Integrated Symbolic Design of Unstable Nonlinear Networked Control Systems

Alessandro Borri, Giordano Pola and Maria Domenica Di Benedetto

***Abstract*— The research area of Networked Control Systems (NCS) has been the topic of intensive study in the last decade. In this paper we give a contribution to this research line by addressing symbolic control design of (possibly unstable) nonlinear NCS with specifications expressed in terms of automata. We first derive symbolic models that are shown to approximate the given NCS in the sense of (alternating) approximate simulation. We then address symbolic control design with specifications expressed in terms of automata. We finally derive efficient algorithms for the synthesis of the proposed symbolic controllers that cope with the inherent computational complexity of the problem at hand.**

I. Introduction

Networked Control Systems (NCS) are complex, heterogeneous, spatially distributed systems where physical processes interact with distributed computing units through non–ideal communication networks. The complexity and heterogeneity of such systems is given by the interaction of at least three components: a plant process that is often described by continuous dynamics, a controller implementing algorithms on microprocessors for the control of the plant, and a communication network conveying information between the plant and the controller which is often characterized by non-idealities such as variable sampling/transmission intervals, variable communication delays, quantization errors, packet dropouts, communication protocol and limited bandwidth. In the last decade, NCS have been the object of great interest in the research community and important results have been achieved, see e.g. [1] and the references therein. Most of the results on NCS mainly deals with stabilization problems under an imperfect communication network comprising a subset of the aforementioned communication non-idealities. The work in [2] instead, considers all the aforementioned communication non-idealities and proposes control algorithms for solving problems with complex specifications expressed in terms of automata. The main drawbacks of the results reported in [2] are:

(i) The plant in the NCS is supposed to be stable, which is quite restrictive in many application domains of interest.

(ii) The controllers proposed require a large computational complexity in their design.

The present work improves the results established in [2] in two directions:

(i') We extend our results to possibly unstable nonlinear networked control systems;

(ii') We design efficient algorithms that cope with the computational complexity of the approach in [2].

For (i') we generalize the results reported in [3] from nonlinear control systems to nonlinear *networked* control systems. For (ii') we generalize the control algorithms we proposed in [4] for stable nonlinear control systems to *unstable* nonlinear *networked* control systems. Proofs are not included in the paper for lack of space. A full version of this paper can be found in [5].

II. Notation

The symbols $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{R}_0^+$ denote the set of natural, nonnegative integer, integer, real, positive real, and nonnegative real numbers, respectively. Given a set A we denote $A^2 = A \times A$ and $A^{n+1} = A \times A^n$ for any $n \in \mathbb{N}$. Given an interval $[a, b] \subseteq \mathbb{R}$ with $a \leq b$ we denote by $[a; b]$ the set $[a, b] \cap \mathbb{N}$. We denote by $\lceil x \rceil = \min\{n \in \mathbb{Z} | n \geq x\}$ the ceiling of a real number x. Given a vector $x \in \mathbb{R}^n$ we denote by $\|x\|$ the infinity norm and by $\|x\|_2$ the Euclidean norm of x. Given $\mu \in \mathbb{R}^+$ and $A \subseteq \mathbb{R}^n$, we set $[A]_\mu = \mu\mathbb{Z}^n \cap A$; if $B = \bigcup_{i\in[1;N]} A^i$ then $[B]_\mu = \bigcup_{i\in[1;N]}([A]_\mu)^i$. Consider a bounded set $A \subseteq \mathbb{R}^n$ with interior. Let $H = [a_1, b_1]\times[a_2, b_2]\times\cdots\times[a_n, b_n]$ be the smallest hyperrectangle containing A and set $\hat{\mu}_A = \min_{i=1,2,\ldots,n}(b_i - a_i)$. It is readily seen that for any $\mu \leq \hat{\mu}_A$ and any $a \in A$ there always exists $b \in [A]_\mu$ such that $\|a - b\| \leq \mu$. Given $a \in A \subseteq \mathbb{R}^n$ and a precision $\mu \in \mathbb{R}^+$, the symbol $[a]_\mu$ denotes a vector in $\mu\mathbb{Z}^n$ such that $\|a - [a]_\mu\| \leq \mu/2$. Any vector $[a]_\mu$ with $a \in A$ can be encoded by a finite binary word of length $\lceil \log_2 |[A]_\mu| \rceil$. Given a pair of sets A and B and a relation $\mathcal{R} \subseteq A \times B$, the symbol $\mathcal{R}^{-1}$ denotes the inverse relation of $\mathcal{R}$, i.e. $\mathcal{R}^{-1} = \{(b, a) \in B \times A : (a, b) \in \mathcal{R}\}$. The cardinality of a finite set A is denoted by $|A|$.

III. Networked Control Systems

The class of Network Control Systems (NCS) that we consider in this paper has been introduced in [2]. In this section we briefly review this model. For more details the interested reader is referred to [2]. The network scheme of the NCS is depicted in Figure 1. The direct branch of the

A. Borri is with the Istituto di Analisi dei Sistemi ed Informatica A. Ruberti, Consiglio Nazionale delle Ricerche (IASI-CNR), Viale Manzoni 30, 00185 Rome, Italy, alessandro.borri@iasi.cnr.it.

G. Pola and M.D. Di Benedetto are with the Department of Information Engineering, Computer Science and Mathematics, Center of Excellence DEWS, University of L'Aquila, 67100 L'Aquila, Italy, {giordano.pola,mariadomenica.dibenedetto}@univaq.it.

The research leading to these results has been partially supported by the Center of Excellence DEWS and received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n. 257462 HYCON2 Network of excellence.

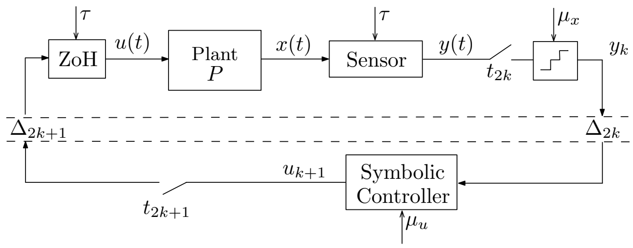

Fig. 1. Networked control system.

network includes the plant P, that is a nonlinear control system of the form:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), \\ x \in X \subseteq \mathbb{R}^n, \\ x(0) \in X_0 \subseteq X, \\ u(\cdot) \in \mathcal{U}, \end{cases} \tag{1}$$

where $x(t)$ and $u(t)$ are the state and the control input at time $t \in \mathbb{R}_0^+$, X is the state space, X_0 is the set of initial states and $\mathcal{U}$ is the set of control inputs that are supposed to be piecewise–constant functions of time from intervals of the form $]a, b[\subseteq \mathbb{R}$ to $U \subseteq \mathbb{R}^m$. We suppose that sets X and U are convex, bounded and with interior. The function $f : X \times U \to X$ is such that $f(0, 0) = 0$ and assumed to be Lipschitz on compact sets. In the sequel we denote by $\mathbf{x}(t, x_0, u)$ the state reached by (1) at time t under the control input u from the initial state x_0; this point is uniquely determined, since the assumptions on f ensure existence and uniqueness of trajectories. We assume that the control system P is forward complete, namely that every trajectory is defined on an interval of the form $]a, \infty[$. On the two sides of the plant P in Figure 1, a Zero-order-Holder (ZoH) and a (ideal) sensor are placed. We assume that the ZoH and the sensor are synchronized and update their output values at times that are integer multiples of the same interval $\tau \in \mathbb{R}^+$, i.e. $u(s\tau+t) = u(s\tau)$, $y(s\tau+t) = y(s\tau) = x(s\tau)$, $t \in [0, \tau[$, $s \in \mathbb{N}_0$, where s is the index of the sampling interval (starting from 0). The evolution of the NCS is described iteratively in the following, starting from the initial time $t = 0$. Consider the k–th iteration in the feedback loop. The sensor requests access to the network and after a waiting time $\Delta_{2k}^{\mathrm{req}} \in [0, \Delta_{\max}^{\mathrm{req}}]$, it sends at time t_{2k} the latest available sample $y_k = [y(t_{2k})]_{\mu_x}$ where μ_x is the precision of the quantizer that follows the sensor in the NCS scheme in Figure 1. The *sensor-to-controller (sc)* link of the network introduces a delay $\Delta_{2k} = \Delta_{\mathrm{send}}^{\mathrm{sc}} + \Delta_{2k}^{\mathrm{delay}}$, with $\Delta_{2k}^{\mathrm{delay}} \in [\Delta_{\min}^{\mathrm{delay}}, \Delta_{\max}^{\mathrm{delay}}]$, where $\Delta_{\mathrm{send}}^{\mathrm{sc}} = \lceil \log_2 |[X]_{\mu_x}| \rceil / B_{\max}$ is the minimum time required to send the information over the sensor-to-controller branch, assuming a digital communication channel of bandwitdh $B_{\max} \in \mathbb{R}^+$ (expressed in bits per second (bps)). The maximum network delay $\Delta_{\max}^{\mathrm{delay}}$ takes into account congestion, other accesses to the communication channel, any kind of scheduling protocol and a finite number of subsequent packet dropouts, which is assumed to be uniformly bounded. After that time, the sensor sample reaches the symbolic controller, that is expressed in terms of the function $C : [X]_{\mu_x} \to [U]_{\mu_u}$, with $\mu_x \leq \hat{\mu}_X$ and $\mu_u \leq \hat{\mu}_U$ so that the domain and co–domain of C are non–empty. After a time $\Delta_k^{\mathrm{ctrl}} \in [\Delta_{\min}^{\mathrm{ctrl}}, \Delta_{\max}^{\mathrm{ctrl}}]$, the value $u_{k+1} = C(y_k)$ is returned and it is sent through the network at time t_{2k+1} (after a bounded waiting time $\Delta_{2k+1}^{\mathrm{req}} \in [0, \Delta_{\max}^{\mathrm{req}}]$). The *controller-to-actuator (ca)* link of the network introduces a delay $\Delta_{2k+1} = \Delta_{\mathrm{send}}^{\mathrm{ca}} + \Delta_{2k+1}^{\mathrm{delay}}$, where $\Delta_{2k+1}^{\mathrm{delay}} \in [\Delta_{\min}^{\mathrm{delay}}, \Delta_{\max}^{\mathrm{delay}}]$ and $\Delta_{\mathrm{send}}^{\mathrm{ca}} = \lceil \log_2 |[U]_{\mu_u}| \rceil / B_{\max}$ is the minimum time required to send the information over the controller-to-actuator branch of the network. After that time, the sample reaches the ZoH and at time $t = A_{k+1}\tau$ the ZoH is refreshed to the control value u_{k+1}, with $A_{k+1} = \lceil (t_{2k+1} + \Delta_{2k+1})/\tau \rceil$. The next iteration starts and the sensor requests access to the network again. Consider now the sequence of control values $\{u_k\}_{k \in \mathbb{N}_0}$. Each value is held for $N_k = A_{k+1} - A_k$ sampling intervals. Due to the bounded delays, one gets $N_k \in [N_{\min}; N_{\max}]$, with:

$$N_{\min} = \lceil \Delta_{\min}/\tau \rceil, \quad N_{\max} = \lceil \Delta_{\max}/\tau \rceil, \tag{2}$$

where we set $\Delta_{\min} = \Delta_{\mathrm{send}}^{\mathrm{sc}} + \Delta_{\min}^{\mathrm{ctrl}} + \Delta_{\mathrm{send}}^{\mathrm{ca}} + 2\Delta_{\min}^{\mathrm{delay}}$, $\Delta_{\max} = \Delta_{\mathrm{send}}^{\mathrm{sc}} + \Delta_{\max}^{\mathrm{ctrl}} + \Delta_{\mathrm{send}}^{\mathrm{ca}} + 2\Delta_{\max}^{\mathrm{req}} + 2\Delta_{\max}^{\mathrm{delay}}$. In the sequel we refer to the described NCS by Σ and to a trajectory of Σ with initial state x_0 and control input u by $\mathbf{x}(., x_0, u)$.

IV. Systems, Approximate Equivalence and Composition

We use the notion of system as a unified mathematical framework to describe NCS as well as their symbolic models.

Definition 4.1: [6] 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 $\longrightarrow \subseteq X \times U \times X$, a set of outputs Y and an output function $H : X \to Y$. A transition $(x, u, x') \in \longrightarrow$ is denoted by $x \xrightarrow{u} x'$. For such a transition, state x' is called a u-successor, or simply a successor, of state x.

A state run of S is a (possibly infinite) sequence of transitions $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \ldots$ with $x_0 \in X_0$. An output run is a (possibly infinite) sequence $\{y_i\}_{i \in \mathbb{N}_0}$ such that there exists a state run with $y_i = H(x_i)$, $i \in \mathbb{N}_0$. 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 state $x' \in X$ such that $x \xrightarrow{u} x'$ for some $u \in U$, *non–blocking* if for any $x \in X$ there exists at least one state $x' \in X$ such that $x \xrightarrow{u} x'$ for some $u \in U$, *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.

Definition 4.2: 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 $x \in X_1$.

In the sequel we consider (alternating) approximate simulation relations [6] to relate properties of NCS and symbolic models.

Definition 4.3: [7], [8] 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 let $\varepsilon \in \mathbb{R}_0^+$ be a given precision. Consider a relation $\mathcal{R} \subseteq X_1 \times X_2$ satisfying the following conditions:

(i) $\forall x_1 \in X_{0,1}$ $\exists x_2 \in X_{0,2}$ such that $(x_1, x_2) \in \mathcal{R}$;
(ii) $\forall (x_1, x_2) \in \mathcal{R}$, $d(H_1(x_1), H_2(x_2)) \leq \varepsilon$.

Relation $\mathcal{R}$ is an *ε–approximate simulation relation* from S_1 to S_2 if it enjoys conditions (i), (ii) and the following one:

(iii) $\forall (x_1, x_2) \in \mathcal{R}$ $\forall x_1 \xrightarrow[1]{u_1} x_1'$ $\exists x_2 \xrightarrow[2]{u_2} x_2'$ such that $(x_1', x_2') \in \mathcal{R}$.

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 *alternating ε–approximate (AεA) simulation relation* from S_1 to S_2 if it enjoys conditions (i), (ii) and the following one:

(iii′) $\forall (x_1, x_2) \in \mathcal{R}$ $\forall u_1 \in U_1$ $\exists u_2 \in U_2$ $\forall x_2 \xrightarrow[2]{u_2} x_2'$ $\exists x_1 \xrightarrow[1]{u_1} x_1'$ such that $(x_1', x_2') \in \mathcal{R}$.

System S_1 is alternating ε–simulated by S_2 or S_2 alternating ε–simulates S_1, denoted $S_1 \preceq_\varepsilon^{\text{alt}} S_2$, if there exists an $A\varepsilon A$ simulation relation from S_1 to S_2.
For more details on the above notions we refer to [6], [7], [8]. We conclude this section with the notion of approximate feedback composition, that is employed in the sequel to capture feedback interaction between non-deterministic systems and symbolic controllers.

Definition 4.4: [6] Consider a pair of metric systems $S_i = (X_i, X_{0,i}, U_i, \xrightarrow[i]{}, Y_i, H_i)$ $(i = 1,2)$ with the same output sets $Y_1 = Y_2$ and metric d. Let $\mathcal{R}$ be an $A\theta A$ simulation relation from S_2 to S_1. The θ–approximate feedback composition of S_1 and S_2, with composition relation $\mathcal{R}$, is the system $S_1 \times_\theta^{\mathcal{R}} S_2 = (X, X_0, U, \xrightarrow{}, Y, H)$, where $X = \mathcal{R}^{-1}$, $X_0 = X \cap (X_{0,1} \times X_{0,2})$, $U = U_1$, $(x_1, x_2) \xrightarrow{u_1} (x_1', x_2')$ if $x_1 \xrightarrow[1]{u_1} x_1'$ and $x_2 \xrightarrow[2]{u_2} x_2'$, $Y = Y_1$, $H(x_1, x_2) = H_1(x_1)$ for any $(x_1, x_2) \in X$.

V. Symbolic Models for NCS

In this section we propose symbolic models that approximate NCS in the sense of (alternating) approximate simulation. For notational simplicity we denote by u any constant control input $\tilde{u} \in \mathcal{U}$ s.t. $\tilde{u}(t) = u$ at all times $t \in \mathbb{R}^+$. Set $X_e = \cup_{N \in [N_{\min}; N_{\max}]} X^N$.

Definition 5.1: [2] Given the NCS Σ, consider the system $S(\Sigma) = (X_\tau, X_{0,\tau}, U_\tau, \xrightarrow[\tau]{}, Y_\tau, H_\tau)$ where:

- X_τ is the subset of $X_0 \cup X_e$ such that for any $x = (x_1, x_2, ..., x_N) \in X_\tau$, with $N \in [N_{\min}; N_{\max}]$, the following conditions hold: $x_{i+1} = \mathbf{x}(\tau, x_i, u^-)$ $(i \in [1; N-2])$ and $x_N = \mathbf{x}(\tau, x_{N-1}, u^+)$ for some constant functions u^-, $u^+ \in [U]_{\mu_u}$;
- $X_{0,\tau} = X_0$;
- $U_\tau = [U]_{\mu_u}$;
- $x^1 \xrightarrow[\tau]{u} x^2$, where $x_{i+1}^1 = \mathbf{x}(\tau, x_i^1, u_1^-)$ for all $i \in [1; N_1 - 2]$, $x_{N_1}^1 = \mathbf{x}(\tau, x_{N_1-1}^1, u_1^+)$, $x_{i+1}^2 = \mathbf{x}(\tau, x_i^2, u_2^-)$ for all $i \in [1; N_2 - 2]$, $x_{N_2}^2 = \mathbf{x}(\tau, x_{N_2-1}^2, u_2^+)$, $u_2^- = u_1^+$, $u_2^+ = u$, $x_1^2 = \mathbf{x}(\tau, x_{N_1}^1, u_2^-)$, for some $N_1, N_2 \in [N_{\min}; N_{\max}]$;
- $Y_\tau = X_\tau$;
- $H_\tau(x) = x$.

Note that $S(\Sigma)$ is non-deterministic because, depending on the values of N_2, more than one u–successor of x^1 may exist. Since the state vectors of $S(\Sigma)$ are built from trajectories of Σ sampled every τ time units, $S(\Sigma)$ collects all the information of the NCS Σ available at the sensor (see Figure 1) as formally stated in Theorem 5.1 of [2]. System $S(\Sigma)$ can be regarded as metric with the metric d_{Y_τ} on Y_τ naturally induced by the metric $d_X(x_1, x_2) = \|x_1 - x_2\|$ on X, as follows. Given any $x^i = (x_1^i, x_2^i, ..., x_{N_i}^i)$, $i = 1,2$, we set $d_{Y_\tau}(x^1, x^2) = \max_{i \in [1;N]} \|x_i^1 - x_i^2\|$, if $N_1 = N_2 = N$ and $d_{Y_\tau}(x^1, x^2) = +\infty$, otherwise. Although system $S(\Sigma)$ contains all the information of the NCS Σ available at the sensor, it is not a finite model. In the following, we propose a system that approximates $S(\Sigma)$ and is symbolic. A key property for our developments is the notion of incremental forward completeness, as recalled hereafter.

Definition 5.2: [3] Control system (1) is incrementally forward complete (δ-FC) if it is forward complete and there exists a continuous function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ such that for every $s \in \mathbb{R}^+$, the function $\beta(\cdot, s)$ belongs to class $\mathcal{K}_\infty$, and for any $x_1, x_2 \in X$, any $\tau \in \mathbb{R}^+$, and any $u \in \mathcal{U}$, the following condition is satisfied for all $t \in [0, \tau]$:

$$\|\mathbf{x}(t, x_1, u) - \mathbf{x}(t, x_2, u)\| \leq \beta(\|x_1 - x_2\|, t).$$

Incremental forward completeness requires the distance between two arbitrary trajectories to be bounded by a continuous function capturing the mismatch between initial conditions. The class of δ-FC control systems is rather large and includes also some subclasses of unstable control systems; for instance unstable linear systems are δ-FC. The notion of δ-FC can be described in terms of Lyapunov-like functions.

Definition 5.3: A smooth function $V : X \times X \to \mathbb{R}$ is called a δ–FC Lyapunov function for the control system (1) if there exist $\lambda \in \mathbb{R}$ and $\mathcal{K}_\infty$ functions $\underline{\alpha}$ and $\overline{\alpha}$ such that, for any $x_1, x_2 \in X$ and any $u \in U$, the following conditions hold true:

(i) $\underline{\alpha}(\|x_1 - x_2\|) \leq V(x_1, x_2) \leq \overline{\alpha}(\|x_1 - x_2\|)$,
(ii) $\frac{\partial V}{\partial x_1} f(x_1, u) + \frac{\partial V}{\partial x_2} f(x_2, u) \leq \lambda V(x_1, x_2)$.

The existence of a δ-FC Lyapunov function was proven in [3] to be a sufficient condition for δ-FC of a control system. In the following we suppose that the control system P in the NCS Σ enjoys the following properties:

(H1) There exists a δ–FC Lyapunov function V satisfying the inequality (ii) in Definition 5.3 for some $\lambda \in \mathbb{R}$;
(H2) There exists a $\mathcal{K}_\infty$ function γ such that $V(x, x') - V(x, x'') \leq \gamma(\|x' - x''\|)$, for every $x, x', x'' \in X$.

Given a design parameter $\eta \in \mathbb{R}^+$, define the following system $S_*(\Sigma) = (X_*, X_{0,*}, U_*, \underset{*}{\longrightarrow}, Y_*, H_*)$ where:

- X_* is the subset of $[X_0 \cup X_e]_{\mu_x}$ such that for any $x^* = (x_1^*, x_2^*, ..., x_N^*) \in X_*$ with $N \in [N_{\min}; N_{\max}]$ the following condition holds:
$$V(\mathbf{x}(\tau, x_i^*, u_*^-), x_{i+1}^*) \le e^{\lambda\tau}\underline{\alpha}(\eta) + \gamma(\mu_x), \quad i \in [1; N-2]; \tag{3}$$
$$V(\mathbf{x}(\tau, x_{N-1}^*, u_*^+), x_N^*) \le e^{\lambda\tau}\underline{\alpha}(\eta) + \gamma(\mu_x); \tag{4}$$
for some constant functions u_*^-, $u_*^+ \in [U]_{\mu_u}$;
- $X_{0,*} = [X_0]_{\mu_x}$;
- $U_* = [U]_{\mu_u}$;
- $x^1 \xrightarrow[*]{u_*} x^2$, where $V(\mathbf{x}(\tau, x_i^1, u_1^-), x_{i+1}^1) \le e^{\lambda\tau}\underline{\alpha}(\eta) + \gamma(\mu_x)$ for all $i \in [1; N_1 - 2]$, $V(\mathbf{x}(\tau, x_{N_1-1}^1, u_1^+), x_{N_1}^1) \le e^{\lambda\tau}\underline{\alpha}(\eta) + \gamma(\mu_x)$, $V(\mathbf{x}(\tau, x_i^2, u_2^-), x_{i+1}^2) \le e^{\lambda\tau}\underline{\alpha}(\eta) + \gamma(\mu_x)$ for all $i \in [1; N_2 - 2]$, $V(\mathbf{x}(\tau, x_{N_2-1}^2, u_2^+), x_{N_2}^2) \le e^{\lambda\tau}\underline{\alpha}(\eta) + \gamma(\mu_x)$, $u_2^- = u_1^+$, $u_2^+ = u_*$, $V(\mathbf{x}(\tau, x_{N_1}^1, u_1^+), x_1^2) \le e^{\lambda\tau}\underline{\alpha}(\eta) + \gamma(\mu_x)$, for some $N_1, N_2 \in [N_{\min}; N_{\max}]$;
- $Y_* = X_\tau$;
- $H_*(x^*) = x^*$.

System $S_*(\Sigma)$ is metric when we regard the set of outputs Y_* as being equipped with the metric d_{Y_τ}. We now have all the ingredients to present one of the main results of this paper.

Theorem 5.4: Consider the NCS Σ and suppose that the control system P enjoys properties (H1) and (H2). Then for any desired precision $\varepsilon \in \mathbb{R}^+$, any sampling time $\tau \in \mathbb{R}^+$, any state quantization $\mu_x \in \mathbb{R}^+$ and any choice of the design parameter $\eta \in \mathbb{R}^+$ satisfying the inequality
$$\mu_x \le \min\{\hat{\mu}_X, \overline{\alpha}^{-1}(\underline{\alpha}(\varepsilon))\} \le \eta, \tag{5}$$
we have $S_*(\Sigma) \preceq_\varepsilon^{\text{alt}} S(\Sigma) \preceq_\varepsilon S_*(\Sigma)$.

This result is important because it provides symbolic models for *possibly unstable* nonlinear NCS, with guaranteed approximation bounds. This result generalizes the ones in [2], which instead require incrementally stable NCS.

VI. ROBUST SYMBOLIC CONTROL DESIGN

We consider a control design problem where the NCS Σ has to satisfy a given specification robustly with respect to the non-idealities of the communication network. Our specification is a collection of transitions $\underset{\bar{q}}{\longrightarrow} \subseteq \bar{X}_q \times \bar{X}_q$, where $\bar{X}_q$ is a finite subset of $\mathbb{R}^n$. Given a set of initial states $\bar{X}_q^0 \subseteq \bar{X}_q$, we now reformulate the specification in the form of the system $\mathcal{Q} = (X_q, X_q^0, U_q, \underset{q}{\longrightarrow}, Y_q, H_q)$, where:

- X_q is the subset of $\bar{X}_q^0 \cup \left(\bigcup_{N\in[N_{\min};N_{\max}]} \bar{X}_q^N\right)$ such that for any $x = (x_1, x_2, ..., x_N) \in X_q$, with $N \in [N_{\min}; N_{\max}]$, for any $i \in [1; N-1]$, the transition $x_i \underset{q}{\longrightarrow} x_{i+1}$ is in $\underset{\bar{q}}{\longrightarrow}$;
- $X_q^0 = \bar{X}_q^0$;
- $U_q = \{\bar{u}_q\}$, where $\bar{u}_q$ is a *dummy* symbol;
- $x^1 \xrightarrow[q]{\bar{u}_q} x^2$, where $x^1 = (x_1^1, x_2^1, ..., x_{N_1}^1)$, $x^2 = (x_1^2, x_2^2, ..., x_{N_1}^2)$, $N_1, N_2 \in [N_{\min}; N_{\max}]$ and the transition $x_{N_1}^1 \underset{q}{\longrightarrow} x_1^2$ is in $\underset{\bar{q}}{\longrightarrow}$;
- $Y_q = X_q$;
- $H_q = 1_{X_q}$,

where $N_{\min}$ and $N_{\max}$ are as in (2). We are now ready to state the control problem that we address in this section.

Problem 6.1: Consider the NCS Σ, a specification $\mathcal{Q}$ and a desired precision $\varepsilon \in \mathbb{R}^+$. Find a symbolic controller C, a parameter $\theta \in \mathbb{R}^+$ and a $A\theta A$ simulation relation $\mathcal{R}$ from C to $S(\Sigma)$ such that:

(1) $\varnothing \ne S(\Sigma) \times_\theta^{\mathcal{R}} C \preceq_\varepsilon \mathcal{Q}$;
(2) $S(\Sigma) \times_\theta^{\mathcal{R}} C$ is non-blocking.

Note that the approximate similarity inclusion in (1) requires the state trajectories of the NCS to be close to the ones of specification $\mathcal{Q}$ up to the accuracy ε robustly with respect to the non-determinism imposed by the network. The non-blocking condition (2) prevents deadlocks in the interaction between the plant and the controller. In the following definition, we provide the controller C^* that is shown in the sequel to solve Problem 6.1.

Definition 6.2: Let C^* be the maximal non-blocking subsystem[1] C of $S_*(\Sigma)$ such that $C \preceq_{\mu_x} \mathcal{Q}$ and $C \preceq_0^{\text{alt}} S_*(\Sigma)$.

From the above definition it is easy to see that C^* is symbolic. We are now ready to solve Problem 6.1.

Theorem 6.3: Consider the NCS Σ and the specification $\mathcal{Q}$. Suppose that the control system P in Σ enjoys Assumptions (H1) and (H2). Then for any desired precision $\varepsilon \in \mathbb{R}^+$, choose the parameters $\theta, \mu_x, \eta \in \mathbb{R}^+$ such that:
$$\mu_x + \theta \le \varepsilon, \tag{6}$$
$$\mu_x \le \min\{\hat{\mu}_X, \overline{\alpha}^{-1}(\underline{\alpha}(\theta))\} \le \theta \le \eta. \tag{7}$$
Let $\bar{\mathcal{R}}$ be the maximal $A\theta A$ simulation relation[2] from C^* to $S(\Sigma)$. If $\bar{\mathcal{R}} \ne \varnothing$, Problem 6.1 is solved with $C = C^*$ and $\mathcal{R} = \bar{\mathcal{R}}$.

VII. INTEGRATED DESIGN OF SYMBOLIC CONTROLLERS

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

1 Compute the system $S_*(\Sigma)$;
2 Compute the system $\mathcal{Q}$ from the transition relation $\underset{\bar{q}}{\longrightarrow}$;
3 Compute the controller C^*.

Algorithm 1: Construction of the controller C^*.

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

(i) It requires the preliminary construction of the symbolic system $S_*(\Sigma)$, representing the NCS, and of the system $\mathcal{Q}$, representing the specification.

[1] Here maximality is defined with respect to the preorder induced by the notion of $A0A$ simulation.

[2] The maximal $A\theta A$ simulation relation is the unique $A\theta A$ simulation relation that contains all the $A\theta A$ simulation relations.

(ii) It considers the whole state space of the plant P, while a more efficient algorithm would consider only the accessible part[3] of P.

In order to cope with the drawbacks listed above, inspired by the integrated procedure developed in [4] for the simpler case of symbolic control design of nonlinear systems, we now present a procedure that *integrates each step of Algorithm 1 in one algorithm*. The pseudo-code of the proposed procedure is reported in Algorithm 2 and Algorithm 3. Algorithm 2 is the main one while Algorithm 3 introduces function **BuildTree** 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 $j.i$. Algorithm 2 proceeds as follows. In line 2.2 the set $\mathbf{X}_{target}$ of to-be-processed states is initialized and the set Bad of blocking states is empty. At each basic step, Algorithm 2 processes a (non–processed) state x in line 2.4. The test in line 2.6 verifies the existence of a control input u such that all the states (collected in the vector $\mathbf{x}(N_{\min}\tau{:}N_{\max}\tau, x, u)$) that are reachable from x in the plant in time intervals from $N_{\min}\tau$ to $N_{\max}\tau$ are also reachable (up to the accuracy θ) in the specification through a path of length between $N_{\min}$ and $N_{\max}$. If that happens, the control input u is good for state x (it is added to the controller in line 2.7) and function **BuildTree** is called (line 2.14) from all the states reached in the plant that are not equal to the state x that is being processed (lines 2.11–2.12). If there exists a controller fulfilling the specification for all those states, the boolean variable $Found$ is set to $true$ and a solution is found (lines 2.24–2.25), otherwise it is guaranteed that C^* defined in Definition 6.2 is empty. Algorithm 3 (function **BuildTree**) checks the existence of a control input starting from the current state such that the specification is fulfilled robustly, up to the precision θ. If that happens, the control input is added to the controller (line 3.5) and function **BuildTree** itself is called (line 3.13) recursively from all the states reached in the plant that have not been processed yet (lines 3.8–3.11). If there exists a controller fulfilling the specification for all those states, the function returns true (line 3.16), otherwise (line 3.19) it returns false and the current state is added to the set of bad states (line 3.20). Termination, correctness and complexity of the integrated procedure are discussed in the remainder of this section.

Theorem 7.1: Algorithm 2 terminates in a finite number of steps.

We now show that the controller C^{**}, synthesized in Algorithm 2, solves Problem 6.1.

Theorem 7.2: Let $S_{cl}(\Sigma)$ be the maximal sub-system of $S(\Sigma)$ including all the transitions $x^1 \xrightarrow[\tau]{u} x^2$ in $S(\Sigma)$, with $x^i = (x_1^i, x_2^i, ..., x_{N_i}^i)$, $i = 1, 2$, such that $u = C^{**}(x_{N_1}^1)$. Then $S_{cl}(\Sigma) \preceq_\varepsilon \mathcal{Q}$ and $S_{cl}(\Sigma)$ is non–blocking.

Theorem 7.2 extends the results reported in [2] from stable nonlinear control systems to δ-FC nonlinear NCS. Finally, a comparison of the following results shows that the space complexity of Algorithm 2 is smaller than or equal to the one of Algorithm 1.

[3]The *accessible part* of a system S is the unique accessible system $Ac(S)$ such that $S' \sqsubseteq Ac(S) \sqsubseteq S$, for any accessible system $S' \sqsubseteq S$.

1 **Input:** NCS Σ, specification $\mathcal{Q}$, precision $\varepsilon \in \mathbb{R}^+$, quantization parameters $\theta, \mu_x, \eta \in \mathbb{R}^+$ satisfying the inequalities in (6–7);
2 **Init:** $\mathbf{X}_{target} = \{x_p \in X_{0,*} : \exists x_q \in X_q^0 : \|x_p - x_q\| \leq \theta\}$, **global** $Bad = \varnothing$, **global** C^{**}, $found = false$;
3 **while** $\mathbf{X}_{target} \neq \varnothing \wedge found == false$ **do**
4   **choose** $x \in \mathbf{X}_{target}$;
5   $\mathbf{U}_{target} = U_*$;
6   **while** $\mathbf{U}_{target} \neq \varnothing \wedge found == false$ **do**
7   **choose** $u \in \mathbf{U}_{target}$;
8     $C^{**} = \varnothing$;
9   **if** $\mathbf{x}(N_{\min}\tau{:}N_{\max}\tau, x, u)$ *meets* $\mathcal{Q}$ *up to* θ **then**
10   **for** $N = N_{\min} : N_{\max}$ **do**
11   **if** $\exists x_c \in \mathbf{Domain}(C^{**})$: $\|\mathbf{x}(N\tau, x, u)] - x_c\| \leq \theta$ **then**
12           $Flag_N = true$;
13   **else**
14           $Flag_N =$ **BuildTree**$([\mathbf{x}(N\tau, x, u)]_{\mu_x})$;
15   **end**
16   **end**
17   **end**
18     $found = \bigwedge_{N=N_{\min}}^{N_{\max}} Flag_N$;
19     $\mathbf{U}_{target} = \mathbf{U}_{target} \setminus \{u\}$;
20   **end**
21   $\mathbf{X}_{target} = \mathbf{X}_{target} \setminus \{x\}$;
22 **end**
23 **if** $found == true$ **then**
24   $C^{**}(x) = u$;
25   **Controller found successfully!**
26 **else**
27   $C^{**} = \varnothing$;
28 **end**
29 **output:** C^{**}.

Algorithm 2: Integrated Symbolic Control Design.

Proposition 7.3: The space complexity of Algorithm 1 is $O(|[X]_{\mu_x}|^{N_{\max}-N_{\min}+1})$.

Proposition 7.4: The space complexity of Algorithm 2 is $O(|[X]_{\mu_x}|)$.

VIII. An Illustrative Example

We consider the model of a unicycle P described by the following differential equation:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = f(x, u) = \begin{bmatrix} u_1 \cos(x_3) \\ u_1 \sin(x_3) \\ u_2 \end{bmatrix}, \qquad (8)$$

where the state x belongs to the set $X = X_0 = [-1, 1[\times [-1, 1[\times [-\pi, \pi[$ and the control input u belongs to the set $U = [-1, 1[\times [-1, 1[$. The state quantities are the 2D-coordinates of the center of the vehicle and its orientation,

Function $flag =$ **BuildTree**(x);
Init: $flag = false$, $\mathbf{U}_{target} = U_*$;
while $\mathbf{U}_{target} \neq \varnothing \wedge flag == false$ **do**
  choose $u \in \mathbf{U}_{target}$;
  $C^{**}(x) = u$;
  if $\mathbf{x}(N_{\min}\tau{:}N_{\max}\tau, x, u)$ *meets* $\mathcal{Q}$ *up* θ **then**
    for $N = N_{\min} : N_{\max}$ **do**
      if $\exists x_c \in$ **Domain**(C^{**}) : $\|\mathbf{x}(N\tau, x, u)] - x_c\| \leq \theta$ **then**
        $flag_N = true$;
      else if $[\mathbf{x}(N\tau, x, u)]_{\mu_x} \in Bad$ **then**
        $flag_N = false$;
      else
        $flag_N =$ **BuildTree**$([\mathbf{x}(N\tau, x, u)]_{\mu_x})$;
      end
    end
    $flag = \bigwedge_{N=N_{\min}}^{N_{\max}} flag_N$;
  end
end
if $flag == false$ **then**
  $Bad = Bad \cup \{x\}$;
end

Algorithm 3: Recursive computation of subcontrollers.

while the inputs are the forward and angular velocity. By choosing the quadratic Lyapunov-like function $V(x, x') = 0.5\,\|x - x'\|_2^2$ it is possible to show that control system (8) is δ–FC. The network/computation parameters are $B_{\max} = 1$ kbit$/s$, $\tau = 0.2s$, $\Delta_{\min}^{\text{ctrl}} = 0.001s$, $\Delta_{\max}^{\text{ctrl}} = 0.01s$, $\Delta_{\max}^{\text{req}} = 0.05s$, $\Delta_{\min}^{\text{delay}} = 0.02s$, $\Delta_{\max}^{\text{delay}} = 0.1s$, resulting in $N_{\min} = 1$, $N_{\max} = 2$ from Eqn. (2). In order to construct a symbolic model for Σ, we apply Theorem 5.4. Assumptions (H1)–(H2) are fulfilled for P with $\lambda = 2u_{1,\max}$ and $\gamma(r) = 2\pi r$. For a precision $\varepsilon = 0.15$, and the choice of parameters $\eta = 0.11$, $\mu_x = 0.02$ and $\mu_u = 0.25$, the inequality in (5) holds. We now consider a specification given in the form of a motion planning problem with respect to the position variables x_1 and x_2 of the unicycle. Starting from the origin, the vehicle is required to follow a trajectory visiting (in order) the 4 regions of the plane $Z_1 = [0,1[\times[0,1[$, $Z_2 = [-1,0[\times[0,1[$, $Z_3 = [-1,0[\times[-1,0[$, and $Z_4 = [0,1[\times[-1,0[$, to finally go back to a neighbourhood of the origin. For the choice of the interconnection parameter $\theta = 0.9\varepsilon$, Theorem 6.3 holds and the controller C^* from Definition (6.2) solves the control problem. We also solve the problem by means of the integrated procedure illustrated in Section VII and in the following we compare the results in terms of the computational complexity needed to construct C^* and C^{**}. The total memory occupation and time required to construct C^{**} are respectively 1345 integers and $916\,$s. We did not compute the controller C^*; estimates of space complexity and time complexity in constructing C^* result respectively in $5.8{\cdot}10^{12}$ integers and $4.19{\cdot}10^6$ s. In Figure 2, we show the simulation results for a particular realization of the network uncertainties: it is easy to see that the specifications are indeed met.

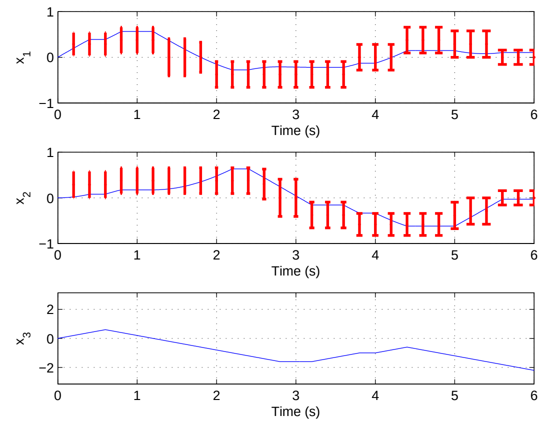

Fig. 2. State trajectory of the NCS Σ.

IX. Conclusions

In this paper we proposed an integrated symbolic design approach to nonlinear NCS. Under the assumption of incremental forward completeness, symbolic models were derived which approximate NCS in the sense of (alternating) approximate simulation. Symbolic control design of NCS was then addressed where specifications are expressed in terms of automata. Finally efficient algorithms were proposed which integrate the construction of symbolic models with the design of robust symbolic controllers.

References

[1] W. Heemels and N. van de Wouw, "Stability and stabilization of networked control systems," in *Networked Control Systems*, ser. Lecture notes in control and information sciences, A. Bemporad, W. Heemels, and M. Johansson, Eds. London: Springer Verlag, 2011, vol. 406, pp. 203–253.

[2] A. Borri, G. Pola, and M. D. Di Benedetto, "A symbolic approach to the design of nonlinear networked control systems," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, ser. HSCC '12. New York, NY, USA: ACM, 2012, pp. 255–264.

[3] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1804–1809, July 2012.

[4] G. Pola, A. Borri, and M. D. Di Benedetto, "Integrated design of symbolic controllers for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 2, pp. 534 –539, feb. 2012.

[5] A. Borri, G. Pola, and M. D. Di Benedetto, "Integrated symbolic design of unstable nonlinear networked control systems," 2012, available at arXiv:1209.0320v2 [cs.SY].

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

[7] 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.

[8] 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.