⊆ A × B, R^(-1) denotes the inverse relation of R, i.e. R^(-1) := {(b, a) ∈ B × A : (a, b) ∈ R}. The symbols N, Z, R, R^+, and R^+_0 denote the set of natural, integer, real, positive real, and nonnegative real numbers, respectively. Given a vector x ∈ 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 : R^n → R^n, the (essential) supremum of f is denoted by ∥f∥_∞: we recall that ∥f∥_∞ = (ess sup {∥f(x)∥, t ≥ 0}); f is essentially bounded if ∥f∥_∞ < ∞. Given x ∈ R^n and ε ∈ R^+, the symbol B_ε(x) denotes the set \{x ∈ R^n : ∥x - x\|_1 ≤ \epsilon\} and the symbol B_{ε_0}(x) denotes the set \{-ε + x_1, x_1 + ε\} × ... × \{-ε + x_n, x_n + ε\} where x_i is the i-th element of x.

It is readily seen that if x ∈ B_{ε_0}(y) and y ∈ B_{ε_0}(z) then x ∈ B_{ε_0}(y+z). For any A ⊆ R^n and μ ∈ R^+, define [A]_μ := \{a ∈ A \mid a_i = k_i, k_i ∈ Z, i = 1, 2, ..., n\}. The set [A]_μ is used as an approximation of the set A with precision μ/2. A function f : [a, b] → R^n is said to be absolutely continuous on [a, b] if for any ε ∈ R^+ there exists δ ∈ R^n so that for every k ∈ N and for every sequence of points a ≤ a_1 < a_2 < ... < a_k < b_k ≤ b, if \sum_{i=1}^k (a_k - a_{k-1}) < δ then \sum_{i=1}^k ∥f(a_i) - f(a_{i-1})∥ < ε. A function f : [a, b] → 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 t ∈ R^+, define f_t so that f_t(t) = f(t), for any t ∈ [0, τ], and f(t) = 0 elsewhere; f is said to be locally essentially bounded if for any t ∈ R^+, f_t is essentially bounded. A continuous function γ : R^n → R^+, is said to belong to class K if it is strictly increasing and γ(0) = 0; γ is said to belong to class K_∞ if γ ∈ K and γ(r) → ∞ as r → ∞. A continuous function β : R^n → R^+_0 is said to belong to class K_L if, for each fixed s, the map β(r, s) → 0 as r → ∞. References


