Symbolic Abstractions of Networked Control Systems

Majid Zamani, *Senior Member, IEEE*, Manuel Mazo, Jr., *Member, IEEE*, Mahmoud Khaled, *Student Member, IEEE*, and Alessandro Abate, *Member, IEEE*

***Abstract*—The last decade has witnessed significant attention on networked control systems (NCSs) due to their ubiquitous presence in industrial applications, and in the particular case of wireless NCSs, because of their architectural flexibility and low installation and maintenance costs. In wireless NCSs, the communication between sensors, controllers, and actuators is supported by a communication channel that is likely to introduce variable communication delays, packet losses, limited bandwidth, and other practical nonidealities leading to numerous technical challenges. Although stability properties of NCSs have been investigated extensively in the literature, results for NCSs under more complex and general objectives, and, in particular, results dealing with verification or controller synthesis for logical specifications, are much more limited. This paper investigates how to address such complex objectives by constructively deriving symbolic models of NCSs, while encompassing the mentioned network nonidealities. The obtained abstracted (symbolic) models can then be employed to synthesize hybrid controllers enforcing rich logical specifications over the concrete NCS models. Examples of such general specifications include properties expressed as formulas in linear temporal logic or as automata on infinite strings. We thus provide a general synthesis framework that can be flexibly adapted to a number of NCS setups. We illustrate the effectiveness of the results over some case studies.**

***Index Terms*—Automata, control system synthesis, formal verification, networked control systems.**

I. Introduction

OVER the last decade, the analysis and synthesis of networked control systems (NCS) have received significant attention. NCSs are ubiquitous in most of the industrial applications due to their many advantages over traditional control systems, such as increased architectural flexibility and reduced installation and maintenance costs, particularly for wireless NCSs. The numerous nonidealities of the network in an NCS introduce new challenges for the analysis of the behavior (such as the stability) of the plant and for the synthesis of new control schemes. The various nonidealities of the network can be broadly categorized as follows: i) quantization errors; ii) packet dropouts; iii) time-varying sampling/transmission intervals; iv) time-varying communication delays; and v) communication constraints (e.g., scheduling protocols). The limited bandwidth of the network does not require a separate classification as it is captured by a combination of quantization errors i) and the communication delays iv). As pointed out later in this paper, category ii) can also be incorporated in category iv), as long as the maximum number of subsequent dropouts over the network is bounded [1].

Recently, there have been many studies focused mostly on the stability properties of NCS: in [2], iii)–v) are simultaneously considered; in [3], i), ii), and iv) are taken into account; in [4], studies ii) and v); [5] focuses on ii) and iii); in [6] and [7], ii)–iv) are considered; and, finally, in [8], i), iii), and v) are taken into account. Despite all of the progress on the stability analysis of NCSs as reported in [2]–[8], there are no mature results in the literature dealing with more complex objectives, such as model verification or formal (controller) synthesis for richer properties expressed as temporal logic specifications [9]. Examples of those specifications include linear temporal logic (LTL) formulas or automata over infinite strings [9], which cannot be investigated with the existing approaches for NCSs. A promising direction to study these complex properties is the use of *symbolic models* [10]. A symbolic model is an abstract description of the original (concrete) dynamical model, where each abstract state (or symbol) corresponds to an aggregate of continuous states in the concrete model. When a finite symbolic model is obtained and is formally related to the original model via the notions of (alternating) approximate (bi)simulations [10] or feedback refinement relations [11], one can leverage algorithmic machinery for controller synthesis of symbolic systems [12] to automatically synthesize hybrid controllers for the original concrete model [10].

To the best of our knowledge, the first results in the literature on the construction of symbolic models for an NCS are [13] and [14]: these results provide symbolic models for the NCS obtained via gridding techniques (discretization of state and control sets); they simultaneously consider the network nonidealities i), ii), and iv); they address symbolic control design with objectives only expressed in terms of nondeterministic automata; the

Manuscript received November 21, 2016; revised June 30, 2017; accepted July 29, 2017. Date of publication August 14, 2017; date of current version December 14, 2018. This work was supported in part by the German Research Foundation (DFG) under Grant ZA 873/1-1. Recommended by Associate Editor H. Ishii. *(Corresponding author: Majid Zamani.)*

M. Zamani and M. Khaled are with the Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany (e-mail: zamani@tum.de; khaled.mahmoud@tum.de).

M. Mazo, Jr., is with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: m.mazo@tudelft.nl).

A. Abate is with the Department of Computer Science, University of Oxford, Oxford OX1 3QD, U.K. (e-mail: alessandro.abate@cs.ox.ac.uk).

Digital Object Identifier 10.1109/TCNS.2017.2739645

possibility of out-of-order packet arrivals is not considered; they exclusively consider static (that is, memoryless) symbolic controllers; and, furthermore, in order to apply standard algorithms for verification and synthesis to the obtained symbolic model, often the given specification requires an additional reformulation over an extended state space, which can lead to significant computation overheads. An extension of the results in [13] and [14] to consider dynamic symbolic controllers was recently proposed in [15].

In this paper, we provide a general construction of symbolic models for the NCS, which can directly employ available and well-investigated symbolic models from the literature that are obtained exclusively for the plant (that is, without the need to encompass the presence of the network explicitly in the construction). As such, one can directly leverage the existing results to obtain symbolic models for the plant, such as grid-based approaches in [11], [16], and [17]; recent results in [18] and [19] that do not require state-space discretization but only input-set discretization; or formula-guided (nongrid-based) approaches in [20]. In this paper, we show that by having a symbolic model of the plant, one can then construct symbolic models for the overall NCS. As a consequence, as long as there exists some types of symbolic abstraction of the plant, one can always use the results provided in this paper to construct symbolic models for the overall complex NCS. As a relevant side result, the techniques discussed in this paper can also be used for models of stochastic plants, in view of recent literature providing symbolic models for such systems [18], [19], [21], [22]. In this paper, we explicitly consider the network nonidealities i), ii), and iv) acting on the NCS simultaneously. We further consider possible out-of-order packet arrivals and message rejections, that is, the effect of older data being neglected because a more recent one is available. Let us also remark that this work is not limited to problems where the controller is static. As a result, without requiring any specific reformulation, we enable the study of large classes of logical specifications, such as those expressed as general LTL formulas or as automata on infinite strings, which are often shown to require dynamic (that is, with memory) symbolic controllers (cf., the example section) [9].

This paper presents a detailed and mature description of the results announced in [23], including a detailed discussion on dealing with the quantized measurements, on the symbolic controller synthesis and refinement, and on the space complexity, and several case studies. Furthermore, we have added a section on related work and provided a detailed comparison with the results in [13] and [14].

II. Notations and Basic Concepts

A. Notations

The identity map on a set A is denoted by 1_A. The symbols $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{R}_0^+$ denote the set of natural, non-negative integer, integer, real, positive, and non-negative real numbers, respectively. Given a set A, define $A^{n+1} = A \times A^n$ for any $n \in \mathbb{N}$. Given a vector $x \in \mathbb{R}^n$, we denote by x_i the ith element of x, and by $\|x\|$ the infinity norm of x. 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 $[\mathbb{R}^n]_\eta = \{a \in \mathbb{R}^n \mid a_i = k_i\eta,\ k_i \in \mathbb{Z},\ i = 1, \ldots, n\}$.

Given a measurable function $f : \mathbb{R}_0^+ \to \mathbb{R}^n$, the (essential) supremum of f is denoted by $\|f\|_\infty$, where $\|f\|_\infty :=$ (ess)sup$\{\|f(t)\|, t \geq 0\}$. A continuous function $\gamma : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$; γ is said to belong to class $\mathcal{K}_\infty$ if $\gamma \in \mathcal{K}$ and $\gamma(r) \to \infty$ as $r \to \infty$. A continuous function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is said to belong to class $\mathcal{KL}$ if, for each fixed s, the map $\beta(r, s)$ belongs to class $\mathcal{K}$ with respect to r and, for each fixed nonzero r, the map $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to \infty$. We identify a relation $R \subseteq A \times B$ with the map $R : A \to 2^B$ defined by $b \in R(a)$ iff $(a, b) \in R$. Given a relation $R \subseteq A \times B$, R^{-1} denotes the inverse relation defined by $R^{-1} = \{(b, a) \in B \times A : (a, b) \in R\}$. When R is an equivalence relation[1] on a set A, we denote by $[a]$ the equivalence class corresponding to the element $a \in A$, by A/R the set of all equivalence classes (quotient set), and by $\pi_R : A \to A/R$ the natural projection map taking a point $a \in A$ to its equivalence class $\pi(a) = [a] \in A/R$.

B. Control Systems

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

Definition 2.1: A *control system* Σ is a tuple $\Sigma = (\mathbb{R}^n, \mathsf{U}, \mathcal{U}, f)$, where:

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

A locally absolutely continuous curve $\xi :]a, b[\to \mathbb{R}^n$ is said to be a *trajectory* of Σ if there exists $\upsilon \in \mathcal{U}$ satisfying

$$\dot{\xi}(t) = f(\xi(t), \upsilon(t))$$

for almost all $t \in\ ]a, b[$. Although we have defined trajectories over open domains, we shall as well refer to trajectories $\xi : [0, t] \to \mathbb{R}^n$ defined on closed domains $[0, t]$, $t \in \mathbb{R}^+$, with the understanding of the existence of a trajectory $\xi' :]a, b[\to \mathbb{R}^n$ such that $\xi = \xi'|_{[0,t]}$ with $a < 0$ and $b > t$. We also write $\xi_{x\upsilon}(t)$ to denote the point reached at time t under the input υ from the initial condition $x = \xi_{x\upsilon}(0)$; the point $\xi_{x\upsilon}(t)$ is uniquely determined due to the assumptions on f [24]. A control system Σ is said to be forward complete if every trajectory is defined on an interval of the form $]a, \infty[$ [25].

C. Notions of Stability and of Completeness

Some of the existing results recalled in this paper require certain stability properties (or lack thereof) on Σ. First, we recall a stability property, introduced in [26], as defined next.

[1] An equivalence relation $R \subseteq X \times X$ is a binary relation on a set X if it is reflexive, symmetric, and transitive.

Definition 2.2: A control system Σ is incrementally input-to-state stable (δ-ISS) if it is forward complete and there exists a $\mathcal{KL}$ function β and a $\mathcal{K}_\infty$ function γ such that for any $t \in \mathbb{R}_0^+$, any $x, \hat{x} \in \mathbb{R}^n$, and any $\upsilon, \hat{\upsilon} \in \mathcal{U}$, the following condition is satisfied:

$$\|\xi_{x\upsilon}(t) - \xi_{\hat{x}\hat{\upsilon}}(t)\| \leq \beta(\|x - \hat{x}\|, t) + \gamma(\|\upsilon - \hat{\upsilon}\|_\infty). \quad \text{(II.1)}$$

Next, we recall a completeness property, introduced in [17], which can be satisfied by larger classes of (even unstable) control systems.

Definition 2.3: A control system Σ is incrementally forward complete (δ-FC) if it is forward complete and there exist continuous functions $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ and $\gamma : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ such that for each fixed s, the functions $\beta(r, s)$ and $\gamma(r, s)$ belong to class $\mathcal{K}_\infty$ with respect to r, and for any $t \in \mathbb{R}_0^+$, any $x, \hat{x} \in \mathbb{R}^n$, and any $\upsilon, \hat{\upsilon} \in \mathcal{U}$, the following condition is satisfied:

$$\|\xi_{x\upsilon}(t) - \xi_{\hat{x}\hat{\upsilon}}(t)\| \leq \beta(\|x - \hat{x}\|, t) + \gamma(\|\upsilon - \hat{\upsilon}\|_\infty, t). \quad \text{(II.2)}$$

As explained in [17, Remark 2.3], δ-FC implies uniform continuity of the map $\phi_t : \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}^n$ defined by $\phi_t(x, \upsilon) = \xi_{x\upsilon}(t)$ for any fixed $t \in \mathbb{R}_0^+$.

We refer the interested readers to the results in [26] (respectively, [17]) providing a characterization (respectively, description) of δ-ISS (respectively, δ-FC) in terms of the existence of so-called *incremental Lyapunov functions*.

III. Systems and Approximate Equivalence Notions

We now recall the notion of *system*, as introduced in [10], that we later use to describe NCS as well as their symbolic abstractions.

Definition 3.1: A system S is a tuple $S = (X, X_0, U, \longrightarrow, Y, H)$ consisting of a (possibly infinite) set of states X, a (possibly infinite) set of initial states $X_0 \subseteq X$, a (possibly infinite) set of inputs U, a transition relation $\longrightarrow \subseteq X \times U \times X$, a set of outputs Y, and an output map $H : X \to Y$.

A transition $(x, u, x') \in \longrightarrow$ is also denoted by $x \xrightarrow{u} x'$. If $x \xrightarrow{u} x'$, state x' is called a u-successor of state x. We denote by $\mathbf{Post}_u(x)$ the set of all u-successors of a state x, and by $U(x)$ the set of inputs $u \in U$ for which $\mathbf{Post}_u(x)$ is nonempty. We denote by $\mathcal{T}(U, Y)$ the set of all systems associated with a set of inputs U and a set of outputs Y. A system S is said to be:

1) *metric*, if the output set Y is equipped with a metric $\mathsf{d} : Y \times Y \to \mathbb{R}_0^+$;
2) *finite* (or *symbolic*), if X and U are finite sets;
3) *countable*, if X and U are countable sets;
4) *deterministic*, if for any state $x \in X$ and any input $u \in U(x)$, $|\mathbf{Post}_u(x)| = 1$;
5) *nondeterministic*, if there exist a state $x \in X$ and an input $u \in U$ such that $|\mathbf{Post}_u(x)| > 1$.

Given a system $S = (X, X_0, U, \longrightarrow, Y, H)$, we denote by $|S|$ the size of S, defined as $|S| := |\longrightarrow|$, which is equal to the total number of transitions in S. Note that it is more reasonable to consider $|\longrightarrow|$ as the size of S rather than $|X|$ because, in practice, it is the transitions of S that are required to be stored rather than just the states of S.

We recall the notions of (alternating) approximate (bi)simulation relation, introduced in [27] and [28], which are useful to relate properties of NCSs to those of their symbolic models. First, we recall the notion of approximate (bi)simulation relation, introduced in [27].

Definition 3.2: Let $S_a = (X_a, X_{a0}, U_a, \underset{a}{\longrightarrow}, Y_a, H_a)$ and $S_b = (X_b, X_{b0}, U_b, \underset{b}{\longrightarrow}, Y_b, H_b)$ be metric systems with the same output sets $Y_a = Y_b$ and metric d. For $\varepsilon \in \mathbb{R}_0^+$, a relation $R \subseteq X_a \times X_b$ is said to be an ε-approximate simulation relation from S_a to S_b if the following three conditions are satisfied.

i) For every $x_{a0} \in X_{a0}$, there exists $x_{b0} \in X_{b0}$ with $(x_{a0}, x_{b0}) \in R$.
ii) For every $(x_a, x_b) \in R$, we have $\mathsf{d}(H_a(x_a), H_b(x_b)) \leq \varepsilon$.
iii) For every $(x_a, x_b) \in R$, the existence of $x_a \underset{a}{\xrightarrow{u_b}} x'_a$ in S_a implies the existence of $x_b \underset{b}{\xrightarrow{u_b}} x'_b$ in S_b satisfying $(x'_a, x'_b) \in R$.

A relation $R \subseteq X_a \times X_b$ is said to be an ε-approximate bisimulation relation between S_a and S_b if R is an ε-approximate simulation relation from S_a to S_b and R^{-1} is an ε-approximate simulation relation from S_b to S_a.

System S_a is ε-approximately simulated by S_b, denoted by $S_a \preceq_{\mathcal{S}}^{\varepsilon} S_b$, if there exists an ε-approximate simulation relation from S_a to S_b. System S_a is ε-approximately bisimilar to S_b, denoted by $S_a \cong_{\mathcal{S}}^{\varepsilon} S_b$, if there exists an ε-approximate bisimulation relation between S_a and S_b.

As explained in [28], for nondeterministic systems, we need to consider relationships that explicitly capture the adversarial nature of nondeterminism. Furthermore, these types of relations become crucial to enable the refinement of symbolic controllers [10].

Definition 3.3: Let $S_a = (X_a, X_{a0}, U_a, \underset{a}{\longrightarrow}, Y_a, H_a)$ and $S_b = (X_b, X_{b0}, U_b, \underset{b}{\longrightarrow}, Y_b, H_b)$ be metric systems with the same output sets $Y_a = Y_b$ and metric d. For $\varepsilon \in \mathbb{R}_0^+$, a relation $R \subseteq X_a \times X_b$ is said to be an alternating ε-approximate simulation relation from S_a to S_b if conditions i) and ii) in Definition 3.2, as well as the following condition, are satisfied:

iii) For every $(x_a, x_b) \in R$ and for every $u_a \in U_a(x_a)$, there exists some $u_b \in U_b(x_b)$ such that for every $x'_b \in \mathbf{Post}_{u_b}(x_b)$, there exists $x'_a \in \mathbf{Post}_{u_a}(x_a)$ satisfying $(x'_a, x'_b) \in R$.

A relation $R \subseteq X_a \times X_b$ is said to be an alternating ε-approximate bisimulation relation between S_a and S_b if R is an alternating ε-approximate simulation relation from S_a to S_b and R^{-1} is an alternating ε-approximate simulation relation from S_b to S_a.

System S_a is alternatingly ε-approximately simulated by S_b, denoted by $S_a \preceq_{\mathcal{AS}}^{\varepsilon} S_b$, if there exists an alternating ε-approximate simulation relation from S_a to S_b. System S_a is alternatingly ε-approximately bisimilar to S_b, denoted by $S_a \cong_{\mathcal{AS}}^{\varepsilon} S_b$, if there exists an alternating ε-approximate bisimulation relation between S_a and S_b.

It can be readily seen that the notions of approximate (bi)simulation relation and of alternating approximate (bi)simulation relation coincide when the systems involved are deterministic, in the sense of Definition 3.1.

Let us introduce a metric system $S_\tau(\Sigma) := (X_\tau, X_{\tau 0}, U_\tau, \underset{\tau}{\rightarrow}, Y_\tau, H_\tau)$, which captures all of the information contained in the forward complete control system Σ at sampling times $k\tau, \forall k \in \mathbb{N}_0$: $X_\tau = \mathbb{R}^n$, $X_{\tau 0} = \mathbb{R}^n$, $U_\tau = \mathcal{U}$, $Y_\tau = \mathbb{R}^n/Q$ for some given equivalence relation $Q \subseteq X_\tau \times X_\tau$, $H_\tau = \pi_Q$, and $x_\tau \xrightarrow[\tau]{\upsilon_\tau} x'_\tau$ if there exists a trajectory $\xi_{x_\tau \upsilon_\tau} : [0, \tau] \to \mathbb{R}^n$ of Σ satisfying $\xi_{x_\tau \upsilon_\tau}(\tau) = x'_\tau$.

Notice that the set of states and inputs of $S_\tau(\Sigma)$ are uncountable and that $S_\tau(\Sigma)$ is a deterministic system in the sense of Definition 3.1 (cf., Section II-B) since the trajectory of Σ is uniquely determined. We also assume that the output set Y_τ is equipped with a metric $\mathsf{d}_{Y_\tau} : Y_\tau \times Y_\tau \to \mathbb{R}_0^+$.

We refer the interested readers to [16]–[19], proposing results on the existence of symbolic abstractions $S_{\mathsf{q}}(\Sigma) := (X_{\mathsf{q}}, X_{\mathsf{q}0}, U_{\mathsf{q}}, \underset{\mathsf{q}}{\rightarrow}, Y_{\mathsf{q}}, H_{\mathsf{q}})$ for $S_\tau(\Sigma)$. In particular, the results in [16]–[19] provide symbolic abstractions $S_{\mathsf{q}}(\Sigma)$ for δ-ISS and δ-FC control systems Σ, respectively, such that $S_{\mathsf{q}}(\Sigma) \cong^\varepsilon_{\mathcal{S}} S_\tau(\Sigma)$ (equivalently, $S_{\mathsf{q}}(\Sigma) \cong^\varepsilon_{\mathcal{AS}} S_\tau(\Sigma)$)[2] and $S_{\mathsf{q}}(\Sigma) \preceq^\varepsilon_{\mathcal{AS}} S_\tau(\Sigma) \preceq^\varepsilon_{\mathcal{S}} S_{\mathsf{q}}(\Sigma)$, respectively. The results in [16] and [17] assume that Q is the identity relation in the definition of $S_\tau(\Sigma)$, implying that $Y_\tau = \mathbb{R}^n$ and $\pi_Q = 1_{\mathbb{R}^n}$, $\mathcal{U}$ is the set of piecewise constant curves over intervals of length τ [cf. (IV.3)], and the metric d_{Y_τ} is the natural infinity norm metric. While the abstraction results in [16] and [17] are based on state-space discretization, the ones in [18] and [19] do not require any state-space discretization and are potentially more efficient than those in [16] and [17] when dealing with high-dimensional plants.

Remark 3.4: Consider a metric system $S_\tau(\Sigma)$ admitting an abstraction $S_{\mathsf{q}}(\Sigma)$. Since plant Σ is forward complete, one can readily verify that given any state $x_\tau \in X_\tau$, there always exists a υ_τ-successor of x_τ, for any $\upsilon_\tau \in U_\tau$. Hence, $U_\tau(x_\tau) = U_\tau$ for any $x_\tau \in X_\tau$. Therefore, without loss of generality, one can also assume that $U_{\mathsf{q}}(x_{\mathsf{q}}) = U_{\mathsf{q}}$ for any $x_{\mathsf{q}} \in X_{\mathsf{q}}$.

IV. Models of NCSs

Consider an NCS $\widetilde{\Sigma}$ as depicted schematically in Fig. 1, and similar to those discussed in [6, Fig. 1], [7, Fig. 1], and [13, Fig. 1]. The NCS $\widetilde{\Sigma}$ includes a plant Σ, a time-driven sampler, and an event-driven zero-order hold (ZOH), all of which are described in more detail later. The NCS consists of a forward complete plant $\Sigma = (\mathbb{R}^n, \mathsf{U}, \mathcal{U}, f)$, which is connected to a symbolic controller, explained in more detail in the next subsection, over a communication network that induces delays (Δ^{sc} and Δ^{ca}). The state measurements of the plant are sampled by a time-driven sampler at times $s_k := k\tau$, $k \in \mathbb{N}_0$, and we denote $x_k := \xi(s_k)$. The discrete-time control values computed by the symbolic controller at times s_k are denoted by u_k. Time-varying network-induced delays, that is, the sensor-to-controller delay (Δ_k^{sc}) and the controller-to-actuator delay (Δ_k^{ca}), are included in the model. Moreover, packet dropouts in both channels of the network can be incorporated in the delays Δ_k^{sc} and Δ_k^{ca} (increasing them), as long as the maximum number of subsequent dropouts over the network is bounded [1]; we refer the interested readers to [1] for more detailed information. Finally, the time-varying computation time needed to evaluate the symbolic controller is incorporated into Δ_k^{ca} [1]. We assume that the time-varying delays are bounded and are integer multiples of the sampling time τ, that is, $\Delta_k^{\mathrm{sc}} := N_k^{\mathrm{sc}}\tau$, where $N_k^{\mathrm{sc}} \in [N_{\min}^{\mathrm{sc}}; N_{\max}^{\mathrm{sc}}]$, and $\Delta_k^{\mathrm{ca}} := N_k^{\mathrm{ca}}\tau$, where $N_k^{\mathrm{ca}} \in [N_{\min}^{\mathrm{ca}}; N_{\max}^{\mathrm{ca}}]$, for some $N_{\min}^{\mathrm{sc}}, N_{\max}^{\mathrm{sc}}, N_{\min}^{\mathrm{ca}}, N_{\max}^{\mathrm{ca}} \in \mathbb{N}_0$. Note that this assumption implies perfect clock synchronization in the network. Nonetheless, with current technologies, it is possible to reach synchronization at the microsecond level (even on wireless networks) (see, e.g., [29] and [30]). Thus, one can assume that synchronization errors, in general, have a rather small effect that could be easily incorporated in the form of bounded sensor noise (due to signals excursion in that time interval). Furthermore, we model the occurrence of message rejection, that is, the effect of older data being neglected because more recent data are available before the older data arrival, as done in [6] and [7]. The ZOH function (see Fig. 1) is placed before the plant Σ to transform the discrete-time control inputs u_k, $k \in \mathbb{N}_0$, to a continuous-time control input $\upsilon(t) = u_{k^*(t)}$, where $k^*(t) := \max\{k \in \mathbb{N}_0 \mid s_k + \Delta_k^{\mathrm{ca}} \leq t\}$. As argued in [6] and [7], within the sampling interval $[s_k, s_{k+1}[$, $\upsilon(t)$ can be explicitly described by

$$\upsilon(t) = u_{k+j_*^k - N_{\max}^{\mathrm{ca}}}, \quad \text{for } t \in [s_k, s_{k+1}[\tag{IV.1}$$

where $j_*^k \in [0; N_{\max}^{\mathrm{ca}} - N_{\min}^{\mathrm{ca}}]$, the required time-indexing shift needed to determine the control input available at the ZOH, is defined as

$$j_*^k = \lambda\left(\widehat{N}_{N_{\min}^{\mathrm{ca}}}, \widehat{N}_{N_{\min}^{\mathrm{ca}}+1}, \ldots, \widehat{N}_{N_{\max}^{\mathrm{ca}}}\right) \tag{IV.2}$$

and where $\widehat{N}_\ell$, for $\ell \in [N_{\min}^{\mathrm{ca}}; N_{\max}^{\mathrm{ca}}]$, is the delay suffered by the control packet sent ℓ samples beforehand, namely, $\widehat{N}_{N_{\max}^{\mathrm{ca}}-i} = N_{k-N_{\max}^{\mathrm{ca}}+i}^{\mathrm{ca}}$ for any $i \in [0; N_{\max}^{\mathrm{ca}} - N_{\min}^{\mathrm{ca}}]$, and

$$\lambda(\widehat{N}_{N_{\min}^{\mathrm{ca}}}, \ldots, \widehat{N}_{N_{\max}^{\mathrm{ca}}}) := \max\{\arg\min_j \kappa(j, \widehat{N}_{N_{\min}^{\mathrm{ca}}}, \ldots, \widehat{N}_{N_{\max}^{\mathrm{ca}}})\}$$

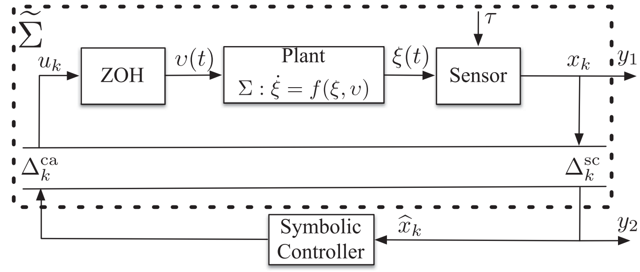

Fig. 1. Schematics of an NCS $\widetilde{\Sigma}$.

[2] Recall that the notions of alternating approximate (bi)simulation and approximate (bi)simulation relation coincide when the systems involved are deterministic.

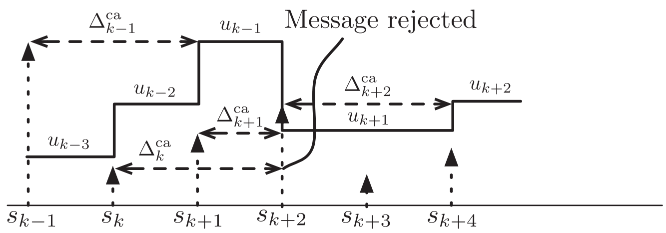

Fig. 2. Time delays in the controller-to-actuator branch of the network with $\Delta_k^{\text{ca}} \in \{\tau, 2\tau, 3\tau\}$.

where

$$\kappa(j, \widehat{N}_{N^{\text{ca}}_{\min}}, \ldots, \widehat{N}_{N^{\text{ca}}_{\max}}) := \min\Big\{\max\{0, \widehat{N}_{N^{\text{ca}}_{\max}-j} + j - N^{\text{ca}}_{\max}\}, \max\{0, \widehat{N}_{N^{\text{ca}}_{\max}-1-j} + j - N^{\text{ca}}_{\max} + 1\}, \ldots, \max\{0, \widehat{N}_{N^{\text{ca}}_{\min}} - N^{\text{ca}}_{\min}\}, 1\Big\}$$

with $j \in [0; N^{\text{ca}}_{\max} - N^{\text{ca}}_{\min}]$. Note that the expression for the continuous-time control input in (IV.1) and (IV.2) takes into account the possible out-of-order packet arrivals and message rejection. For example, in Fig. 2, the time delays in the controller-to-actuator branch of the network are allowed to take values in $\{\tau, 2\tau, 3\tau\}$, resulting in a message rejection at time s_{k+2}. We refer the interested readers to [6, Lemma 1] to understand how the proposed choices for j^k_* (IV.2), λ, and κ can take care of the possible out-of-order packet arrivals and message rejections.

A. Architecture of the Symbolic Controller

A symbolic controller is a finite system that takes the observed states $x_k \in \mathbb{R}^n$ as inputs and produces as outputs the actions $u_k \in \mathsf{U}$ that need to be fed into the system Σ in order to satisfy a given complex logical specification. We refer the interested readers to [10] for the formal definition of symbolic controllers. Although for some LTL specifications (e.g., certain safety or reachability problems), it may be sufficient to consider only static controllers (that is, without memory) [31], we do not limit our work by such an assumption, and the proposed approach in this paper is indeed applicable to general LTL specifications [9]. Due to the presence of a ZOH, from now on, we assume that the set $\mathcal{U}$ contains only curves that are constant over intervals of length $\tau \in \mathbb{R}^+$ and take values in U, i.e.,

$$\mathcal{U} = \big\{\upsilon : \mathbb{R}^+_0 \rightarrow \mathsf{U} | \upsilon(t) = \upsilon((s-1)\tau), t \in [(s-1)\tau, s\tau[, s \in \mathbb{N}\big\}. \tag{IV.3}$$

Correspondingly, one should update U_τ to $\mathcal{U}$ in (IV.3) in the definition of $S_\tau(\Sigma)$ (cf. Section III).

Similar to what was assumed at the connection between the controller and the plant, we also consider possible occurrences of message rejection for the measurement data sent from the sensor to the symbolic controller. The symbolic controller uses $\widehat{x}_k$ as an input at the sampling times $s_k := k\tau$, where

$$\widehat{x}_k = x_{k+\ell^k_* - N^{\text{sc}}_{\max}} \tag{IV.4}$$

where $\ell^k_* \in [0; N^{\text{sc}}_{\max} - N^{\text{sc}}_{\min}]$ is defined as

$$\ell^k_* = \lambda(\widetilde{N}_{N^{\text{sc}}_{\min}}, \widetilde{N}_{N^{\text{sc}}_{\min}+1}, \ldots, \widetilde{N}_{N^{\text{sc}}_{\max}}) \tag{IV.5}$$

where $\widetilde{N}_\ell$, for $\ell \in [N^{\text{sc}}_{\min}; N^{\text{sc}}_{\max}]$, is the delay suffered by the measurement packet sent ℓ samples ago, namely, $\widetilde{N}_{N^{\text{sc}}_{\max}-i} = N^{\text{sc}}_{k-N^{\text{sc}}_{\max}+i}$ for any $i \in [0; N^{\text{sc}}_{\max} - N^{\text{sc}}_{\min}]$, and λ is the function appearing in (IV.2). Note that the expression for the input of the controller in (IV.4) and (IV.5) takes into account the possible out-of-order packet arrivals and message rejections. We again refer the interested readers to [6] and [7] for more details on the proposed choice for ℓ^k_* (IV.5). Here, we assume that the symbolic controller applies its previously computed input value if it does not receive a concrete state measurement from the network, which may be the case for a small interval of time after s_0 due to the initialization of the NCS.

B. Describing NCSs as Metric Systems

As emphasized earlier, one of the main objectives of this work is to provide symbolic models for the overall NCS using symbolic models of their plants component and of the network characteristics. Specifically, we need to define a map taking an (in)finite system describing the plant and the minimum and maximum delays suffered in both the controller-to-actuator and the sensor-to-controller branches of the network as its inputs and providing, correspondingly, an (in)finite system describing the overall NCS as its output. Consider the map

$$\mathcal{L} : \mathcal{T}(U, Y) \times \mathbb{N}^4_0 \rightarrow \mathcal{T}(U, Y) \tag{IV.6}$$

defined as the following: $\forall\ \widetilde{N}_{\min}, \widetilde{N}_{\max} \in \mathbb{N}_0$, where $\widetilde{N}_{\min} \leq \widetilde{N}_{\max}$, $\forall\ \widehat{N}_{\min}, \widehat{N}_{\max} \in \mathbb{N}_0$, where $\widehat{N}_{\min} \leq \widehat{N}_{\max}$, and $\forall\ S_a = (X_a, X_{a0}, U_a, \underset{a}{\rightarrow}, Y_a, H_a) \in \mathcal{T}(U_a, Y_a)$, we have $\mathcal{L}(S_a, \widetilde{N}_{\min}, \widetilde{N}_{\max}, \widehat{N}_{\min}, \widehat{N}_{\max}) = S_b \in \mathcal{T}(U_a, Y_a)$, where $S_b = (X_b, X_{b0}, U_a, \underset{b}{\rightarrow}, Y_a, H_b)$ and

1) $X_b = \{X_a \cup q\}^{\widetilde{N}_{\max}} \times U_a^{\widehat{N}_{\max}} \times [\widetilde{N}_{\min}; \widetilde{N}_{\max}]^{\widetilde{N}_{\max}} \times [\widehat{N}_{\min}; \widehat{N}_{\max}]^{\widehat{N}_{\max}}$, where q is a dummy symbol;
2) $X_{b0} = \{(x_0, q, \ldots, q, u_0, \ldots, u_0, \widetilde{N}_{\max}, \ldots, \widetilde{N}_{\max}, \widehat{N}_{\max}, \ldots, \widehat{N}_{\max}) \mid x_0 \in X_{a0}, u_0 \in U_a\}$;
3) $(x_1, \ldots, x_{\widetilde{N}_{\max}}, u_1, \ldots, u_{\widehat{N}_{\max}}, \widetilde{N}_1, \ldots, \widetilde{N}_{\widetilde{N}_{\max}}, \widehat{N}_1, \ldots, \widehat{N}_{\widehat{N}_{\max}}) \overset{u}{\underset{b}{\longrightarrow}} (x', x_1, \ldots, x_{\widetilde{N}_{\max}-1}, u, u_1, \ldots, u_{\widehat{N}_{\max}-1}, \widetilde{N}, \widetilde{N}_1, \ldots, \widetilde{N}_{\widetilde{N}_{\max}-1}, \widehat{N}, \widehat{N}_1, \ldots, \widehat{N}_{\widehat{N}_{\max}-1})$ for all $\widetilde{N} \in [\widetilde{N}_{\min}; \widetilde{N}_{\max}]$ and all $\widehat{N} \in [\widehat{N}_{\min}; \widehat{N}_{\max}]$ if there exists transition $x_1 \overset{u_{\widehat{N}_{\max}-j^k_*}}{\underset{a}{\longrightarrow}} x'$ in S_a where $j^k_* = \lambda(\widehat{N}_{\widehat{N}_{\min}}, \ldots, \widehat{N}_{\widehat{N}_{\max}})$, as defined in (IV.2), and one of the following holds (due to the initialization of the NCS):
   a) $x_{\widetilde{N}_{\max}-\ell^k_*} = q$, where $\ell^k_* = \lambda(\widetilde{N}_{\widetilde{N}_{\min}}, \ldots, \widetilde{N}_{\widetilde{N}_{\max}})$, defined in (IV.5), and $u = u_1$;
   b) $x_{\widetilde{N}_{\max}-\ell^k_*} \neq q$ and the choice of u is free;
4) $H_b(x_1, \ldots, x_{\widetilde{N}_{\max}}, u_1, \ldots, u_{\widehat{N}_{\max}}, \widetilde{N}_1, \ldots, \widetilde{N}_{\widetilde{N}_{\max}}, \widehat{N}_1, \ldots, \widehat{N}_{\widehat{N}_{\max}}) = H_a(x_1)$ where with a slight abuse of notation, we assume that $H_a(q) := q$.

It can be readily seen that the system S_b is (un)countable or symbolic if the system S_a is (un)countable or symbolic, respectively. Although S_a may be a deterministic system, S_b is, in general, a nondeterministic system (if $\widetilde{N}_{\min} < \widetilde{N}_{\max}$ or $\widehat{N}_{\min} < \widehat{N}_{\max}$), since depending on the values of $\widetilde{N}$ or $\widehat{N}$, more than one u-successor of any state of S_b may exist.

We assume additionally that the output set Y_b is equipped with the same metric d_{Y_a}, which is extended so that $\mathsf{d}_{Y_a}(H_a(x), H_a(q)) = +\infty$ for any $x \in \mathbb{R}^n$ and $\mathsf{d}_{Y_a}(H_a(q), H_a(q)) = 0$.

We have now all of the ingredients to describe the NCS $\widetilde{\Sigma}$ as a metric system. Given $S_\tau(\Sigma)$ and the NCS $\widetilde{\Sigma}$, consider the metric system $S(\widetilde{\Sigma}) := (X, X_0, U, \longrightarrow, Y, H)$, capturing all of the information contained in the NCS $\widetilde{\Sigma}$, given as $S(\widetilde{\Sigma}) = \mathcal{L}(S_\tau(\Sigma), N^{\rm sc}_{\min}, N^{\rm sc}_{\max}, N^{\rm ca}_{\min}, N^{\rm ca}_{\max})$.

Note that the choice of the state space X in $S(\widetilde{\Sigma})$ allows us to keep track of an adequate number of measurements and control packets and the corresponding delays suffered by them, which is necessary and sufficient in order to consider out-of-order packet arrivals and message rejections as explained in detail in [6] and [7]. The choice of the set of initial state X_0 keeps the initial input value u_0 in the ZOH until new control input values arrive. Moreover, assigning the maximum delay suffered by the dummy symbols ensures that those symbols will not take over an actual packet at the later iterations of the network. The transition relation of $S(\widetilde{\Sigma})$ captures in a nondeterministic fashion all of the possible successors of a given state of $S(\widetilde{\Sigma})$, based on all of the possible ordering of measurements arriving to the controller, and of inputs arriving to the ZOH and ensuring that the controller applies its previously computed input value if it does not receive any concrete state measurement from the network. Let us also remark that the sets of states and inputs of $S(\widetilde{\Sigma})$ are uncountable.

Remark 4.1: Note that the output value of any state of $S(\widetilde{\Sigma})$ is simply the output value of the state of the plant available at the sensors at times $s_k := k\tau$. We should highlight that the main role of output sets (respectively, maps) in the definition of systems (cf., Definition 3.1) is to describe the set of atomic propositions (respectively, state labeling) used in describing the specifications and, hence, used for the symbolic controller synthesis. We refer the interested readers to [10, Ch. 5] explaining controller synthesis schemes for some classes of specifications in which the output set plays a role; see, in particular, the discussion after the proof of Proposition 6.8 in [10]. For the implementation (refinement) of symbolic controllers and their composition, one requires dealing with the states of systems rather than their outputs [10, Prop. 8.7]. We elaborate more on the symbolic controller synthesis and refinement in Section VI.

V. Symbolic Models for an NCS

This section contains the main contributions of the paper. We show the existence and construction of symbolic models for NCS by using an existing symbolic model for the plant Σ, namely $S_{\mathsf{q}}(\Sigma) := (X_{\mathsf{q}}, X_{\mathsf{q}0}, U_{\mathsf{q}}, \underset{\mathsf{q}}{\longrightarrow}, Y_{\mathsf{q}}, H_{\mathsf{q}})$.

Given the metric system $S_{\mathsf{q}}(\Sigma)$, define the new metric system $S_*(\widetilde{\Sigma}) := (X_*, X_{*0}, U_*, \underset{*}{\longrightarrow}, Y_*, H_*)$ as $S_*(\widetilde{\Sigma}) = \mathcal{L}(S_{\mathsf{q}}(\Sigma), N^{\rm sc}_{\min}, N^{\rm sc}_{\max}, N^{\rm ca}_{\min}, N^{\rm ca}_{\max})$, where the map $\mathcal{L}$ is defined in (IV.6). System $S_*(\widetilde{\Sigma})$ is constructed in the same way as $S(\widetilde{\Sigma})$, but replacing continuous states, inputs, and the transition relation of $S_\tau(\Sigma)$, with the corresponding ones in $S_{\mathsf{q}}(\Sigma)$.

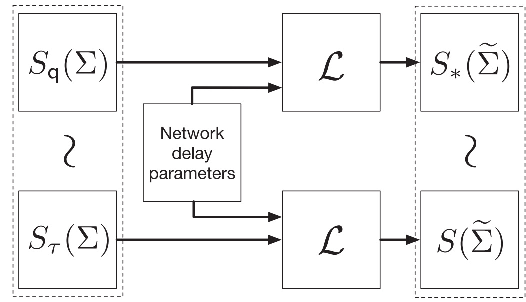

Fig. 3. Symbol $\sim$ represents any of the following relations: $_{\mathcal{S}}^{\varepsilon}\!\succeq$, $\preceq^{\varepsilon}_{\mathcal{AS}}$, and $\cong^{\varepsilon}_{\mathcal{AS}}$.

We can now state the first pair of major technical results of this work, which are schematically represented in Fig. 3.

Theorem 5.1: Consider an NCS $\widetilde{\Sigma}$ and suppose that there exists an abstraction $S_{\mathsf{q}}(\Sigma)$ such that $S_{\mathsf{q}}(\Sigma) \preceq^{\varepsilon}_{\mathcal{AS}} S_\tau(\Sigma) \preceq^{\varepsilon}_{\mathcal{S}} S_{\mathsf{q}}(\Sigma)$. Then, we have $S_*(\widetilde{\Sigma}) \preceq^{\varepsilon}_{\mathcal{AS}} S(\widetilde{\Sigma}) \preceq^{\varepsilon}_{\mathcal{S}} S_*(\widetilde{\Sigma})$.

The proof is provided in [32] and is omitted here due to lack of space.

Corollary 5.2: Consider an NCS $\widetilde{\Sigma}$ and suppose that there exists an abstraction $S_{\mathsf{q}}(\Sigma)$ such that $S_{\mathsf{q}}(\Sigma) \cong^{\varepsilon}_{\mathcal{AS}} S_\tau(\Sigma)$. Then, we have $S_*(\widetilde{\Sigma}) \cong^{\varepsilon}_{\mathcal{AS}} S(\widetilde{\Sigma})$.

The proof is provided in [32] and is omitted here due to lack of space.

Remark 5.3: As discussed earlier, one of the main advantages of the results proposed here in comparison with the ones in [13] and [14] is that one can construct symbolic models for the NCS using symbolic models obtained exclusively for the plant. Therefore, one can readily extend the proposed results to other classes of control systems for the plants, e.g., stochastic control systems, as long as there exist techniques to construct the corresponding symbolic models. For example, one can leverage the recently developed results in [18], [19], [22] (not requiring state-space gridding), and [21] to construct symbolic models for classes of stochastic plants embedded in the NCS.

A. Limited Bandwidth

Assume that an abstraction $S_{\mathsf{q}}(\Sigma)$ exists such that $S_{\mathsf{q}}(\Sigma) \preceq^{\varepsilon}_{\mathcal{AS}} S_\tau(\Sigma)$ equipped with the alternating ε-approximate simulation relation R. From the formal definition of symbolic controllers in [10] constructed based on $S_{\mathsf{q}}(\Sigma)$, one can readily verify the implicit presence of a static set-valued map (a.k.a quantizer map) $\varphi : X_\tau \rightarrow 2^{X_{\mathsf{q}}}$ inside the symbolic controllers, associating with each $x_\tau \in X_\tau$ a set of symbols in X_{q} as follows:

$$\varphi(x_\tau) = \{x_{\mathsf{q}} \in X_{\mathsf{q}} \mid (x_{\mathsf{q}}, x_\tau) \in R\}.$$

Since the map φ is static, one can shift this map toward the sensor in the NCS, as shown in Fig. 4, without affecting any of the presented results. This means that, in general, a set of symbols, rather than only a *quantized* one, needs to be sent over the sensor-to-controller branch of the network. Let us provide a simple example illustrating the problem that may raise if only

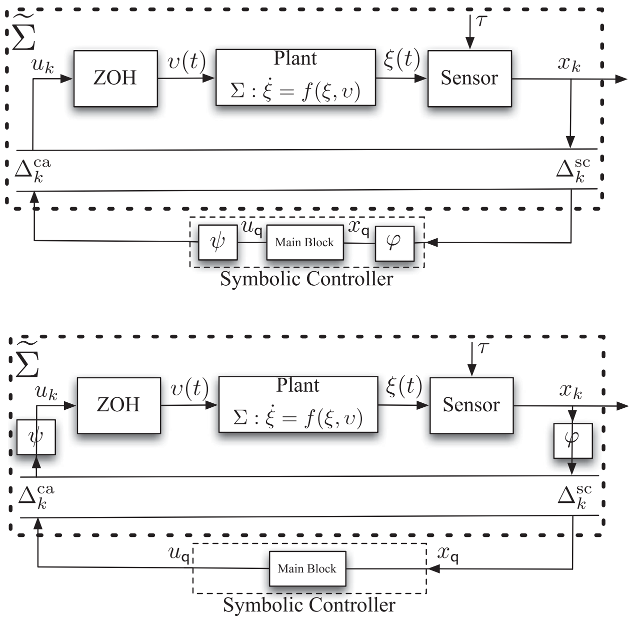

Fig. 4. Shifting maps φ and ψ for the symbolic controller to the other side of the communication network.

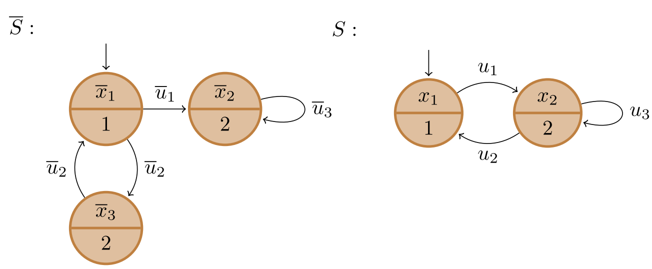

Fig. 5. Finite systems $\overline{S}$ and S.

one of the multiple possible symbols is sent instead of all of them.

Example 5.4: Consider the pair of finite systems in Fig. 5, where the initial states are shown as targets of sourceless arrows, and the lower parts of the states are labeled with their output values. One can readily verify that $R = \{(\overline{x}_1, x_1), (\overline{x}_2, x_2), (\overline{x}_3, x_2)\}$ is an alternating zero-approximate simulation relation from $\overline{S}$ to S. Therefore, $\varphi(x_1) = \{\overline{x}_1\}$ and $\varphi(x_2) = \{\overline{x}_2, \overline{x}_3\}$ is the associated "quantization" map resulting from the relation R. Let us consider the new quantization map $\widetilde{\varphi}$ providing only one state of $\overline{S}$ for each state of S: $\widetilde{\varphi}(x_1) = \{\overline{x}_1\}$ and $\widetilde{\varphi}(x_2) = \{\overline{x}_3\}$. Consider the problem of synthesizing a controller enforcing the output of $\overline{S}$ to reach and stay at set $\{2\}$, namely a controller for the LTL specification $\Diamond\Box\{2\}$. There are infinitely many control sequences over $\overline{S}$ satisfying $\Diamond\Box\{2\}$, e.g., $\overline{u}_1\overline{u}_3\overline{u}_3\cdots$, $\overline{u}_2\overline{u}_2\overline{u}_1\overline{u}_3\overline{u}_3\cdots$, and $\overline{u}_2\overline{u}_2\overline{u}_2\overline{u}_2\overline{u}_1\overline{u}_3\overline{u}_3\cdots$. A possible "static" controller enforcing the desired property could, thus, be obtained by restricting the set of inputs that the controller offers at each state of the abstracted plant, e.g., a map offering at $\bar{x}_1$ input $\bar{u}_1$, at $\bar{x}_3$ input $\bar{u}_2$, and at $\bar{x}_2$ input $\bar{u}_3$. Using the new quantizer map $\widetilde{\varphi}$, and a controller consisting solely of the map in the previous sentence, however, does not allow us to distinguish between $\bar{x}_2$ and $\bar{x}_3$, and the refined control sequences over S would result in $u_1u_2u_1u_2u_1u_2\cdots$. Such a controller would result in the system satisfying infinitely often reaching $\{2\}$ on S, that is, $\Box\Diamond\{2\}$, rather than satisfying the requested specification $\Diamond\Box\{2\}$. While this is a clearly concocted example for illustrative purposes, situations analogous to the one captured by this example arise in the construction of abstractions via notions of (alternating) approximate (bi)simulation (e.g., [17]), in which some concrete states may be associated with several abstract states. For more details on this potential problem, we refer the interested readers to [11].

Remark 5.5: Unfortunately, the problem we just illustrated may arise in the constructions of [13] and [14]. Based on the proposed symbolic abstractions in those works, the set-valued quantizer map $\varphi : \mathbb{R}^n \to 2^{[\mathbb{R}^n]_\eta}$ should be as follows:

$$\varphi(x) = \{x_{\mathsf{q}} \in [\mathbb{R}^n]_\eta \mid \|x - x_{\mathsf{q}}\| \le \varepsilon\}$$

for some given state-space quantization parameter $\eta \in \mathbb{R}^+$ and some precision $\varepsilon \in \mathbb{R}^+$, where $\eta < \varepsilon$; see [13, eq. (18)] and [14, eq. (5)]. However, papers [13] and [14] use the map $\widetilde{\varphi} : x \to [x]_\eta$, where $[x]_\eta \in [\mathbb{R}^n]_\eta$ associates to every $x \in \mathbb{R}^n$ just one quantized state $[x]_\eta \in [\mathbb{R}^n]_\eta$, such that $\|x - [x]_\eta\| \le \eta/2$. For the case of deterministic quantizers (no measurement error), this problem can be readily avoided if the proposed alternating approximate simulation relations in those papers were directly defined over quantized states as proposed in [33]. For the case of nondeterministic quantizers, either one should send a set of symbols to the controllers, as discussed in the beginning of Section V-A, or one should resort to feedback refinement relations [11] (cf., Remark 5.6) and send only one symbol to the controller.

Similarly, a quantization map $\overline{\psi} : X_{\mathsf{q}} \times X_\tau \times U_{\mathsf{q}} \to \mathsf{U}$ is implicitly contained in the symbolic controllers, associating to each symbol $u_{\mathsf{q}} \in U_{\mathsf{q}}(x_{\mathsf{q}})$ generated by the controller an input $u \in U_\tau(x_\tau)$ for some $(x_{\mathsf{q}}, x_\tau) \in R$. Unfortunately, the quantization map $\overline{\psi}$ requires the knowledge of the state of the plant just before the controller. Therefore, one cannot easily shift this map toward the actuator (ZOH) in the NCS scheme. In order to solve this issue, one can simply assume that the set U is finite and $U_{\mathsf{q}} = \mathsf{U}$ and adjust condition (iii) in Definition 3.3 as follows.

iii) For every $(x_{\mathsf{q}}, x_\tau) \in R$, every $u_{\mathsf{q}} \in U_{\mathsf{q}}(x_{\mathsf{q}})$, and every $x'_\tau \in \mathbf{Post}_{u_{\mathsf{q}}}(x_\tau)$ there exists $x'_{\mathsf{q}} \in \mathbf{Post}_{u_{\mathsf{q}}}(x_{\mathsf{q}})$ satisfying $(x'_{\mathsf{q}}, x'_\tau) \in R$,

so that only abstractions $S_{\mathsf{q}}(\Sigma)$ satisfying $S_{\mathsf{q}}(\Sigma) \preceq^{\varepsilon}_{\mathcal{AS}} S_\tau(\Sigma)$ with the new condition (iii) are admitted in our scheme. These modifications simply imply that for each symbolic input u_{q} in $S_{\mathsf{q}}(\Sigma)$, one should apply the same input to $S_\tau(\Sigma)$. Note that we abused notation by identifying u_{q} with the constant input curve with domain $[0, \tau[$ and value u_{q}. With this adjustment, one has a new quantizer map $\psi = 1_{U_{\mathsf{q}}}$, which is static and can be shifted toward the actuator (ZOH) in the NCS, as shown in Fig. 4. Note that the proposed abstractions in [16]–[19], [21], and [22] satisfy this new condition in Definition 3.3 by simply taking $U_{\mathsf{q}} = \mathsf{U}$ in those results. In general, this is a rather natural assumption to be taken as, in practice, one usually considers a finite set of inputs available and constructs abstractions accordingly. We emphasize that the results in Theorem 5.1 and Corollary 5.2 still hold with this modification on condition (iii) in Definition 3.3. Observe that a similar adjustment as this condition (iii) was also proposed in [15, Def. 5].

Remark 5.6: Observe that one can use the recently developed notion of feedback refinement relations introduced in [11] in order to establish the relation between the concrete systems and their symbolic models. This new relation resolves both issues explained in the previous paragraphs: 1) the refined controller only requires the quantized state information of the concrete system; and 2) the abstraction does not need to be used as a building block inside the refined controller, and consequently, a smaller amount of memory is required. We refer the interested readers to [34] showing that the proposed map $\mathcal{L}$ in (IV.6) also preserves the feedback refinement relations and that similar results as in Theorem 5.1 hold for this new relation as well.

VI. Symbolic Controller Synthesis and Refinements

A. Symbolic Controller Synthesis

Although the main contribution of the paper is on the construction of symbolic models for the NCS with some nonidealities, the provided abstractions are amenable to any off-the-shelf symbolic controller synthesis toolbox such as `SCOTS` [35] and `Slugs` [36]. To further elaborate on this, let us consider the following example. Let $A \subset \mathbb{R}^n$ be a compact set. Consider a safety problem, formulated as the satisfaction of the LTL formula[3] $\Box\varphi_A$, where φ_A is a label (or atomic proposition) characterizing the set A. The goal is to synthesize a controller enforcing $\Box\varphi_A$ over the output of the plant, available at the sensors before the network. To do so, we first construct a discrete controller enforcing $\Box\varphi_A$ over the output of $S_*(\widetilde{\Sigma}) = (X_*, X_{*0}, U_*, \underset{*}{\longrightarrow}, Y_*, H_*)$. Whenever $Y_* \neq X_*$ and $H_* \neq 1_{X_*}$, it suffices to consider a new safe set $\widehat{A} \subseteq X_*$ defined by $\widehat{A} = \{\mathsf{x}_* \in X_* \mid H_*(\mathsf{x}_*) \in A\}$. Now, one can apply Theorem 6.6 in [10] to auxiliary system $\widehat{S}_*(\widetilde{\Sigma}) = (X_*, X_{*0}, U_*, \underset{*}{\longrightarrow}, X_*, 1_{X_*})$ and the specification set $\widehat{A}$ to synthesize a discrete controller enforcing $\Box\varphi_A$ over the output of $S_*(\widetilde{\Sigma})$. The main subtlety here is in the refinement of the constructed discrete controller enforcing $\Box\varphi_A$ over the output of the plant which requires the whole state tuple x_* of $S_*(\widetilde{\Sigma})$, while only one of the elements of the tuple is available based on the packet arrived before the controller. We elaborate on the refinement of symbolic controllers in the next subsection and propose a class of the NCS, in which the whole state tuple x_* of $S_*(\widetilde{\Sigma})$ can be recovered inside the controllers.

B. Symbolic Controller Refinement

In order to refine the synthesized symbolic controllers in our setup, we target a class of NCSs, where the upper and lower bounds of the delays are equal at each channel. This implies that all packets suffer the same delay (that is, $\widetilde{N}_k = N^{\mathrm{sc}}_{\min} = N^{\mathrm{sc}}_{\max}$ and $\widehat{N}_k = N^{\mathrm{ca}}_{\min} = N^{\mathrm{ca}}_{\max}$ for any $k \in \mathbb{N}_0$) in each channel. This can be readily achieved by performing extra prolongation (if needed) of the delays suffered already by the packets. For the sensor-to-controller channel, this can be readily done inside the controller. The controller needs to have a buffer to hold arriving packets and keep them in the buffer until their delays reach the maximum. For the controller-to-actuator channel, the same needs to be implemented inside the ZOH. Therefore, in this setting, state (respectively, input) packets are allowed to have any delay (not necessarily integer multiples of the sampling time) between 0 and $N^{\mathrm{sc}}_{\max}$ (respectively, $N^{\mathrm{ca}}_{\max}$), where $N^{\mathrm{sc}}_{\max}$ and $N^{\mathrm{ca}}_{\max}$ are integer multiples of the sampling time.

In this special class of NCSs, the information contained in the NCS $\widetilde{\Sigma}$ is captured by the metric system $S'(\widetilde{\Sigma}) := \mathcal{L}(S_\tau(\Sigma), N^{\mathrm{sc}}_{\max}, N^{\mathrm{sc}}_{\max}, N^{\mathrm{ca}}_{\max}, N^{\mathrm{ca}}_{\max})$. We also denote by $S'_*(\widetilde{\Sigma}) := \mathcal{L}(S_{\mathsf{q}}(\Sigma), N^{\mathrm{sc}}_{\max}, N^{\mathrm{sc}}_{\max}, N^{\mathrm{ca}}_{\max}, N^{\mathrm{ca}}_{\max})$ the corresponding symbolic model of $S'(\widetilde{\Sigma})$. Recall that $S_*(\widetilde{\Sigma})$ denotes the symbolic model of the NCS without the prolongation of delays suffered by packets in both channels of the network. Here, we provide a brief comparison between $S'_*(\widetilde{\Sigma})$ and $S_*(\widetilde{\Sigma})$.

1) $S'_*(\widetilde{\Sigma})$ has no nondeterminism caused by different delay possibilities in comparison with $S_*(\widetilde{\Sigma})$. This results in a smaller transition relation making the controller synthesis less complex.
2) $S'_*(\widetilde{\Sigma})$ is less conservative in comparison with $S_*(\widetilde{\Sigma})$ in terms of the existence of symbolic controllers satisfying some given logic specifications. We elaborate more on this in a lemma later.
3) In terms of actual implementation, the controllers designed for $S'(\widetilde{\Sigma})$ may be more complex than those for $S(\widetilde{\Sigma})$ because they need to have a buffer to hold arriving packets till they reach the required maximum delay; the same needs to be implemented for the ZOH.

Lemma 6.1: Consider a symbolic model S_a and $\widetilde{N}_{\min}$, $\widetilde{N}_{\max}, \widehat{N}_{\min}, \widehat{N}_{\max} \in \mathbb{N}_0$, where $\widetilde{N}_{\min} \leq \widetilde{N}_{\max}$ and $\widehat{N}_{\min} \leq \widehat{N}_{\max}$. We have $S_* \preceq^0_{\mathcal{AS}} S'_*$, where $S_* := \mathcal{L}(S_a, \widetilde{N}_{\min}, \widetilde{N}_{\max}, \widehat{N}_{\min}, \widehat{N}_{\max})$ and $S'_* := \mathcal{L}(S_a, \widetilde{N}_{\max}, \widetilde{N}_{\max}, \widehat{N}_{\max}, \widehat{N}_{\max})$.

The proof is provided in [32] and is omitted here due to lack of space. The result in Lemma 6.1 implies that if there exists a symbolic controller enforcing some complex specifications over S_*, then there exists a symbolic controller enforcing the same complex specifications over S'_* which confirms item 2 in the above comparison between $S'_*(\widetilde{\Sigma})$ and $S_*(\widetilde{\Sigma})$.

Finally, in order to refine the constructed symbolic controllers in closed-loop fashion, one needs to have the symbolic state tuple of the form:

$$(x_{*1}, \ldots, x_{*N^{\mathrm{sc}}_{\max}}, u_1, \ldots, u_{N^{\mathrm{ca}}_{\max}}, N^{\mathrm{sc}}_{\max}, \ldots, N^{\mathrm{sc}}_{\max}, N^{\mathrm{ca}}_{\max}, \ldots, N^{\mathrm{ca}}_{\max}).$$

The controller already knows what control inputs has generated during the $N_{\max} := N^{\mathrm{ca}}_{\max} + N^{\mathrm{sc}}_{\max} - 1$ previous sampling times (that is, $u_1, \ldots, u_{N_{\max}}$). Hence, it just needs to store them in a buffer. The first $N^{\mathrm{ca}}_{\max}$ control inputs (that is, $u_1, \ldots, u_{N^{\mathrm{ca}}_{\max}}$) will be used directly in the sym-

[3] We refer the interested readers to [9] for the formal semantics of the temporal formula $\Box\varphi_A$ expressing the safety property over the set A.

bolic state tuple and the rest for the construction of states $x_{*1}, \ldots, x_{*(N^{\rm sc}_{\max}-1)}$. Now, consider two different cases. *Case 1:* we assume that the symbolic model of the plant (that is, $S_{\sf q}(\Sigma)$) is deterministic (cf., the example section). The controller gets states $x_{*N^{\rm sc}_{\max}}$ using the current measurement packet (that is, $x_{N^{\rm sc}_{\max}}$) and the relation between $S_\tau(\Sigma)$ and $S_{\sf q}(\Sigma)$. Using $x_{*N^{\rm sc}_{\max}}$, previously generated control inputs (that is, $u_{N^{\rm ca}_{\max}+1}, \ldots, u_{N^{\rm ca}_{\max}+N^{\rm sc}_{\max}-1}$), and symbolic model $S_{\sf q}(\Sigma)$, the controller can construct other symbolic state information as follows: $x_{*1} = \mathbf{Post}_{u_{N^{\rm ca}_{\max}+1}}(x_{*2}), x_{*2} = \mathbf{Post}_{u_{N^{\rm ca}_{\max}+2}}(x_{*3}), \ldots$, and $x_{*(N^{\rm sc}_{\max}-1)} = \mathbf{Post}_{u_{N^{\rm ca}_{\max}+N^{\rm sc}_{\max}-1}}(x_{*N^{\rm sc}_{\max}})$. *Case 2:* we assume that the controller has access to the current state measurement of the plant (that is, $x_{N^{\rm sc}_{\max}}$) and the model of the plant. Here, the controller can construct all the state measurements still traveling inside the sensor-to-controller channel up to the current state of the plant (that is, $x_1, \ldots, x_{N^{\rm sc}_{\max}-1}$) using the current packet it receives (that is, $x_{N^{\rm sc}_{\max}}$), previously generated control inputs (that is, $u_{N^{\rm ca}_{\max}+1}, \ldots, u_{N^{\rm ca}_{\max}+N^{\rm sc}_{\max}-1}$), and the model of the plant: $x_1 = \xi_{x_2 u_{N^{\rm ca}_{\max}+1}}(\tau)$, $x_2 = \xi_{x_3 u_{N^{\rm ca}_{\max}+2}}(\tau)$, …, and $x_{N^{\rm sc}_{\max}-1} = \xi_{x_{N^{\rm sc}_{\max}} u_{N^{\rm ca}_{\max}+N^{\rm sc}_{\max}-1}}(\tau)$ (solving the differential equation, possibly numerically, online). Therefore, using the relation between $S_\tau(\Sigma)$ and $S_{\sf q}(\Sigma)$ and $x_1, \ldots, x_{N^{\rm sc}_{\max}}$, symbolic states $x_{*1}, \ldots, x_{*N^{\rm sc}_{\max}}$ are constructed inside the controller.

Remark 6.2: One can use a quantized version of $x_{N^{\rm sc}_{\max}}$ rather than itself in Case 2 above to construct symbolic states $x_{*1}, \ldots, x_{*N^{\rm sc}_{\max}}$ of (not necessarily deterministic) $S_{\sf q}(\Sigma)$ inside the controller. We can use a quantizer with appropriately chosen precision based on the abstraction precision ε, the Lipschitz constant Z in Definition 2.1, and the proposed techniques in [11, Sec. VI-B] to construct symbolic states $x_{*1}, \ldots, x_{*(N^{\rm sc}_{\max}-1)}$ using the model of the plant. On the other hand, one can try to synthesize symbolic controllers with partial information (see, e.g., [37]) $(x_{*N^{\rm sc}_{\max}}, u_1, \ldots, u_{N^{\rm ca}_{\max}}, N^{\rm sc}_{\max}, \ldots, N^{\rm sc}_{\max}, N^{\rm ca}_{\max}, \ldots, N^{\rm ca}_{\max})$, which is left as object of future research. Remark that the computational complexity of synthesis with partial information is usually much larger than the synthesis with full state information [37]. Therefore, there is a tradeoff between having simpler controller synthesis scheme (cf., Section VI-A) amenable to any off-the-shelf synthesis toolbox but more complex refinement scheme (cf., Section VI-B) or having more complex controller synthesis scheme (see, e.g., [37]) not necessarily tractable using off-the-shelf synthesis toolbox but simpler refinement procedure.

VII. Space Complexity Analysis

We compare the results provided here with those in [13] and [14] in terms of the size of the obtained symbolic models. For the sake of a fair comparison, assume that we use also a grid-based symbolic abstraction for the plant Σ using the same sampling time and quantization parameters as the ones in [13] and [14]. Note that the provided comparison may not be complete still, because we do not need any requirement on the symbolic controller, while in [13] and [14], it is assumed that the symbolic controllers are static. By assuming that we are only interested in the dynamics of Σ on a compact set ${\sf D} \subset \mathbb{R}^n$, the cardinality of the set of states of the symbolic models provided in [13] and [14] is

$$|X_\star| = \sum_{i\in\{\{1\}\cup[N_{\min};N_{\max}]\}} \left|[{\sf D}]_\eta\right|^i$$

where $N_{\min} = N^{\rm sc}_{\min} + N^{\rm ca}_{\min}$, $N_{\max} = N^{\rm sc}_{\max} + N^{\rm ca}_{\max}$, and $[{\sf D}]_\eta = {\sf D} \cap [\mathbb{R}^n]_\eta$ for some quantization parameters $\eta \in \mathbb{R}^+$.

Meanwhile, the size of the set of states for the abstractions provided by Theorem 5.1 and Corollary 5.2 is at most

$$\begin{aligned} |X_*| = &\left(\left|[{\sf D}]_\eta\right| + 1\right)^{N^{\rm sc}_{\max}} \cdot \left|[{\sf U}]_\mu\right|^{N^{\rm ca}_{\max}} \cdot \\ &(N^{\rm sc}_{\max} - N^{\rm sc}_{\min} + 1)^{N^{\rm sc}_{\max}} \cdot (N^{\rm ca}_{\max} - N^{\rm ca}_{\min} + 1)^{N^{\rm ca}_{\max}} \end{aligned}$$

where $[{\sf U}]_\mu = {\sf U} \cap [\mathbb{R}^m]_\mu$ for some quantization parameters $\mu \in \mathbb{R}^+$. Note that there may exist some states of X_* that are not reachable from any of the initial states $x_{*0} \in X_{*0}$ due to the combination of the delays in both channels of the network, and hence, one can exclude them from the set of states X_* without loss of generality. Therefore, the actual size of the state set X_* may be less than the aforementioned computed ones.

One can easily verify that the size of the symbolic models proposed in [13] and [14] is at most

$$\begin{aligned} \left|S_\star(\widetilde{\Sigma})\right| &= |X_\star| \cdot |[{\sf U}]_\mu| \cdot (N_{\max} - N_{\min} + 1) \cdot K \\ &= \Big(\sum_{i\in\{\{1\}\cup[N_{\min};N_{\max}]\}} \left|[{\sf D}]_\eta\right|^i\Big) \\ &\quad \cdot |[{\sf U}]_\mu| \cdot (N_{\max} - N_{\min} + 1) \cdot K \qquad \text{(VII.1)} \end{aligned}$$

where K is the maximum number of u-successors of any state of the symbolic model $S_{\sf q}(\Sigma)$ for $u \in [{\sf U}]_\mu$. Note that with the results proposed in [16], one has $K = 1$ because $S_{\sf q}(\Sigma)$ is a deterministic system, while with the ones proposed in [17], one has $K \geq 1$ because $S_{\sf q}(\Sigma)$ is a nondeterministic system and the value of K depends on the functions β and γ in (II.2)—see [17] for more details. The size of the symbolic models provided in this paper is at most

$$\begin{aligned} \left|S_*(\widetilde{\Sigma})\right| &= |X_*| \cdot |[{\sf U}]_\mu| \cdot (N^{\rm sc}_{\max} - N^{\rm sc}_{\min} + 1) \\ &\quad \cdot (N^{\rm ca}_{\max} - N^{\rm ca}_{\min} + 1) .K \\ &= \left(\left|[{\sf D}]_\eta\right| + 1\right)^{N^{\rm sc}_{\max}} \cdot \left|[{\sf U}]_\mu\right|^{N^{\rm ca}_{\max}+1} \\ &\quad \cdot (N^{\rm sc}_{\max} - N^{\rm sc}_{\min} + 1)^{N^{\rm sc}_{\max}+1} \\ &\quad \cdot (N^{\rm ca}_{\max} - N^{\rm ca}_{\min} + 1)^{N^{\rm ca}_{\max}+1} \cdot K \qquad \text{(VII.2)} \end{aligned}$$

with the same K as in (VII.1). The symbolic model $S_*(\widetilde{\Sigma})$ can have a smaller size for some large values of $N_{\max}$ and for $\left|[{\sf D}]_\eta\right| >> \left|[{\sf U}]_\mu\right|$, as depicted in Fig. 6 (upper panel) by fixing $N^{\rm ca}_{\max} = N^{\rm sc}_{\max} = 6$ and $N^{\rm ca}_{\min} = N^{\rm sc}_{\min} = 1$. On the other hand, the symbolic model $S_\star(\widetilde{\Sigma})$ can have a smaller size for some large values of $\left|[{\sf U}]_\mu\right|$ and of $N^{\rm ca}_{\max} - N^{\rm ca}_{\min}$ (or $N^{\rm sc}_{\max} - N^{\rm sc}_{\min}$), as depicted in Fig. 6 (lower panel) by fixing $|\,[{\sf D}]_\eta\,| = 10^7$.

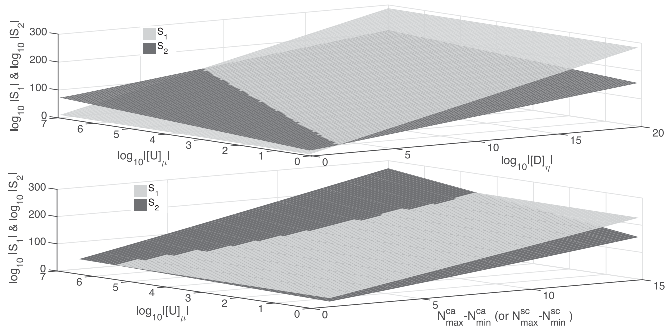

Fig. 6. Upper panel: sizes of $S_\star(\widetilde{\Sigma})$ and $S_*(\widetilde{\Sigma})$ for different values of $|[\mathsf{D}]_\eta|$ and $|[\mathsf{U}]_\mu|$, where $N_{\max}^{\mathsf{ca}} = N_{\max}^{\mathsf{sc}} = 6$, $N_{\min}^{\mathsf{ca}} = N_{\min}^{\mathsf{sc}} = 1$, and $S_1 = S_\star(\widetilde{\Sigma})$ and $S_2 = S_*(\widetilde{\Sigma})$. Lower panel: sizes of $S_\star(\widetilde{\Sigma})$ and $S_*(\widetilde{\Sigma})$ for different values of $|[\mathsf{U}]_\mu|$ and of $N_{\max}^{\mathsf{ca}} - N_{\min}^{\mathsf{ca}}$ (or $N_{\max}^{\mathsf{sc}} - N_{\min}^{\mathsf{sc}}$), where $|[\mathsf{D}]_\eta| = 10^7$.

Note that in the special case when $N_{\max}^{\mathrm{sc}} = N_{\min}^{\mathrm{sc}} = 1$, the dummy symbol q is not necessary in the definition of X_*; hence

$$|X_*| = \left|[\mathsf{D}]_\eta\right| \cdot \left|[\mathsf{U}]_\mu\right|^{N_{\max}^{\mathsf{ca}}} \cdot (N_{\max}^{\mathsf{ca}} - N_{\min}^{\mathsf{ca}} + 1)^{N_{\max}^{\mathsf{ca}}}. \tag{VII.3}$$

Remark 7.1: In [13, Remark 5.2], the authors suggest a more concise representation for their proposed finite abstractions of NCS, in order to reduce the space complexity. However, this representation is only applicable if the plant Σ is δ-ISS. Hence, for general classes of plants Σ in the NCS, the approach proposed in this work can be more appropriate in terms of the size of the abstractions, particularly for large values of $N_{\max}$ and for $\left|[\mathsf{D}]_\eta\right| >> \left|[\mathsf{U}]_\mu\right|$.

Remark 7.2: One can readily see in the example section that the computation time and memory required for computing symbolic abstractions of NCSs using the proposed method here are several orders of magnitude smaller than those required using techniques in [13] and [14]. The main reason for this is because modular construction of abstractions as proposed in this paper is highly favored by the binary decision diagram (BDD) data structure, which compactly represents both sets of states and the transition relation between these states.

VIII. Example

In this section, we present some case studies where we construct symbolic models of the NCS from the symbolic models of the plants inside them. We consider the setup presented in Section VI in order to refine the constructed symbolic controllers in closed-loop fashion. First, we present results for the construction of symbolic models of the NCS for several systems. Then, we provide an example where a dynamic controller is synthesized using the derived symbolic model of the NCS. The synthesized controller is simulated in closed-loop fashion using both MATLAB and OMNeT++ [38]. The computation of the abstract systems $S'_*(\widetilde{\Sigma})$ (cf., Section VI-B) and the symbolic controllers have been implemented by the software tool SENSE [39].

A. Symbolic Models of NCSs From the Ones of Plants in Them

We use the tool SCOTS [35] to construct symbolic models of the plants which are stored as BDD objects. The BDD objects are fed as inputs to the tool SENSE along with NCS delay bounds to construct symbolic models of NCS. Notice that the tool SENSE constructs symbolic models of NCS directly by operating with BDD objects of the symbolic models of the plants. This results in a large reduction in the computation time in comparison with constructing them from scratch, which is the case using the techniques proposed in [13] and [14] (cf., see later for a comparison for some of the case studies). Table I summarizes the results for different network delay configurations. Six case studies are considered. For each case study, we show the size of the symbolic model of the plant. For different network delay configurations $(N_{\max}^{\mathrm{sc}}, N_{\max}^{\mathrm{ca}})$, we show the size of the symbolic models of NCS, the time in seconds required to construct them, and the memory in kilobytes used to store them. First, we consider an already given symbolic model of the plant (denoted by SM) consisting of 13 states and 26 transitions. Then, we consider a plant as a double integrator (denoted by DI) inside an NCS, where its dynamic is given by

$$\Sigma : \left\{ \dot{\xi} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \upsilon \right.$$

with the set of states restricted to $[0, 3.2] \times [-1.5, 1.5]$, state quantization parameters as $(0.2, 0.3)$, input set restricted to $[-0.3, 0.3]$, input quantization parameter of 0.2, and sampling time $\tau = 0.3$. The third case study, denoted by Robot, corresponds to a mobile robot whose dynamics is given by [40]

$$\Sigma \left\{ \dot{\xi} = \begin{bmatrix} \upsilon_1 \\ \upsilon_1 \end{bmatrix} \right. .$$

The states represent the position of the robot. We consider the state set restricted to $[0, 63] \times [0, 63]$ and state quantization parameter as 1. The input set is restricted to $[-1, 1] \times [-1, 1]$ with input quantization parameter of 1, and sampling time is $\tau = 1$. The last three case studies, denoted by Vehicle1, Vehicle2, and Vehicle3, respectively, correspond to a vehicle whose dynamics is given by

$$\Sigma \left\{ \dot{\xi} = \begin{bmatrix} \upsilon_1 \cos(\alpha + \xi_3) \cos(\alpha)^{-1} \\ \upsilon_1 \sin(\alpha + \xi_3) \cos(\alpha)^{-1} \\ \upsilon_1 \tan(\upsilon_2) \end{bmatrix} \right. \tag{VIII.1}$$

where $\alpha = \arctan(\tan(\upsilon_2)/2)$. The first and second states represent the position of the vehicle, while the third represents the heading angle. The control inputs represent rear wheel velocity and the steering angle. We consider state quantization parameter as 0.2, input set restricted to $[-1, 1] \times [-1, 1]$, and input set quantization parameter as 0.3, and a sampling time of $\tau = 0.3$ for the last three case studies. We consider the state set restricted to $[0, 6] \times [0, 5] \times [-3.54, 3.54]$ for Vehicle1 and Vehicle2 case studies. The state set is

TABLE I
RESULTS FOR CONSTRUCTING SYMBOLIC MODELS OF THE NCS USING SYMBOLIC MODELS OF THEIR PLANTS

Case Study	$\lvert S_{\mathsf{q}}(\Sigma)\rvert$		(2, 2)	(2, 3)	(2, 4)	(2, 5)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(4, 2)	(4, 3)
SM	26	$\lvert S_*(\widetilde{\Sigma})\rvert$	214	422	838	1670	430	846	1678	3342	862	1694
		Time (seconds)	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
		Memory (kB)	1.9	2.9	1.2	1.2	3.5	1.6	1.6	1.6	1.9	1.9
DI	2039	$\lvert S_*(\widetilde{\Sigma})\rvert$	170272	681088	2.7×10^6	1.1×10^7	900384	3.6×10^6	1.4×10^7	5.8×10^7	4.8×10^6	1.9×10^7
		Time (seconds)	<1	<1	<1	<1	<1	<1	<1	<1	<1	<1
		Memory (kB)	2.0	2.4	3.1	2.9	3.0	2.9	3.1	3.2	5.2	4.3
Robot	29280	$\lvert S_*(\widetilde{\Sigma})\rvert$	6.4×10^7	1.0×10^9	1.6×10^{10}	2.6×10^{11}	5.6×10^8	9.0×10^9	3.4×10^{11}	2.3×10^{12}	4.9×10^9	7.8×10^{10}
		Time (seconds)	<1	<1	<1	<1	<1	<1	<1	<1	1.4	1.6
		Memory (kB)	15	14	17	16	16	21	22	19	35	33
Vehicle1	9.1×10^6	$\lvert S_*(\widetilde{\Sigma})\rvert$	2.4×10^{12}	1.5×10^{14}	9.7×10^{15}	6.2×10^{17}	1.5×10^{14}	9.8×10^{15}	6.3×10^{17}	4.0×10^{19}	1.0×10^{16}	6.4×10^{17}
		Time (seconds)	70	129	89	107	5587	944	1598	1705	7399	6182
		Memory (kB)	734	992	961	881	7577.6	5529.6	8294.4	7782.4	9932.8	9523.2
Vehicle2	9.9×10^6	$\lvert S_*(\widetilde{\Sigma})\rvert$	2.7×10^{12}	1.7×10^{14}	1.1×10^{16}	7.0×10^{17}	1.8×10^{14}	1.2×10^{16}	7.4×10^{17}	4.6×10^{19}	1.2×10^{16}	7.91×10^{17}
		Time (seconds)	66.8	104.6	107	61.8	833	781	1247	4363	5137	8462
		Memory (kB)	826	683	733	709	5222.4	4710.4	4608	11 059.2	8396.8	8806.4
Vehicle3	1.89×10^7	$\lvert S_*(\widetilde{\Sigma})\rvert$	4.2×10^{12}	2.7×10^{14}	1.7×10^{16}	1.1×10^{18}	2.3×10^{14}	1.4×10^{16}	9.3×10^{17}	5.9×10^{19}	1.3×10^{16}	7.94×10^{17}
		Time (seconds)	273.3	285	238.4	173	22 344	54 919	27 667	36 467	39 065	145 390
		Memory (kB)	1638.4	1945.6	1843.2	1945.6	23040	40 652.8	30 208	22 425.6	21 094.4	36 556.3

restricted to $[0,10]\times[0,10]\times[-3.54,3.54]$ for the Vehicle3 case study. Some parts of the state sets of the last four case studies were removed to represent obstacles that need to be avoided when synthesizing the symbolic controllers. The symbolic models were constructed using a PC (Intel Core i7 3.6 GHz and 32 GB RAM). The CUDD library [41] was used to operate with BDDs. Note that the inconsistencies in the execution time and storage memory reported in Table I are due to the heuristic algorithms implemented in the CUDD library for operating with BDDs to automatically reorder binary variables for optimizing BDD operations. We also implemented the construction of symbolic models of NCS using the schemes proposed in [13] and [14]. The computation time and memory storage for the construction of symbolic model for NCS containing DI with delay parameters $N^{\mathrm{sc}}_{\max}=N^{\mathrm{ca}}_{\max}=2$ amounted to 1.17 s and 42.6 kB, respectively. For the Vehicle1 case with delay parameters $N^{\mathrm{sc}}_{\max}=N^{\mathrm{ca}}_{\max}=2$, the computation time amounted to more than two days and the memory usage exceeded 32 GB. This shows that the computation times and memory required to construct symbolic models using the schemes in [13] and [14] are several orders of magnitude more than those using the proposed scheme in this paper which amounted to 0.019 and 117.4 s, respectively (including the computation time required by the tool SCOTS to construct the symbolic models of the plants inside the NCS), while the storage memory is already reported in Table I.

B. Controller Synthesis and Refinement: The Robot Case

We consider the third case study from Table I with the network delays $(N^{\mathrm{sc}}_{\max}=2, N^{\mathrm{ca}}_{\max}=2)$. The control objective is to enforce the robot to infinitely often visit two target sets of states described by propositions $\mathsf{Target}1$ and $\mathsf{Target}1$, which are defined by the hyperintervals $[5,15]\times[45,55]$ and $[45,55]\times[5,15]$, respectively. Moreover, the robot needs to avoid a set of nine obstacles defined by the propositions $\mathrm{Obstacle}_i$, $i\in\{1,\ldots,9\}$, which are defined by the hyperintervals $[5,15]\times[20,22]$, $[15,17]\times[5,22]$, $[48,50]\times[45,60]$, $[51,58]\times[45,47]$, $[27,36]\times[20,45]$, $[44,49]\times[27,36]$, $[27,36]\times[52,57]$, $[27,36]\times[5,10]$, and $[14,19]\times[27,36]$, respectively. This control objective can be described by the following LTL formula:

$$\psi=\left(\bigwedge_{i=1}^{9}\Box(\neg\mathsf{Obstacle}_i)\right)\wedge\Box\Diamond(\mathsf{Target}1)\wedge\Box\Diamond(\mathsf{Target}2).$$

The controller was synthesized using fixed-point computations as implemented in SENSE. Remark that the resulting controller is a dynamic controller with two discrete states. The computation of the symbolic controller amounted to 4.7 s. Fig. 7 shows the closed-loop simulation of the NCS. For a more realistic simulation environment, we consider OMNeT++ [38], a common simulation framework for networks. Communication channels are modeled using a random propagation-delay communication channels in OMNeT++. Fig. 8 shows the closed-loop simulation results in OMNeT++. We make use of the animation capabilities of OMNeT++ to visualize both packet transfers over the network as well as the movement of the robot through the state set as illustrated in [39]. Controller synthesis and refinement for the vehicle dynamic in (VIII.1), for a configuration of network delays, and for an LTL specification are provided in [39].

IX. DISCUSSION AND CONCLUSION

In this paper, we have provided a construction of symbolic models for NCS, subject to the following nonidealities: variable communication delays, quantization errors, packet losses, and limited bandwidth. This novel approach is practically relevant, since it can leverage any existing symbolic model for

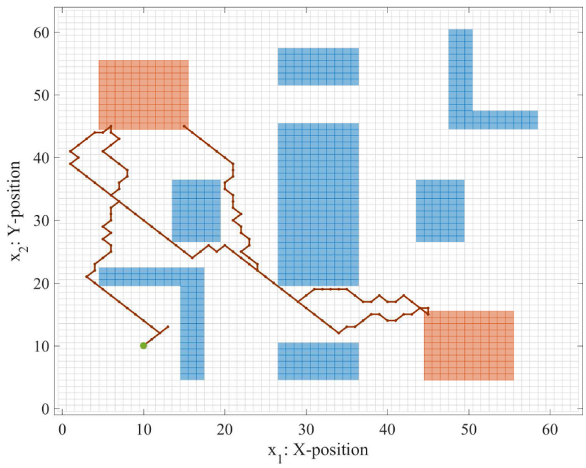

Fig. 7. Closed-loop simulation of the NCS with the robot system in MATLAB. The target sets are indicated with the red boxes and obstacles with the blue boxes.

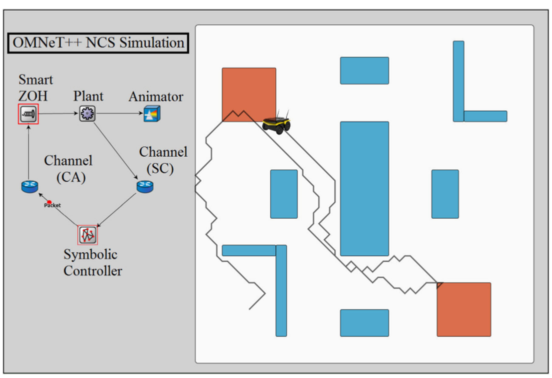

Fig. 8. Closed-loop simulation of the NCS with the robot system in OMNeT++. On the left, we illustrate how the packets move between different parts of the network. On the right, the movement of the robot over the state set is illustrated.

the plant and, in particular, is not limited to grid-based ones and extendible to work over stochastic plants—both features are current focus of active investigation elsewhere. Furthermore, this approach can be applied to treat any specification expressed as a formula in LTL (cf., the example section) or as an automaton on infinite strings, without requiring any additional reformulation.

Future work will concentrate on the following goals:

1) the construction of symbolic models for the NCS with explicit probabilistic structure over the transmission intervals, communication delays, and packet dropouts;
2) the construction of symbolic models for still more general NCSs, by considering additional network nonidealities, in particular time-varying sampling and transmission intervals;
3) the study of interconnections and synthesis employing the different outputs enabled by our abstractions at the sensor and controller side.

Acknowledgment

The authors would like to thank M. Rungger for fruitful technical discussions over Section V-A.

References

[1] W. P. M. H. Heemels and N. van de Wouw, "Stability and stabilization of networked control systems," in *Networked Control Systems (ser. Lect. Notes Control Inf. Sci.)*, vol. 406, A. Bemporad, W. P. M. H. Heemels, and M. Johansson, Eds. London, U.K.: Springer, 2010, pp. 203–253.

[2] N. W. Bauer, P. J. H. Maas, and W. P. M. H. Heemels, "Stability analysis of networked control systems: A sum of squares approach," *Automatica*, vol. 48, no. 8, pp. 1514–1524, 2012.

[3] H. Gao, T. Chen, and J. Lam, "A new delay system approach to network-based control," *Automatica*, vol. 44, no. 1, pp. 39–52, 2008.

[4] R. Alur, A. D'Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss, "Compositional modeling and analysis of multi-hop control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2345–2357, Oct. 2011.

[5] D. Antunes, J. P. Hespanha, and C. Silvestre, "Volterra integral approach to impulsive renewal systems: Application to networked control," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 607–619, Mar. 2012.

[6] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1575–1580, Jul. 2009.

[7] N. van de Wouw, D. Nesic, and W. P. M. H. Heemels, "A discrete-time framework for stability analysis of nonlinear networked control systems," *Automatica*, vol. 48, no. 6, pp. 1144–1153, Jun. 2012.

[8] D. Nesic and D. Liberzon, "A unified framework for design and analysis of networked and quantized control systems," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 732–747, Apr. 2009.

[9] C. Baier and J. P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, Apr. 2008.

[10] P. Tabuada, *Verification and Control of Hybrid Systems, A Symbolic Approach*. New York, NY, USA: Springer, 2009.

[11] G. Reissig, A. Weber, and M. Rungger, "Feedback refinement relations for the synthesis of symbolic controllers," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1781–1796, Apr. 2017.

[12] O. Maler, A. Pnueli, and J. Sifakis, "On the synthesis of discrete controllers for timed systems," *in Proc. 12th Annu. Symp. Theor. Aspects Comput. Sci.*, 1995, vol. 900, pp. 229–242.

[13] A. Borri, G. Pola, and M. Di Benedetto, "A symbolic approach to the design of nonlinear networked control systems," in *Proc. 15th Int. Conf. Hybrid Syst. Comput. Control*, Apr. 2012, pp. 255–264.

[14] A. Borri, G. Pola, and M.Di Benedetto, "Integrated symbolic design of unstable nonlinear networked control systems," in *Proc. 51st IEEE Conf. Decision Control*, Dec. 2012, pp. 1374–1379.

[15] A. Borri, G. Pola, and M. Di Benedetto, "Symbolic control design of nonlinear networked control systems," *arXiv: 1404.0237*, Mar. 2016.

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

[17] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1804–1809, Jul. 2012.

[18] M. Zamani, I. Tkachev, and A. Abate, "Bisimilar symbolic models for stochastic control systems without state-space discretization," in *Proc. 17th Int. Conf. Hybrid Syst. Comput. Control*, Apr. 2014, pp. 41–50.

[19] M. Zamani, I. Tkachev, and A. Abate, "Towards scalable synthesis of stochastic control systems," *Discrete Event Dyn. Syst.*, vol. 27, no. 2, pp. 341–369, Jul. 2017.

[20] B. Yordanov, J. Tumova, I. Cerna, J. Barnat, and C. Belta, "Formal analysis of piecewise affine systems through formula-guided refinement," *Automatica*, vol. 49, no. 1, pp. 261–266, Jan. 2013.

[21] M. Zamani, P. M. Esfahani, A. Abate, and J. Lygeros, "Symbolic models for stochastic control systems without stability assumptions," in *Proc. Eur. Control Conf.*, Jul. 2013, pp. 4257–4262.

[22] M. Zamani, P. M. Esfahani, R. Majumdar, A. Abate, and J. Lygeros, "Symbolic control of stochastic systems via approximately bisimilar finite abstractions," *IEEE Trans. Autom. Control (Special Issue Control Cyber-Phys. Syst.)*, vol. 59, no. 12, pp. 3135–3150, Dec. 2014.

[23] M. Zamani, M. Mazo Jr., and A. Abate, "Finite abstractions of networked control systems," in *Proc. 53rd IEEE Conf. Decision Control*, Dec. 2014, pp. 95–100.

[24] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, vol. 6, 2nd ed. New York, NY, USA: Springer, 1998.

[25] D. Angeli and E. D. Sontag, "Forward completeness, unboundedness observability, and their Lyapunov characterizations," *Syst. Control Lett.*, vol. 38, pp. 209–217, 1999.

[26] D. Angeli, "A Lyapunov approach to incremental stability properties," *IEEE Trans. Autom. Control*, vol. 47, no. 3, pp. 410–421, Mar. 2002.

[27] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 782–798, May 2007.

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

[29] H. Dai and R. Han, "Tsync: A lightweight bidirectional time synchronization service for wireless sensor networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 8, no. 1, pp. 125–139, 2004.

[30] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[31] A. Girard, "Synthesis using approximately bisimilar abstractions: State-feedback controllers for safety specifications," in *Proc. 13th Int. Conf. Hybrid Syst. Comput. Control*, Apr. 2010, pp. 111–120.

[32] M. Zamani, M. Mazo Jr., M. Khaled, and A. Abate, "Symbolic models for networked control systems," *arXiv:1401.6396*, 2016.

[33] A. Girard, "Low-complexity quantized switching controllers using approximate bisimulation," *Nonlinear Anal. Hybrid Syst.*, vol. 10, pp. 34–44, Nov. 2013.

[34] M. Khaled, M. Rungger, and M. Zamani, "Symbolic models of networked control systems: A feedback refinement relation approach," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2016, pp. 187–193.

[35] M. Rungger and M. Zamani, "SCOTS: A tool for the synthesis of symbolic controllers," in *Proc. 19th Int. Conf. Hybrid Syst. Comput. Control*, Apr. 2016, pp. 99–104.

[36] R. Ehlers and V. Raman, "Slugs: Extensible GR(1) synthesis," in *Computer Aided Verification (CAV) (ser. Lect. Notes Comput. Sci.)*, vol. 9780, S. Chaudhuri and A. Farzan, Eds. New York, NY, USA: Springer, Jul. 2016, pp. 333–339.

[37] K. Chatterjee, L. Doyen, T. A. Henzinger, and J. F. Raskin, "Algorithms for omega-regular games with imperfect information," in *Computer Science Logic (ser. Lect. Notes Comput. Sci.)*, vol. 4207, Z. Esik, Ed. Berlin Germany: Springer, 2006, pp. 287–302.

[38] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simul. Tools Tech. Commun. Netw. Syst. Workshops*, 2008, pp. 1–10.

[39] M. Khaled, M. Rungger, and M. Zamani, SENSE: A tool for the symbolic controller construction and controller synthesis for NCS, Nov. 2016. [Online]. Available: http://www.hcs.ei.tum.de/software/sense

[40] C. Belta and V. Kumar, "Abstractions and control for groups of robots," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 865–875, Oct. 2004.

[41] F. Somenzi, *CUDD: CU Decision Diagram Package*, 3rd ed., Dec. 2015.

Majid Zamani (M'12–SM'16) received the B.Sc. degree in electrical engineering from Isfahan University of Technology, Isfahan, Iran, in 2005, the M.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2007, and the M.A. degree in mathematics and the Ph.D. degree in electrical engineering from the University of California, Los Angeles, CA, USA, both in 2012.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany. Between 2012 and 2013, he was a Postdoctoral Researcher with the Delft Centre for Systems and Control, Delft University of Technology, Delft, Netherlands, where he was an Assistant Professor in the Design Engineering Department from 2013 to 2014. His research interests include verification and control of hybrid systems, embedded control software synthesis, networked control systems, and incremental properties of nonlinear control systems.

Manuel Mazo, Jr. (S'99–M'11) received the Telecommunications Engineering "Ingeniero" degree from the Polytechnic University of Madrid, Madrid, Spain, and a "Civilingenjör" degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, both in 2003, and the Ph.D. and M.Sc. degrees in electrical engineering from the University of California, Los Angeles (UCLA), CA, USA, in 2010 and 2007, respectively.

He has been an Assistant Professor with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, since 2012. Between 2010 and 2012, he held a joint postdoctoral position with the University of Groningen and the Innovation Centre INCAS3 (The Netherlands). His main research interests include the formal study of problems emerging in modern control system implementations and, in particular, the study of networked control systems and the application of formal verification and synthesis techniques to control.

Dr. Mazo has received a University of Newcastle Research Fellowship (2005), the Spanish Ministry of Education/UCLA Fellowship (2005–2009), and the Henry Samueli Scholarship from the UCLA School of Engineering and Applied Sciences (2007/2008).

Mahmoud Khaled (S'14) received the B.Sc. degree in computer and systems engineering and the M.Sc. degree in electrical engineering from the Faculty of Engineering, Minya University, Minya, Egypt, in 2009 and 2014, respectively, and is currently pursuing the Ph.D. degree in electrical and computer engineering at the Technical University of Munich, Munich, Germany.

His current research interests include the broad area of formal methods in control, including control of cyber-physical systems and automated synthesis of controllers.

Alessandro Abate (S'02–M'08) received the Laurea degree in electrical engineering (Hons.) from the University of Padova, Padova, Italy, in 2002. He then received the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 2004 and 2007, respectively.

He is an Associate Professor with the Department of Computer Science, University of Oxford, Oxford, U.K., and a Fellow of the Alan Turing Institute, London, U.K. While studying, he was an International Fellow in the CS Lab at SRI International, Menlo Park, CA, USA. Following his Ph.D., he was a Postdoctoral Researcher with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, USA, working with C. Tomlin on systems biology in affiliation with the Stanford School of Medicine. From 2009 to 2013, he was an Assistant Professor with the Delft Center for Systems and Control, Delft University of Technology, working with his research group on verification and synthesis over complex systems and on smart energy applications.