Integrated design of symbolic controllers for nonlinear systems

Giordano Pola, *Member, IEEE,* Alessandro Borri, *Member, IEEE,* and Maria D. Di Benedetto, *Fellow, IEEE*

***Abstract*—Symbolic models of continuous and hybrid systems have been studied for a long time, because they provide a formal approach to solve control problems where software and hardware interact with the physical world. While being powerful, this approach often encounters some limitations in concrete applications, because of the large size of the symbolic models needed to be constructed. Inspired by on–the–fly techniques for verification and control of finite state machines, in this note we propose an algorithm that integrates the construction of the symbolic models with the design of the symbolic controllers. Computational complexity of the proposed algorithm is discussed and an illustrative example is included.**

***Index Terms*—Symbolic models, approximate bisimulation, digital control systems, nonlinear systems, on–the–fly design.**

I. Introduction

Symbolic models of continuous and hybrid systems have been studied for a long time, because they provide a formal approach to solve control problems where software and hardware interact with the physical world. Symbolic models are abstract descriptions of control systems in which a symbolic state corresponds to an aggregate of states. Several classes of dynamical and control systems that admit symbolic models were identified during the last few years, see e.g. [1], [12] and the references therein. In particular, incrementally stable [2] nonlinear control systems were shown in [7], [10] to admit symbolic models. This last result has been further generalized to incrementally stable nonlinear switched systems in [6], incrementally stable nonlinear time–delay systems in [8], [9] and incrementally forward complete nonlinear control systems in [15]. The use of symbolic models for the control design of continuous and hybrid systems has been investigated in [11], [14]. As discussed in [12], this approach provides the designer with a systematic method to address a wide spectrum of novel specifications, that are difficult to enforce by means of conventional control design paradigms. Examples of such specifications include logic specifications expressed in terms of linear temporal logic formulae or automata on infinite strings. The use of these specifications has been shown to be relevant in the control design of important domains of application, including robot motion planning and systems biology (see e.g. [14] and the references therein). While being powerful, this approach often encounters some limitations in concrete applications, because of the large size of the symbolic models needed to be constructed. In this note we propose one approach to cope with this drawback. We consider a symbolic control design problem for nonlinear control systems. Given a nonlinear control plant and a specification expressed in terms of a finite automaton on infinite strings, we face the problem of designing a symbolic controller that implements the specification with arbitrarily good accuracy. The symbolic controller is furthermore requested to avoid blocking behaviors, when interacting with the plant. This problem can be viewed as an approximate version of similarity games, as discussed in [12]. Related control design problems have been studied in [11] and [14]. The first contribution of this note lies in the derivation of an explicit solution to the control problem under study. *The symbolic controller is proven to be the non–blocking part [3] of the approximate parallel composition [12] between the specification automaton and the symbolic model of the plant*. The synthesis of such a controller requires the preliminary construction of the symbolic model of the plant, which is generally demanding from the computational complexity point of view. Inspired by the research line on on–the–fly verification and control of finite state machines (see e.g. [4], [13]), we give the second contribution of this note consisting in *an efficient algorithm that integrates the construction of the symbolic model of the plant with the design of the symbolic controller*. Computational complexity of the proposed algorithm is discussed and an illustrative example is included.

The authors are with the Department of Electrical and Information Engineering, Center of Excellence for Research DEWS, University of L'Aquila, Via G. Gronchi, 67040, L'Aquila, Italy, giordano.pola,alessandro.borri,mariadomenica.dibenedetto@univaq.it.

This work has been partially supported by European Commission under STREP project HYCON2, and by the Center of Excellence for Research DEWS, University of L'Aquila, Italy.

Manuscript received ???,; revised ???, .

II. Preliminary Definitions

A. Notation

The symbol $|A|$ denotes the cardinality of a finite set A. The identity map on a set A is denoted by 1_A. Given a relation $\mathcal{R} \subseteq A \times B$, the symbol $\mathcal{R}^{-1}$ denotes the inverse relation of $\mathcal{R}$, i.e. $\mathcal{R}^{-1} = \{(b,a) \in B \times A : (a,b) \in A \times B\}$. The symbols $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{R}_0^+$ denote the set of integer, real, positive real, and nonnegative real numbers, respectively. The symbol $\|x\|$ denotes the infinity norm of $x \in \mathbb{R}^n$. Given a measurable function $f : \mathbb{R}_0^+ \to \mathbb{R}^n$, the (essential) supremum of f is denoted by $\|f\|_\infty$. Given $x \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}^+$, the symbols $\mathcal{B}_\varepsilon(x)$ and $\mathcal{B}_{[\varepsilon[}(x)$ denote the set $\{x \in \mathbb{R}^n \,|\, \|x\| \leq \varepsilon\}$ and the set $[-\varepsilon + x_1, x_1 + \varepsilon[\times[-\varepsilon + x_2, x_2 + \varepsilon[\times ... \times [-\varepsilon + x_n, x_n + \varepsilon[$, respectively. Given $\mu \in \mathbb{R}^+$ and $A \subseteq \mathbb{R}^n$, we denote by μA the set $\{b \in \mathbb{R}^n \,|\, \exists a \in A \text{ s.t. } b = \mu a\}$. For any $x \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^+$ the symbol $[x]_\mu$ denotes the unique vector in $\mu\,\mathbb{Z}^n$ so that $x \in \mathcal{B}_{[\mu/2[}([x]_\mu)$.

B. Control Systems

In this note we consider the nonlinear control system:

$$\Sigma : \begin{cases} \dot{x}(t) = f(x(t), u(t)), t \in \mathbb{R}_0^+, \\ x(0) \in X_0, \end{cases} \quad (1)$$

where $x(t) \in X \subseteq \mathbb{R}^n$ is the state at time t, u belongs to the set $\mathcal{U}$ of locally essentially bounded functions of time from intervals of the form $]a, b[\subseteq \mathbb{R}$ to $U \subseteq \mathbb{R}^m$, $X_0 \subseteq X$ is the set of initial states and $f : \mathbb{R}^n \times U \to \mathbb{R}^n$ is Lipschitz on compact sets. In the sequel we refer to the nonlinear control system Σ in (1) by means of the tuple $\Sigma = (X, X_0, U, \mathcal{U}, f)$, where each entity has been defined above. A curve $\xi :]a, b[\to \mathbb{R}^n$ is 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 $\xi = \xi'|_{[0,\tau]}$. In the sequel we write $\xi_{xu}(\tau)$ to denote the point reached at time τ under the input u from the initial condition x. A control system Σ is forward complete if every trajectory is defined on an interval of the form $]a, \infty[$. In this note we make use of the following stability notion.

Definition 1: [2] 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:

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

C. Symbolic Systems, Approximate Bisimulation and Composition

We start by recalling the notion of systems that have been introduced in [12], as a unified mathematical framework to describe control systems as well as their symbolic models.

Definition 2: [12] A system S is a sextuple $(X, X_0, U, \longrightarrow, Y, H)$ 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, and an output function $H : X \to Y$. A transition $(x, u, x') \in \longrightarrow$ 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 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 state $x' \in X$ such that $x \xrightarrow{u} x'$, *non–blocking*, if for any $x \in X$ there exists at least one state $x' \in X$ such that $x \xrightarrow{u} x'$, *accessible*, if for any $x \in X$ there exists a finite number of transitions $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \ldots \xrightarrow{u_N} x$ from an initial state $x_0 \in X_0$ to state x.

For a detailed description of the notion of system and of its properties we refer to [12]. As done in [11], [7], [10], in this note we consider the notions of approximate simulation and bisimulation relations to relate properties of control systems and symbolic systems.

Definition 3: [5] Let $S_i = (X_i, X_{0,i}, U_i, \xrightarrow[i]{}, Y_i, H_i)$ ($i = 1, 2$) 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 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 $(x_1, x_2) \in \mathcal{R}$ existence of $x_1 \xrightarrow[1]{u_1} x_1'$ in S_1 implies 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–simulated by S_2 or S_2 ε–simulates S_1, denoted $S_1 \preceq_\varepsilon S_2$, if there exists an ε–approximate simulation relation from S_1 to S_2. Relation $\mathcal{R}$ is an ε–approximate bisimulation relation between S_1 and S_2 if $\mathcal{R}$ is an ε–approximate simulation relation from S_1 to S_2 and $\mathcal{R}^{-1}$ is an ε–approximate simulation relation from S_2 to S_1. Furthermore, systems S_1 and S_2 are ε–bisimilar, denoted $S_1 \cong_\varepsilon S_2$, if there exists an ε–approximate bisimulation relation $\mathcal{R}$ between S_1 and S_2. When $\varepsilon = 0$ systems S_1 and S_2 are said to be exactly bisimilar.

Lemma 1: [5] If $S_1 \cong_{\varepsilon_{12}} S_2$ and $S_2 \cong_{\varepsilon_{23}} S_3$ then $S_1 \cong_{\varepsilon_{12}+\varepsilon_{23}} S_3$.

Lemma 2: [5] For any $\varepsilon_1 \leq \varepsilon_2$, $S_1 \cong_{\varepsilon_1} S_2$ implies $S_1 \cong_{\varepsilon_2} S_2$.

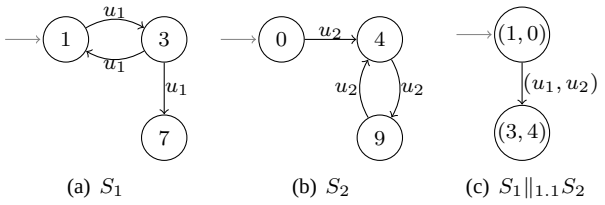
In the sequel we make use of the notion of approximate parallel composition introduced in [11], to capture (feedback) interaction between control systems and symbolic controllers.

Definition 4: [11] Let $S_i = (X_i, X_{0,i}, U_i, \xrightarrow[i]{}, Y_i, H_i)$ ($i = 1, 2$) be metric systems with the same output sets $Y_1 = Y_2$ and metric d and a precision $\theta \in \mathbb{R}_0^+$. The θ–approximate parallel composition of S_1 and S_2 is the system $S_1 \|_\theta S_2 = (X, X_0, U, \longrightarrow, Y, H)$, where $X = \{(x_1, x_2) \in X_1 \times X_2 \mid d(H_1(x_1), H_2(x_2)) \leq \theta\}$, $X_0 = X \cap (X_{0,1} \times X_{0,2})$, $U = U_1 \times U_2$, $(x_1, x_2) \xrightarrow{(u_1,u_2)} (x_1', x_2')$ if $x_1 \xrightarrow[1]{u_1} x_1'$ and $x_2 \xrightarrow[2]{u_2} x_2'$, $Y = Y_1$, and $H(x_1, x_2) = H_1(x_1)$ for any $(x_1, x_2) \in X$.

Lemma 3: For any $\theta \in \mathbb{R}_0^+$, $S_1 \|_\theta S_2 \preceq_\theta S_2$.

The interested reader is referred to [11], [12] for a detailed description of this notion of composition and of its properties. We conclude this section with the following example.

Example 1: Consider two metric systems $S_i = (X_i, X_{0,i}, U_i, \xrightarrow[i]{}, Y_i, H_i)$ ($i = 1, 2$), where $X_1 = \{1, 3, 7\} \subseteq \mathbb{R}$, $X_{0,1} = \{1\}$, $U_1 = \{u_1\}$, $1 \xrightarrow[1]{u_1} 3$, $3 \xrightarrow[1]{u_1} 1$, $3 \xrightarrow[1]{u_1} 7$, $Y_1 = X_1$ and $H_1(x) = x$ for any $x \in X_1$; $X_2 = \{0, 4, 9\} \subseteq \mathbb{R}$, $X_{0,2} = \{0\}$, $U_2 = \{u_2\}$, $0 \xrightarrow[2]{u_2} 4$, $4 \xrightarrow[2]{u_2} 9$, $9 \xrightarrow[2]{u_2} 4$, $Y_2 = X_2$ and $H_2(x) = x$ for any $x \in X_2$. Consider a precision $\varepsilon = 1.1$. By Definition 4, metric system $S_1 \|_\varepsilon S_2 = (X^\varepsilon, X_0^\varepsilon, U^\varepsilon, \xrightarrow[\varepsilon]{}, Y^\varepsilon, H^\varepsilon)$ ($\varepsilon = 1.1$) is given by $X^{1.1} = \{(1, 0), (3, 4)\}$, $X_0^{1.1} = \{(1, 0)\}$, $U^{1.1} = \{(u_1, u_2)\}$, $(1, 0) \xrightarrow[1.1]{(u_1,u_2)} (3, 4)$, $Y^{1.1} = Y_1$, $H^{1.1}(1, 0) = 1$ and $H^{1.1}(3, 4) = 3$. Systems S_1, S_2 and $S_1 \|_{1.1} S_2$ are depicted in Figure 1.



(a) S_1 (b) S_2 (c) $S_1 \|_{1.1} S_2$

Fig. 1. Metric systems S_1 and S_2 and their approximate parallel composition $S_1 \|_{1.1} S_2$, as in Example 1.

III. Symbolic Control Design of Nonlinear Systems

Given the nonlinear control system $\Sigma = (X, X_0, U, \mathcal{U}, f)$ and a sampling time $\tau \in \mathbb{R}^+$, consider the system $S_\tau(\Sigma) = (X, X_0, \mathcal{U}_\tau, \underset{\tau}{\longrightarrow}, Y, H)$, where $\mathcal{U}_\tau$ denotes the set of constant functions from $[0, \tau]$ to U, $x \xrightarrow[\tau]{u} x'$ if there exists a trajectory $\xi : [0, \tau] \to X$ of Σ satisfying $\xi_{xu}(\tau) = x'$, $Y = X$, and $H = 1_X$. System $S_\tau(\Sigma)$ is metric when we regard $Y = X$ as being equipped with the metric $d(p, q) = \|p - q\|$. System $S_\tau(\Sigma)$ can be thought of as the time discretization of the control system Σ. We can now state the control design problem focused on in this note.

Problem 1: Given a plant nonlinear control system $P = (X^p, X_0^p, U^p, \mathcal{U}^p, f^p)$, consider a specification described by a deterministic symbolic system $Q = (X^q, X_0^q, U^q, \underset{q}{\longrightarrow}, Y^q, H^q)$, where X^q is a finite subset of $\mathbb{R}^n$, $X_0^q \subseteq X^q$, $U^q = \{u^q\}$, $\underset{q}{\longrightarrow} \subseteq X^q \times U^q \times X^q$, $Y^q = X^q$, and $H^q = 1_{X^q}$. For any desired precision $\varepsilon \in \mathbb{R}^+$, find a sampling time parameter $\tau \in \mathbb{R}^+$, a parameter $\theta \in \mathbb{R}^+$ and a (controller) symbolic system C so that the approximate parallel composition $S_\tau(P) \|_\theta C$ is non–blocking and $(S_\tau(P) \|_\theta C) \preceq_\varepsilon Q$.

This problem asks for the synthesis of a symbolic controller C so that the controlled plant is non–blocking and implements the specification Q with arbitrarily good accuracy. This problem can be viewed as an approximate version of similarity games, as discussed in [12]. The solution to Problem 1 requires some preliminary notions and results, which are recalled hereafter. Given two systems $S_i = (X_i, X_{0,i}, U_i, \underset{i}{\longrightarrow}, Y_i, H_i)$ $(i = 1, 2)$, S_1 is a *sub–system* of S_2, denoted $S_1 \sqsubseteq S_2$, if $X_1 \subseteq X_2$, $X_{0,1} \subseteq X_{0,2}$, $U_1 \subseteq U_2$, $\underset{1}{\longrightarrow} \subseteq \underset{2}{\longrightarrow}$, $Y_1 \subseteq Y_2$ and $H_1(x) = H_2(x)$ for any $x \in X_1$. The *non–blocking part* of a system S is the unique non–blocking system $Nb(S)$ so that $S' \sqsubseteq Nb(S) \sqsubseteq S$, for any non–blocking system $S' \sqsubseteq S$.

Definition 5: Given the plant P, a sampling time $\tau \in \mathbb{R}^+$, a state quantization $\eta \in \mathbb{R}^+$ and an input quantization $\mu \in \mathbb{R}^+$ consider the following system:

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

where $X_{\tau\eta\mu} = 2\eta\mathbb{Z}^n \cap X^p$, $X_{0,\tau\eta\mu} = X_{\tau\eta\mu} \cap X_0^p$, $U_{\tau\eta\mu} = 2\mu\mathbb{Z}^m \cap U^p$, $x \xrightarrow[\tau\eta\mu]{u} y$ if $\xi_{xu}(\tau) \in \mathcal{B}_{[\eta[}(y) \cap X^p$, $Y_{\tau\eta\mu} = X^p$, and $H_{\tau\eta\mu}(x) = x$ for any $x \in X_{\tau\eta\mu}$.

It is readily seen that $S_{\tau\eta\mu}(P)$ is deterministic, countable, and it becomes symbolic when X^p and U^p are bounded sets. The following result relates systems $S_\tau(P)$ and $S_{\tau\eta\mu}(P)$.

Theorem 1: Suppose that P is δ–ISS and consider a desired precision $\varepsilon \in \mathbb{R}^+$. For any $\tau, \eta, \mu \in \mathbb{R}^+$ satisfying the inequality $\beta(\varepsilon, \tau) + \gamma(\mu) + \eta \leq \varepsilon$, systems $S_{\tau\eta\mu}(P)$ and $S_\tau(P)$ are ε–bisimilar.

The proof of the above result is similar to the one of Theorem 5.1 in [7] and is therefore omitted. We now have all the ingredients to give the solution to Problem 1.

Theorem 2: Suppose that P is δ–ISS and consider any precision $\varepsilon \in \mathbb{R}^+$. Choose quantization parameters $\tau, \theta, \eta, \mu \in \mathbb{R}^+$ satisfying the following inequalities:

$$\beta(\theta, \tau) + \gamma(\mu) + 2\eta \leq \theta + \eta \leq \varepsilon, \tag{2}$$

Then, the non–blocking part $Nb(C^*)$ of the controller $C^* = S_{\tau\eta\mu}(P) \|_\eta Q$ solves Problem 1.

Proof: We first prove $(S_\tau(P) \|_\theta Nb(C^*)) \preceq_\varepsilon Q$. By Lemma 3, $(S_\tau(P) \|_\theta Nb(C^*)) \preceq_\theta Nb(C^*)$ and $(S_{\tau\eta\mu}(P) \|_\eta Q) \preceq_\eta Q$. Since $Nb(C^*)$ is a sub–system of C^* then $Nb(C^*) \preceq_0 C^* = S_{\tau\eta\mu}(P) \|_\eta Q$. By Lemmas 1 and 2, and since by (2), $\theta + \eta \leq \varepsilon$, the above approximate similarity inclusions imply $(S_\tau(P) \|_\theta Nb(C^*)) \preceq_\varepsilon Q$, which concludes the proof. We now prove that $S_\tau(P) \|_\theta Nb(C^*)$ is non–blocking. Consider any state (p_1, p_2, q) of $S_\tau(P) \|_\theta Nb(C^*)$. Since $Nb(C^*)$ is non–blocking, for the state (p_2, q) of $Nb(C^*)$ there exists a state (p_2^+, q^+) of $Nb(C^*)$ so that $(p_2, q) \xrightarrow{(u_2, u_3)} (p_2^+, q^+)$ is a transition of $Nb(C^*)$. Since by the first inequality in (2) and Theorem 1, $S_\tau(P)$ and $S_{\tau\eta\mu}(P)$ are θ–bisimilar, for the transition $p_2 \xrightarrow{u_2} p_2^+$ in $S_{\tau\eta\mu}(P)$ there exists a transition $p_1 \xrightarrow{u_1} p_1^+$ in $S_\tau(P)$ so that $d(H^p(p_1^+), H^p(p_2^+)) \leq \theta$. This implies from Definition 4 that (p_1^+, p_2^+, q^+) is a state of $S_\tau(P) \|_\theta Nb(C^*)$ and therefore that $(p_1, p_2, q) \xrightarrow{(u_1, (u_2, u_3))} (p_1^+, p_2^+, q^+)$ is a transition of $S_\tau(P) \|_\theta Nb(C^*)$, which concludes the proof. ■

IV. Integrated Symbolic Control Design

The construction of the symbolic controller $Nb(C^*)$ relies upon the procedure illustrated in Algorithm 1.

1 Compute the system $S_{\tau\eta\mu}(P)$;
2 Compute the composition $S_{\tau\eta\mu}(P) \|_\eta Q$;
3 Compute the non–blocking part $Nb(S_{\tau\eta\mu}(P) \|_\eta Q)$ of $S_{\tau\eta\mu}(P) \|_\eta Q$.

Algorithm 1: Construction of the controller $Nb(C^*)$.

Remark 1: This procedure is not efficient from the computational complexity point of view, because:

(i) It requires the preliminary construction of the symbolic system $S_{\tau\eta\mu}(P)$ of the plant P, before designing the symbolic controller $Nb(C^*)$. A more efficient algorithm would compute the symbolic controller in conjunction with the symbolic system of the plant.
(ii) 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.

Current work on symbolic control design of continuous and hybrid systems, see e.g. [14], [11], exhibit the same drawback as in (i), since they first construct the symbolic model of the plant to then derive the symbolic controller.

In order to cope with the aforementioned drawbacks, we now present a procedure that *integrates each step of Algorithm 1 in one algorithm*. The proposed procedure is reported in Algorithm 2 and Algorithm 3. Algorithm 2 is the main one while Algorithm 3 introduces function **NonBlock** that is used in Algorithm 2. The outcome of Algorithm 2 is the symbolic controller C^{**}. In the sequel, line i of Algorithm j will be recalled as line $j.i$. 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. $\mathbf{X}_{source}(T) = \{x \in X : \exists y \in Y \wedge \exists u \in U$ s.t. $(x,u,y) \in T\}$. Algorithm 2 proceeds as follows. In line 2.2 the set X_0 of initial states of C^{**} is initialized to the set $\{x_p \in X_{0,\tau\eta\mu} \,|\, \exists x_q \in X^q$ s.t. $\|x_p - x_q\| \leq \eta\}$, the set $\mathbf{X}_{target}$ of to–be–processed states to X_0, the set T of transitions of C^{**} and the set Bad of blocking states to the empty sets. At each basic step, Algorithm 2 processes a (non–processed) state x in line 2.4. It considers a transition $v \xrightarrow[q]{u^q} y$ in Q with source state $v \in X^q$ so that $\|x - v\| \leq \eta$ and target state y for which $[y]_{2\eta} \notin Bad$ (line 2.5) and it searches for a control input $u \in U_{\tau\eta\mu}$ by which the plant P meets the specification Q, i.e. $\|z - y\| \leq \eta$ (line 2.10) where $z = [\xi_{xu}(\tau)]_{2\eta}$ (line 2.9). If such a control input u exists, then boolean variable $Flag$ is updated to 1 (line 2.11), the transition (x,u,z) is added to T (line 2.15), and the state z is added to $\mathbf{X}_{target}$ (line 2.16). If either state y is blocking or no input is found for the plant P to meet the specification Q (line 2.19), then state x is declared blocking and function **NonBlock**(T, x, Bad) is called (line 2.20). Algorithm 2 proceeds with further basic steps, until there are no more states to be processed. When Algorithm 2 terminates, it returns the symbolic controller C^{**} in line 2.24. Function **NonBlock** extracts the non–blocking part from T. The set $Badx$ is initialized to $\{x\}$ (line 3.2). At each basic step, for any $y \in Badx$, function **NonBlock** removes from T all transitions (z,u,y) with target state y (line 3.6), it adds z to $Badx$ (line 3.7) and it moves y from $Badx$ to Bad (lines 3.10 and 3.11). Function **NonBlock** terminates when there are no more states to be processed and it returns in line 3.13 the updated sets T and Bad. Termination of this procedure 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 every processed state x, depending on the value of the boolean variable $Flag$, either line 2.15 or line 2.20 is executed; this ensures by line 2.4 that state x in no longer processed in future iterations. Furthermore, the set $\mathbf{X}_{target}$ is nondecreasing (with respect to the set inclusion operator $\subseteq$), see line 2.16, and contained in the finite set $\{x_p \in X_{\tau\eta\mu} \,|\, \exists x_q \in X^q$ s.t. $\|x_p - x_q\| \leq \eta\}$. Hence, provided that Algorithm 3 terminates in finite time, the result follows. Regarding termination of Algorithm 3, in the worst case the set Bad coincides with the intersection of the sets of accessible states of $S_{\tau\eta\mu}(P)$ and Q (line 3.11) and the set $Badx$ becomes empty (line 3.10). Hence, finite termination of Algorithm 3 is guaranteed by line 3.3. ■

We now show that the controller C^{**}, synthesized in Algorithm 2, solves Problem 1.

```
1  Input: plant P = (X^p, X_0^p, U^p, U^p, f^p), specification
   Q = (X^q, X_0^q, U^q, --q-->, Y^q, H^q), precision
   ε ∈ R^+, quantization parameters τ, η, μ, θ ∈ R^+
   satisfying the inequalities in (2);
2  Init:
   X_0 := {x_p ∈ X_{0,τημ} | ∃x_q ∈ X^q s.t. ||x_p − x_q|| ≤ η},
   X_target := X_0, T := ∅, Bad := ∅;
3  foreach
   x ∈ {x_p ∈ X_{τημ} | ∃x_q ∈ X^q s.t. ||x_p − x_q|| ≤ η} do
4      if x ∈ X_target \ (X_source(T) ∪ Bad) then
5          if [ ∃v ∈ X^q s.t ||x − v|| ≤
           η ] ∧ [ v --u^q/q--> y ] ∧ [ [y]_{2η} ∉ Bad ] then
6              Flag := 0;
7              while Flag = 0 do
8                  choose u ∈ U_{τημ};
9                  compute z = [ξ_{xu}(τ)]_{2η};
10                 if ||z − y|| ≤ η then
11                     Flag := 1;
12                 end
13             end
14             if Flag = 1 then
15                 T := T ∪ {(x, u, z)};
16                 X_target := X_target ∪ {z};
17             end
18         end
19         if [ Flag = 0 ] ∨ [ y ∈ Bad ] then
20             (T, Bad) :=NonBlock(T, x, Bad);
21         end
22     end
23 end
24 output: C** =
   (X_source(T), X_0 ∩ X_source(T), U_{τημ}, T, Y_{τημ}, H_{τημ})
```

Algorithm 2: Integrated Symbolic Control Design.

```
1  Function (T, Bad) :=NonBlock(T, x, Bad);
2  Init: Badx := {x};
3  foreach y ∈ Badx do
4      foreach z ∈ X_source(T) do
5          if [ ∃u ∈ U_{τημ} s.t. (z, u, y) ∈ T ] then
6              T := T \ {(z, u, y)};
7              Badx := Badx ∪ {z};
8          end
9      end
10     Badx := Badx \ {y};
11     Bad := Bad ∪ {y};
12 end
13 output: (T, Bad)
```

Algorithm 3: Non–blocking Algorithm.

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

Proof: Let be $Nb(C^*) = (X_*, X_{0,*}, U_*, \xrightarrow[*]{}, Y_*, H_*)$ and $C^{**} = (X_{**}, X_{0,**}, U_{**}, \xrightarrow[**]{}, Y_{**}, H_{**})$. Consider the relation $\mathcal{R} \subseteq X_* \times X_{**}$ defined by $((x_p, x_q), x_c) \in \mathcal{R}$ if and only if $x_p = x_c$. In the following we only show that $\mathcal{R}$ is a 0–approximate simulation relation from $Nb(C^*)$ to C^{**}. By using similar arguments it is possible to show that $\mathcal{R}^{-1}$ is a 0–approximate simulation relation from C^{**} to $Nb(C^*)$. Consider any $(x_p, x_q) \in X_{0,*}$. Choose $x_c = x_p$. By lines 2.2 and 2.24, $x_c \in X_{0,**}$. Hence, condition (i) in Definition 3 is satisfied. Consider any $((x_p, x_q), x_c) \in \mathcal{R}$. By Definition 4, $H_*(x_p, x_q) = H_{\tau\eta\mu}(x_p)$ and since $x_p = x_c$ and $H_{\tau\eta\mu}$ and H_{**} are the identity functions, $H_{\tau\eta\mu}(x_p) = H_{**}(x_c)$ from which, $H_*(x_p, x_q) = H_{**}(x_c)$. Hence, condition (ii) in Definition 3 is satisfied. Consider any transition $(x_p, x_q) \xrightarrow[*]{(u_p, u_q)} (x_p^+, x_q^+)$. By definition of $Nb(C^*)$, $x_p \xrightarrow[\tau\eta\mu]{u_p} x_p^+$ is a transition of $S_{\tau\eta\mu}(P)$ and $x_q \xrightarrow[q]{u_q} x_q^+$ is a transition of Q. By lines 2.5, 2.9, 2.10 and 2.15, there exists a transition $x_p \xrightarrow[**]{u_p'} x_p^+$ in C^{**}, with $((x_p^+, x_q^+), x_p^+) \in \mathcal{R}$. Hence, condition (iii) in Definition 3 is satisfied, which concludes the proof. ∎

As a consequence of the above result, $Nb(C^*)$ solves Problem 1 if and only if C^{**} solves Problem 1. Hence, it shows that Algorithm 2 is correct. Before presenting the last result of this section, we recall that the minimal exactly bisimilar system of a system S is the system with the smallest number of states, which is bisimilar to S.

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

Proof: From Definition 3, the minimal exactly bisimilar system of an accessible system S with injective output map coincides with S. By exploring Algorithm 2, the controller C^{**} is accessible and the output map of C^{**} is injective. Hence, C^{**} is the minimal exactly bisimilar system of itself. Since by Theorem 4, C^{**} and $Nb(C^*)$ are exactly bisimilar, the result follows. ∎

V. Computational Complexity Analysis

In this section we provide a formal comparison between Algorithms 1 and 2 in terms of space and time complexity. We start by discussing space complexity.

Proposition 1: Space complexity of Algorithm 1 is $O(\max\{|X_{\tau\eta\mu}| \cdot |U_{\tau\eta\mu}|, |X^q|\})$.

Proof: Since $S_{\tau\eta\mu}(P)$ is deterministic, transitions of $S_{\tau\eta\mu}(P)$ are at most $|X_{\tau\eta\mu}| \cdot |U_{\tau\eta\mu}|$. For the same reason, transitions of Q are at most $|X^q|$. By Definition 4, transitions of C^* are at most $\min\{|X^q|, |X_{\tau\eta\mu}|\} \cdot |U_{\tau\eta\mu}|$. Moreover, transitions of $Nb(C^*)$ are less than or equal to the ones of C^*. Hence, the result follows by comparing the above worst case bounds. ∎

Proposition 2: Space complexity of Algorithm 2 is $O(\min\{|X_{\tau\eta\mu}|, |X^q|\})$.

Proof: Any triplet (x, u, y) is added to T only if (x, u, y) is a transition of $S_{\tau\eta\mu}(P)$ and if there exists a pair of states $x', y' \in X^q$ so that $\|x - x'\| \leq \eta$, $\|y - y'\| \leq \eta$ and (x, u^q, y) is a transition of Q. Hence, the result follows from determinism of $S_{\tau\eta\mu}(P)$ and Q. ∎

By comparing Propositions 1 and 2, space complexity of Algorithm 2 is smaller than or equal to space complexity of Algorithm 1. Finally, time complexity is analyzed hereafter.

Proposition 3: Time complexity of Algorithm 1 is $O(|X^q| \cdot |X_{\tau\eta\mu}| \cdot |U_{\tau\eta\mu}|)$.

Proof: The number of steps needed in the construction of C^* is given by $|X^q| \cdot |X_{\tau\eta\mu}| \cdot |U_{\tau\eta\mu}|$. Regarding the computation of $Nb(C^*)$, for any state in the set $X_c = \{x_p \in X_{\tau\eta\mu} \,|\, \exists x_q \in X^q \text{ s.t. } \|x_p - x_q\| \leq \eta\}$ of states of C^*, in the worst case all transitions in C^* are needed to be processed. Since $|X_c| \leq \min\{|X^q|, |X_{\tau\eta\mu}|\}$ and transitions of C^* are at most $\min\{|X^q|, |X_{\tau\eta\mu}|\} \cdot |U_{\tau\eta\mu}|$, the number of steps needed in the computation of $Nb(C^*)$ is given by $(\min\{|X_{\tau\eta\mu}|, |X^q|\})^2 \; |U_{\tau\eta\mu}|$. By comparing the above worst case bounds, the result follows. ∎

Proposition 4: Time complexity of Algorithm 2 is $O(\min\{|X_{\tau\eta\mu}|, |X^q|\}^2 |U_{\tau\eta\mu}|)$.

Proof: By exploring Algorithm 2, the number of steps needed in the computation of C^{**} is upper bounded by $\sum_{i=0}^{N_1}(N_2 + N_3)$, where $N_1 = \min\{|X_{\tau\eta\mu}|, |X^q|\}$, N_2 is an upper bound to the number of steps needed in the execution of lines 2.4–2.18 and N_3 is an upper bound to the number of steps needed in the execution of lines 2.19–2.22. The term $\sum_{i=0}^{N_1} N_2$ is upper bounded by $\min\{|X_{\tau\eta\mu}|, |X^q|\}|U_{\tau\eta\mu}|$. Regarding the term $\sum_{i=0}^{N_1} N_3$, whenever line 2.20 is executed, processed states x are different. Indeed, suppose that at step i state x is processed in line 2.20. When Algorithm 3 is called, state x is added to Bad (see lines 3.2, 3.3 and 3.11). Since at the end of step i state $x \in Bad$, such a state is no longer processed in future iterations (see line 2.4). Since any time Algorithm 3 is called, it processes different states, the overall time complexity due to the term $\sum_{i=0}^{N_1} N_3$ is upper bounded by the time complexity needed in computing the non–blocking part of C^* which, from the proof of Proposition 3, is $(\min\{|X_{\tau\eta\mu}|, |X^q|\})^2 |U_{\tau\eta\mu}|$. By comparing the above worst case bounds, the result follows. ∎

By comparing Propositions 3 and 4, time complexity of Algorithm 2 is smaller than or equal to time complexity of Algorithm 1.

VI. An illustrative example

Consider a nonlinear control plant described by the following equations:

$$P : \begin{cases} \dot{x}_1 = -4x_1 + x_2^2 - u, \\ \dot{x}_2 = 2x_1 - 7\sin x_2, \end{cases} \tag{3}$$

with $X^p = [-1, 1[\times[-1, 1[$, $X_0^p = [-0.5, 0.5[\times[-0.5, 0.5[$ and $u \in U = [-5, 5]$. Consider a specification Q given in terms of the following periodic trajectory:

$$(-0.5, -0.65) \longrightarrow (-0.5, -1) \longrightarrow (-0.3, -0.8) \longrightarrow (0.2, -0.1) \longrightarrow (0.6, 0.6) \longrightarrow (0.4, 0.8) \longrightarrow (0.2, 0.65) \longrightarrow (-0.3, 0) \longrightarrow (-0.5, -0.65). \tag{4}$$

Enforcing periodic trajectories is a rather frequent specification when controlling some kind of systems, as for example

industrial robots that are often requested to perform repetitive tasks. We now design a symbolic controller which enforces the specification Q on the plant P within a chosen precision $\varepsilon = 0.2$. To this aim we apply Theorem 2. By using the Lyapunov characterization of δ–ISS reported in [2], it is possible to show that the plant system P is δ–ISS with functions $\beta(r,s) = \sqrt{2}\, e^{-2\,s}\, r$, $\gamma(r) = \sqrt{0.8\, r}$, $r, s \in \mathbb{R}_0^+$. For the precision $\varepsilon = 0.2$, it is easy to see that the quantization parameters $\theta = 0.16$, $\eta = 0.02$, $\tau = 1$, $\mu = 0.01$ fulfill the inequalities in (2). By running Algorithm 2, the symbolic controller C^{**} has been designed, and reported hereafter:

$$(-0.48,-0.64) \xrightarrow{2.88} (-0.48,-1) \xrightarrow{1.89} (-0.28,-0.8) \xrightarrow{-0.82} (0.2,-0.08) \xrightarrow{-2.2} (0.6,0.6) \xrightarrow{-1.04} (0.4,0.8) \xrightarrow{-0.32} (0.2,0.64) \xrightarrow{1.22} (-0.28,0) \xrightarrow{2.26} (-0.48,-0.64). \tag{5}$$

From Theorems 2 and 4 the controller C^{**} enforces the specification Q on P, within the chosen precision $\varepsilon = 0.2$. For example, consider the initial state $x_0 = (-0.5,-0.5)$ of P. From (4), the specification Q requires $x_0 \in \mathcal{B}_\varepsilon(-0.5,-0.65)$. Since $\|x_0 - (-0.5,-0.65)\| = 0.15 < \varepsilon$, the specification Q is satisfied. Since $x_0 \in \mathcal{B}_\theta(-0.48,-0.64)$, the constant control input $u = 2.88$ (labeling the outgoing transition from state $(-0.48,-0.64)$ in (5)) is applied to P for a time duration τ. By integrating the differential equation in (3), we obtain $\xi_{(-0.5,-0.5)\,2.88}(\tau) = (-0.47,-0.9)$. Since $\|(-0.47,-0.9) - (-0.5,-1)\| = 0.1 < \varepsilon$, where $(-0.5,-1)$ is the state required by Q to be reached at time τ, the specification Q is fulfilled. By applying the subsequent control inputs in (5), it is possible to show that the whole specification Q is fulfilled. Figure 2 illustrates the evolution of state variables of P, when applying the designed controller. We conclude this section by discussing the computational complexity needed to construct $Nb(C^*)$ and C^{**}. The memory occupation[1] required to construct $Nb(C^*)$ is $2,759,580$, while the one required to construct C^{**} is 48. The time of computation needed to construct $Nb(C^*)$ is $5,442$ s, while the one needed to construct C^{**} is 13 s.

VII. Conclusion

In this note we proposed an integrated approach to the design of symbolic controllers for nonlinear control systems. The proposed on–the–fly algorithms perform especially well whenever the specifications involve only small regions of the state space of the plant control system (see Remark 1 (ii)); this is in fact the case in many concrete applications as for example robot motion planning or following periodic orbits. In some control design problems, specifications require the states to be reached within specific time intervals. We plan to investigate integrated symbolic control design of nonlinear systems with timed specifications in future work.

Acknowledgment

The first author thanks Paulo Tabuada for fruitful discussions on the topic of the present paper.

[1]The memory occupation in the construction of $Nb(C^*)$ is evaluated as the sum of the number of transitions of $S_{\tau\eta\mu}(P)$, Q and $S_{\tau\eta\mu}(P)\|_\eta Q$, while the one in the construction of C^{**} as the sum of the number of transitions in C^{**} and of states in Bad. Each transition is weighted as three data and each state as one datum.

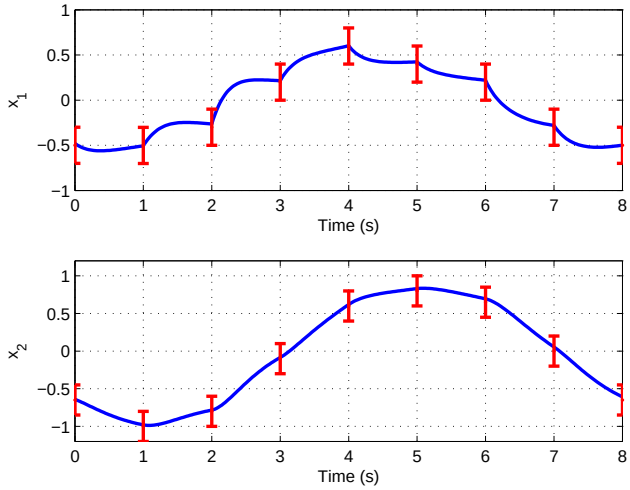


Fig. 2. State trajectory of the plant P controlled by the symbolic controller C^{**}. Vertical intervals represent the precision $\varepsilon = 0.2$.

References

[1] R. Alur, T.A. Henzinger, G. Lafferriere, and G.J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971–984, 2000.

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

[3] C. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Kluwer Academic Publishers, Boston, MA, 1999.

[4] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. Memory-efficient algorithms for the verification of temporal properties. *Formal Methods in System Design*, 1(2-3):275–288, 1992.

[5] A. Girard and G.J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007.

[6] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions of Automatic Control*, 55(1):116–126, January 2010.

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

[8] G. Pola, P. Pepe, M. D. Di Benedetto, and P. Tabuada. Symbolic models for nonlinear time–delay systems using approximate bisimulations. *Systems and Control Letters*, 59:365–373, 2010.

[9] G. Pola, P. Pepe, and M.D. Di Benedetto. Alternating approximately bisimilar symbolic models for nonlinear control systems with unknown time–varying delays. In *49th IEEE Conference on Decision and Control*, pages 7649–7654, Atlanta, USA, December 2010.

[10] G. Pola and P. Tabuada. Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM Journal on Control and Optimization*, 48(2):719–733, 2009.

[11] P. Tabuada. An approximate simulation approach to symbolic control. *IEEE Transactions on Automatic Control*, 53(6):1406–1418, 2008.

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

[13] 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*, volume 1708 of *Lecture Notes in Computer Science*, pages 233 – 252. Springer Verlag, Berlin, September 1999.

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

[15] M. Zamani, M. Mazo Jr., G. Pola, and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions of Automatic Control*, January 2011. Provisionally accepted. arXiv:1002.0822.