Symbolic Control Design of Nonlinear Networked Control Systems

Alessandro Borri, Giordano Pola and Maria Domenica Di Benedetto

Abstract

Networked Control Systems (NCS) are distributed systems where plants, sensors, actuators and controllers communicate over shared networks. Non-ideal behaviors of the communication network include variable sampling/transmission intervals and communication delays, packet losses, communication constraints and quantization errors. NCS have been the object of intensive study in the last few years. However, due to the inherent complexity of NCS, current literature focuses on only a subset of these non-idealities and mostly considers stability and stabilizability problems. Recent technology advances indeed demand that different and more complex control objectives are considered. In this paper we present first a general model of NCS, including all the non-idealities of the communication network; then, we propose a symbolic model approach to the control design with objectives expressed in terms of non-deterministic transition systems. The presented results are based on recent advances in symbolic control design of hybrid and continuous control systems. An example in the context of robot motion planning with remote control is included, showing the effectiveness of the approach taken.

1 Introduction

Networked Control Systems (NCS) are complex, heterogeneous, spatially distributed systems where physical processes interact with distributed computing units through non-ideal communication networks. In the past, NCS were limited in the number of computing units and in the complexity of the interconnection

\*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 NoE.

\[\text{Alessandro Borri is with the Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche (IASI-CNR), 00185 Rome, Italy, alessandro.borri@iasi.cnr.it.}\]

\[\text{Giordano Pola and Maria Domenica Di Benedetto are with the Department of Information Engineering, Computer Science and Mathematics, Center of Excellence for Research DEWS, University of L'Aquila, 67100, L'Aquila, Italy, \{giordano.pola,mariadomenica.dibenedetto@univaq.it\}.}\]
network so that it was possible to obtain reasonable performance by aggregating subsystems that were locally designed and optimized. However, the growth of complexity of the physical systems to control, together with the continuous increase in functions that these systems must perform, requires today to adopt a unified design approach where different disciplines (e.g., control systems engineering, computer science, software engineering and communication engineering) should come together to reach new levels of performance. The heterogeneity of the subsystems that are to be connected in a NCS make the control of these systems a hard but challenging task. NCS have been the focus of much recent research in the control community: Murray et al. in [1] presented control over networks as one of the important future directions for control. Following [2], the most important non-idealities considered in the study of NCS are: (i) variable sampling/transmission intervals; (ii) variable communication delays; (iii) packet dropouts caused by the unreliability of the network; (iv) communication constraints (scheduling protocols) managing the possibly simultaneous transmissions over the shared channel; (v) quantization errors in the digital transmission with finite bandwidth.

There are two approaches to manage such non-idealities: the deterministic approach, which assumes worst-case (deterministic) bounds on the aforementioned imperfections, and the stochastic approach, which provides a stochastic description of the non-ideal communication network. We focus our attention on the deterministic methods, which can be further distinguished according to the modeling assumptions and the controller synthesis for NCS: a) the discrete-time approach (see e.g. [3], [4]) considers discrete-time controllers and plants; b) the sampled-data approach (see e.g. [5], [6]) assumes discrete-time controllers and continuous-time (sampled-data) plants; c) the continuous-time (emulation) approach (see e.g. [7], [8]) focuses on continuous-time controllers and continuous-time (sampled-data) plants. In the deterministic approach, results obtained during the past few years are mostly about stability and stabilizability problems, see e.g. [9, 2, 10], with results that depend on the method considered and the assumptions on the non-ideal communication infrastructure. In addition, current approaches in the literature take into account only a subset of these non-idealities. As reviewed in [2], for example, [11] studies imperfections of type (i), (iv), (v), [12], [13], [14] consider simultaneously (i), (ii), (iii), [15] focuses on (i), (iii), (iv), while [16] manages (ii), (iii) and (v). Three types of non-idealities, namely (i), (ii), (iv), are considered for example in [13], [14], [7]. In [15], the five non-idealities are dealt with but small delay and other restrictive assumptions are considered. Finally, novel results in the stability analysis of NCS can be found in [16], [17], [18], [19]. However, existing results do not address control design of NCS with complex specifications, as for example safety and liveness properties, obstacle avoidance, fairness constraints, language and logic specifications. This paper follows the deterministic approach and constitutes a first step towards a unified theory for NCS control design where the most relevant non-idealities of the communication and computing infrastructures can be dealt with. The approach taken is based on the use of discrete abstractions of continuous and hybrid systems [20, 21]. This approach is a sound paradigm to solve control problems where software
and hardware interact with the physical world and, to address a wealth of novel specifications, which are difficult to enforce by means of conventional control design methods. Examples of such specifications include logic specifications expressed in linear temporal logic or automata. Central to this approach is the construction of symbolic models, which are abstract descriptions of complex systems where a symbol corresponds to an “aggregate” of continuous states and a symbolic control label to an “aggregate” of continuous control inputs. Several classes of dynamical and control systems that admit equivalent symbolic models have been identified in the literature. Within the class of hybrid automata we recall timed automata [22], rectangular hybrid automata [23], and o-minimal hybrid systems [24, 25]. Early results for classes of control systems were based on dynamical consistency properties [26], natural invariants of the control system [27], l-complete approximations [28], and quantized inputs and states [29, 30]. Recent results include work on controllable discrete-time linear systems [31], piecewise-affine and multi-affine systems [32, 33], set-oriented discretization approach for discrete-time nonlinear optimal control problems [34], abstractions based on convexity of reachable sets [35], incrementally stable and incrementally forward complete nonlinear control systems with and without disturbances [36, 37, 38, 39], switched systems [40] and time-delay systems [41, 42]. The interested reader is referred to [43, 21] for an overview on recent advances in this domain.

In this paper we address the control design of a fairly general model of NCS with complex specifications. The main contributions of this paper are:

- **A general model of NCS.** We consider NCS where the plant is a continuous–time nonlinear control system, the computing units are modelled by finite state transition systems, and the communication network non-idealities are quantization errors, time-varying delay in accessing the network, time-varying delay in delivering messages through the network, limited bandwidth and packet dropouts. The proposed model covers non-idealities (i)-(v) in NCS and, due to its flexibility, can embed specific communication protocols, data compression and encryption in the message delivery, and scheduling rules in the communication network and computing units.

- **A symbolic model approach to the control design of NCS.** We propose symbolic models that approximate NCS in the sense of alternating approximate (bi)simulation with arbitrarily good accuracy. More specifically, under the assumption of existence of an incremental forward complete Lyapunov function for the plant of the NCS, we derive symbolic models approximating the NCS in the sense of alternating approximate simulation; for incrementally stable plants we derive symbolic models that approximate the NCS in the sense of alternating approximate bisimulation. The first result is important because it does not require the stability of the open-loop NCS while the second result is important because it provides a completeness property in the control design: if a solution does not exist for the given control problem (with desired accuracy) for the symbolic model, no control strategy exists for the original NCS. Building upon these
symbolic models, we address the NCS control design where specifications are expressed in terms of transition systems. Given a NCS and a specification, a symbolic controller is derived such that the controlled system meets the specification in the presence of the considered non-idealities in the communication network.

This paper follows the approach proposed in [36, 37] based on the construction of symbolic models for nonlinear control systems. It provides an extended version of the preliminary results published in [44, 45], including a comprehensive NCS modeling, extensions and full proofs of the technical results and an example in the context of robot motion planning with remote control. Moreover, while in [44, 45] controllers are assumed to be static, we consider here the general class of dynamic controllers.

The paper is organized as follows. In Section 2 the notation is introduced. In Section 3 a model is proposed for a general class of nonlinear NCS. In Section 4 symbolic models approximating NCS are derived. In Section 5 symbolic control design is addressed. An example of application of the proposed results is included in Section 6. Finally, Section 7 offers some concluding remarks and outlook for future work. The Appendix recalls some technical notions that are instrumental in the paper.

2 Notation and preliminary definitions

Notation. The symbols \( \mathbb{N}, \mathbb{N}_0, \mathbb{Z}, \mathbb{R}, \mathbb{R}^-, \mathbb{R}^+ \) and \( \mathbb{R}_0^+ \) denote the set of natural, nonnegative integer, integer, real, negative real, positive real, and nonnegative real numbers, respectively. The cardinality of a set \( A \) is denoted by \( |A| \). 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 a pair of sets \( A \) and \( B \) and a relation \( R \subseteq A \times B \), the symbol \( R^{-1} \) denotes the inverse relation of \( R \), i.e., \( R^{-1} = \{(b, a) \in B \times A : (a, b) \in R \} \). Given an interval \( [a, b] \subseteq \mathbb{R}_0^+ \), we denote by \( [a; b] \) the set \( [a, b] \cap \mathbb{N} \), if \( a \leq b \), and the empty set \( \emptyset \) otherwise. We denote the ceiling of a real number \( x \) by \( [x] = \min\{n \in \mathbb{Z} | n \geq x \} \). Given a vector \( x \in \mathbb{R}^n \) we denote by \( \|x\| \) the infinity norm and by \( \|x\|_2 \) the Euclidean norm of \( x \).

Preliminary definitions. A continuous function \( \gamma : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \) is said to belong to class \( K \) if it is strictly increasing and \( \gamma(0) = 0 \); a function \( \gamma \) is said to belong to class \( K_\infty \) if \( \gamma \in K \) and \( \gamma(r) \rightarrow \infty \) as \( r \rightarrow \infty \). 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 set \( A \) given as a finite union of hyperrectangles, i.e., \( A = \bigcup_{j \in \{1, J\}} A_j \), for some \( J \in \mathbb{N} \), where \( A_j = \prod_{i \in \{1, n\}} [a_{ji}^j, b_i^j] \subseteq \mathbb{R}^n \) with \( a_i^j < b_i^j \), and define \( \mu_A = \min_{j \in \{1, J\}} \mu_{A_j} \), where \( \mu_{A_j} = \min\{|b_i^j - a_i^j|, \ldots, |b_i^j - a_i^j|\} \). Following [37], for any \( \mu \leq \mu_A \) and any \( a \in A \) there exists \( b \in [A]_\mu \) such that \( \|a - b\| \leq \mu \). Given any \( a \in A \) and \( \mu \leq \mu_A \), in the sequel we denote by \( [a]_\mu \in [A]_\mu \) a vector such that \( \|a - [a]_\mu\| \leq \mu \).
3 Networked Control Systems

The class of NCS that we consider is depicted in Fig. 1. It consists of a nonlinear control system (the plant $P$), whose control loop is closed over a non-ideal communication network, taking into account the most important non-idealities commonly considered in the literature, including finite time-varying network delays, finite bandwidth, signal quantization, communications constraints due to shared access to the network, transmission overhead, finite computational resources and packet losses. A non-ideal network is placed both in the plant-to-controller branch and in the controller-to-plant branch of the loop. The analog-to-digital (sensor and quantizer) and digital-to-analog (ZoH) interfaces of the continuous plant allow the transmission of sensing and control digital samples over a channel with finite bandwidth. The symbolic controller provides quantized control samples depending on the value of the measured output. Our framework is inspired by the models reviewed in [2]. The sub-systems composing the NCS are described hereafter in more detail.

**Plant.** The direct branch of the network includes the plant $P$ that is a nonlinear control system in the form of:

$$
\begin{cases}
    \dot{x}(t) = f(x(t), u(t)), t \in \mathbb{R}_0^+,
    \\
    x \in X \subseteq \mathbb{R}^n, \quad x(0) \in X_0 \subseteq X, \quad u(\cdot) \in \mathcal{U},
\end{cases}
$$

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 a finite non-empty set \(U \subseteq [\mathbb{R}^m]_{\mu_U}\) for some \(\mu_U \in \mathbb{R}^+\). We suppose that the set \(X\) is in the form of a finite union of hyperrectangles. The function \(f : X \times U \rightarrow \mathbb{R}^n\) is assumed to be Lipschitz on compact sets with respect to the first argument. In the sequel we denote by \(x(t, x_0, u)\) the state reached by \(x(t)\) at time \(t\) under the control input \(u\) from the initial state \(x_0\); this point is uniquely determined, since the assumption on \(f\) ensures existence and uniqueness of trajectories. We assume that the control system \(P\) is forward complete, namely that every trajectory \(x(\cdot, x_0, u)\) of \(P\) is defined on an interval of the form \([a, \infty]\). Sufficient and necessary conditions for a control system to be forward complete can be found in [46]. In the remainder of the paper, we abuse notation by denoting the constant control input \(u(t) = u\) in the compact domain \([0, \tau]\) (for some \(\tau \in \mathbb{R}^+\)) by \(u\).

Sensor. On the right-hand side of the plant \(P\) in Fig. 1 a sensor is placed. We assume that:

(A.1) The sensor is synchronized with the plant and updates its output value at times that are integer multiples of \(\tau \in \mathbb{R}^+\), i.e. \(\hat{y}_s = x(s\tau, x_0, u)\), for some \(x_0 \in X_0\) and \(u \in U\), and any \(s \in \mathbb{N}_0\), where \(s\) is the index identifying the sampling interval (starting from 0).

The above synchronization assumption is not restrictive since the sensor is physically connected to the plant.

Quantizer. A quantizer follows the sensor. For simplicity, we assume that the quantizer is uniform, with accuracy \(\mu_X \in ]0, \hat{\mu}_X[\). The role of the quantizer is: i) to discretize the continuous-valued sensor measurement sequence \(\{\hat{y}_s\}_{s \in \mathbb{N}_0}\) to get the quantized sequence \(\{y_s\}_{s \in \mathbb{N}_0}\), with \(y_s = [\hat{y}_s]_{\mu_X}\); ii) to encode the signals into digital messages of length \(\lceil \log_2 |X|_{\mu_X} \rceil\) and to add overhead bits, resulting in the sequence of digital messages \(\{\bar{y}_s\}_{s \in \mathbb{N}_0}\). The transmission overhead takes into account the communication protocol, the packet headers, source and channel coding as well as data compression and encryption. We assume a fixed average relative overhead \(N_{pc}^+\) on each data bit; since data compression may be considered, the relative overhead \(N_{pc}^+\) may be negative. More precisely:

(A.2) \(N_{pc}^+\) bits are added per each bit of the digital signal encoding \(y_s\), i.e. the number of bits of message \(\bar{y}_s\) is \((1 + N_{pc}^+)\lceil \log_2 |X|_{\mu_X} \rceil\), for all \(s \in \mathbb{N}_0\).

Network. In the following, the index \(k \in \mathbb{N}\) denotes the current iteration in the feedback loop. Due to the non-idealities of the network, not all the output samples can be transmitted through the network. We assume that only one output sample per iteration is sent. In particular, \(\{M_k\}_{k \in \mathbb{N}} \subseteq \mathbb{N}\) denotes the subsequence of the sampling intervals when the output samples are sent through the network, i.e. at time \(M_k\tau\) the digital message \(\bar{y}_{M_k}\) encodes the output sample \(y_{M_k} = [x(M_k\tau)]_{\mu_X}\) and is sent (iteration \(k\)). We set \(M_1 = 0\). The communication network is characterized by the following features:

(Time-varying access to the network) The digital message \(\bar{y}_{M_k}\) cannot be sent instantaneously to the network, because the communication channel is assumed
to be a resource which is shared with other nodes or processes in the network. The policy by which a signal of a node is sent before or after a message of another node is managed by the network scheduling protocol selected. We assume that:

(A.3) The sequence \( \{\Delta_{k}^{\text{req,pc}}\}_{k \in \mathbb{N}} \) of network waiting times in the plant-to-controller branch of the feedback loop is bounded, i.e. \( \Delta_{k}^{\text{req,pc}} \in [\Delta_{\text{min}}^{\text{req}}, \Delta_{\text{max}}^{\text{req}}], \) for all \( k \in \mathbb{N}, \) for some \( \Delta_{\text{min}}^{\text{req}}, \Delta_{\text{max}}^{\text{req}} \in \mathbb{R}^+_0. \)

At time \( t_{k}^{\text{pc}} := M_{k} \tau + \Delta_{k}^{\text{req,pc}}, \) the message \( \bar{w}_{k} := \bar{y}_{M_{k}} \) is sent through the network.

(Limited bandwidth) In real applications, the capacity of the digital communication channel is limited and time-varying. We denote by \( B_{\text{min}}, B_{\text{max}} \in \mathbb{R}^+_0 \) the minimum and maximum capacities of the channel (expressed in bits per second, bps). In view of the binary coding and the transmission overhead (see Assumption (A.2)), we assume that:

(A.4) At iteration \( k, \) a delay \( \Delta_{k}^{\text{B,pc}} \in [\Delta_{\text{min}}^{\text{B,pc}}, \Delta_{\text{max}}^{\text{B,pc}}] \) due to the limited bandwidth is introduced in the plant-to-controller branch of the feedback loop, with \( \Delta_{\text{min}}^{\text{B,pc}} = (1+\sum_{i=1}^{N_{\text{pc}}} |\log_2 [\mathbb{X}]_{\mu_{\mathbb{X}}}|) / B_{\text{max}} \) and \( \Delta_{\text{max}}^{\text{B,pc}} = (1+\sum_{i=1}^{N_{\text{pc}}} |\log_2 [\mathbb{X}]_{\mu_{\mathbb{X}}}|) / B_{\text{min}}. \)

(Time-varying delivery of messages) The delivery of message \( \bar{w}_{k} \) may be subject to further delays, due to congestion phenomena in the network, etc. We assume that:

(A.5) The sequence \( \{\Delta_{k}^{\text{net,pc}}\}_{k \in \mathbb{N}} \) of network communication delays in the plant-to-controller branch of the feedback loop is bounded, i.e. \( \Delta_{k}^{\text{net,pc}} \in [\Delta_{\text{min}}^{\text{net}}, \Delta_{\text{max}}^{\text{net}}], \) for all \( k \in \mathbb{N}, \) for some \( \Delta_{\text{min}}^{\text{net}}, \Delta_{\text{max}}^{\text{net}} \in \mathbb{R}^+_0. \)

(Packet dropout) In real applications, one or more messages can be lost during the transmission, because of the unreliability of the communication channel. We assume that:

(A.6) The maximum number of successive packet dropouts is \( N_{\text{pd}}. \)

**Symbolic controller.** Unless message \( \bar{w}_{k} \) is lost, it is decoded into the quantized sensor measurement \( w_{k} \) and reaches the controller. The symbolic controller \( C \) is dynamic, remote and asynchronous with respect to the plant and is expressed as a Mealy machine:

\[
C : \begin{cases} \\
\xi_{k+1} \in f_{C}(\xi_{k}, w_{k}), \\
v_{k} = h_{C}(\xi_{k}, w_{k}),
\end{cases}
\]  

where \( \Xi \) is the state space of the controller and \( \text{Dom}_{C} \subseteq \Xi \times [\mathbb{X}]_{\mu_{\mathbb{X}}} \) is the domain of the functions \( f_{C} : \text{Dom}_{C} \rightarrow 2^{\Xi} \) and \( h_{C} : \text{Dom}_{C} \rightarrow \mathcal{U}. \) At each iteration \( k, \) the controller takes as input the measurement sample \( w_{k} \in [\mathbb{X}]_{\mu_{\mathbb{X}}} \) and returns as output the control sample \( v_{k} = h_{C}(\xi_{k}, w_{k}) \in \mathcal{U}, \) which is synthesized by a computing unit that may be employed to execute several tasks. Note that, when \( \Xi \) is a singleton set, \( C \) becomes static. The policy by which a computation is executed before or after another computation depends on the scheduling protocol adopted. We assume that:
(A.7) The computation time $\Delta_k^{\text{ctrl}}$ for the symbolic controller to return its output value $v_k$ is bounded, i.e. $\Delta_k^{\text{ctrl}} \in [\Delta_{\text{min}}^{\text{ctrl}}, \Delta_{\text{max}}^{\text{ctrl}}]$, for all $k \in \mathbb{N}$, for some $\Delta_{\text{min}}^{\text{ctrl}}$, $\Delta_{\text{max}}^{\text{ctrl}} \in \mathbb{R}^+$. The control sample $v_k$ is encoded into a digital signal of length $\lceil \log_2 |U| \rceil$, and some overhead information is added to take into account the communication protocol, the packet headers, source and channel coding as well as data compression and encryption. The resulting message is denoted by $\tilde{v}_k$. We assume a fixed average relative overhead $N_{\text{cp}}^+$ on each data bit, which may also be negative due to possible data compression. The following Assumptions (A.8) to (A.11), describing the non-idealities in the controller-to-plant branch of the network, correspond exactly to Assumptions (A.2) to (A.5), previously given for the plant-to-controller branch:

(A.8) $N_{\text{cp}}^+$ bits are added per each bit of $v_k$, i.e. the number of bits of $\tilde{v}_k$ is $(1+N_{\text{cp}}^+)/\lceil \log_2 |U| \rceil$.

(A.9) The sequence $\{\Delta_k^{\text{req,pc}}\}_{k \in \mathbb{N}}$ of network waiting times in the controller-to-plant branch of the feedback loop is bounded, i.e. $\Delta_k^{\text{req,pc}} \in [\Delta_{\text{min}}^{\text{req,pc}}, \Delta_{\text{max}}^{\text{req,pc}}]$, for all $k \in \mathbb{N}$.

At time $t_k^{\text{cp}} := M_k \tau + \Delta_k^{\text{req,pc}} + \Delta_k^{\text{B,cp}} + \Delta_k^{\text{net,cp}} + \Delta_k^{\text{ctrl}} + \Delta_k^{\text{req,cp}}$, the message $\tilde{v}_k$ is sent.

(A.10) At iteration $k$, a delay $\Delta_k^{\text{B,cp}} \in [\Delta_{\text{min}}^{\text{B,cp}}, \Delta_{\text{max}}^{\text{B,cp}}]$ due to the limited bandwidth is introduced in the controller-to-plant branch of the feedback loop, with $\Delta_{\text{min}}^{\text{B,cp}} = (1+N_{\text{cp}}^+)/\lceil \log_2 |U| \rceil/B_{\text{max}}$ and $\Delta_{\text{max}}^{\text{B,cp}} = (1+N_{\text{cp}}^+)/\lceil \log_2 |U| \rceil/B_{\text{min}}$.

(A.11) The sequence $\{\Delta_k^{\text{net,cp}}\}_{k \in \mathbb{N}}$ of network communication delays in the controller-to-plant branch of the feedback loop is bounded, i.e. $\Delta_k^{\text{net,cp}} \in [\Delta_{\text{min}}^{\text{net,pc}}, \Delta_{\text{max}}^{\text{net,pc}}]$, for all $k \in \mathbb{N}$.

The resulting total delay induced by network and computing unit at iteration $k$ is $\Delta_k := \Delta_k^{\text{req,pc}} + \Delta_k^{\text{B,cp}} + \Delta_k^{\text{net,pc}} + \Delta_k^{\text{ctrl}} + \Delta_k^{\text{req,cp}} + \Delta_k^{\text{B,cp}} + \Delta_k^{\text{net,cp}}$. In the absence of packet dropouts, one has $\Delta_k \in [\Delta_{\text{min}}, \Delta_{\text{max}}]$, where $\Delta_{\text{min}}$, $\Delta_{\text{max}} \in \mathbb{R}^+$ are the minimum and maximum delays computed according to the previous assumptions (excluding (A.6)), as $\Delta_{\text{min}} := \Delta_{\text{min}}^{\text{B,cp}} + \Delta_{\text{min}}^{\text{ctrl}} + 2\Delta_{\text{min}}^{\text{net}} + 2\Delta_{\text{min}}$, and $\Delta_{\text{max}} := \Delta_{\text{max}}^{\text{B,cp}} + \Delta_{\text{max}}^{\text{ctrl}} + 2\Delta_{\text{max}}^{\text{net}} + 2\Delta_{\text{max}}$. We can finally define $N_k := \lceil \frac{\Delta_k}{\tau} \rceil$ as the discrete delay induced by iteration $k$, expressed in terms of number of sampling intervals of duration $\tau$.

**ZoH.** Unless message $\tilde{v}_k$ is lost, it is decoded into the control input $v_k$ and reaches the Zero-order-Holder (ZoH) at time $M_k \tau + \Delta_k$. From the definitions of $M_k$ and $N_k$, we get $M_{k+1} = M_k + N_k$. Note that, since we assumed finite bandwidth $B_{\text{max}} \in \mathbb{R}^+$, one has $N_k \geq 1$ for all $k$. The ZoH is updated to the new value $v_k$ at time $M_{k+1} \tau$. The ZoH input sequence is indicated as $\{\tilde{u}_s\}_{s \in \mathbb{N}_0}$ and is so defined by $\tilde{u}_s = v_k$ for $M_{k+1} \leq s < M_{k+2}$, meaning that the value $v_k$ is held exactly for one iteration. The ZoH is placed on the left-hand side of the plant $P$ in Fig. 1. We assume that:
The ZoH is synchronized with the plant and updates its output value at times that are integer multiples of \( \tau \), i.e., \( u(s\tau + t) = u(s\tau) = \bar{u}_s \), for \( t \in [0, \tau] \) and \( s \in \mathbb{N}_0 \), where \( s \) is the index identifying the sampling interval (starting from 0).

The above synchronization assumption is not restrictive since the sub-system ZoH is physically connected to the plant. The ZoH holds a sample until a new one shows up. At time \( t = 0 \) a reference control input \( \bar{u}_0 \in U \) is held by ZoH. So far we have not considered packet dropouts. Under Assumption (A.6) and following the so-called emulation approach, reformulating packet dropouts in terms of additional delays, see e.g. [2], it is readily seen that iteration \( k \) introduces a time-varying delay \( \Delta_k \in [\Delta_{\min}, \Delta_{\max}] \), with \( \Delta_{\min} = \Delta_{\min} \) and \( \Delta_{\max} = (1 + N_{pd})\Delta_{\max} \), where \( N_{pd} \) is the maximum number of subsequent packet dropouts. From the previous assumptions, we conclude that iteration \( k \) introduces a discrete delay of \( N_k \in [N_{\min}; N_{\max}] \) sampling intervals, where the bounds are given by:

\[
N_{\min} = \left\lceil \frac{\Delta_{\min}}{\tau} \right\rceil, \quad N_{\max} = \left\lceil \frac{\Delta_{\max}}{\tau} \right\rceil.
\]  

The semantics of the NCS described above is formally specified by the following equations:

\[
\Sigma : \begin{cases}
\text{Iteration delay:} & N_k = \left\lceil \frac{\Delta_k}{\tau} \right\rceil, k \in \mathbb{N}, \\
\text{Sampling/holding time sequence:} & \begin{cases} M_{k+1} = M_k + N_k, k \in \mathbb{N}, \\ M_1 = 0, \end{cases} \\
\text{Input sequence:} & \begin{cases} \bar{u}_s = \begin{cases} v_{k-1}, & s \geq N_1 \wedge s \in [M_k; M_{k+1}[ , \\ \bar{u}_0, & \text{otherwise}, \end{cases} \\ u(t) = \sum_{s=0}^{\infty} \bar{u}_s 1_{[s\tau,(s+1)\tau]}(t), t \in \mathbb{R}_0^+ , \\ u(0) = \bar{u}_0, \end{cases} \\
\text{ZoH:} & \begin{cases} \bar{u}_0, \\ \bar{u}_s, \\ \bar{u}_s = \begin{cases} v_{k-1}, & s \geq N_1 \wedge s \in [M_k; M_{k+1}[ , \\ \bar{u}_0, & \text{otherwise}, \end{cases} \\ u(t) = \sum_{s=0}^{\infty} \bar{u}_s 1_{[s\tau,(s+1)\tau]}(t), t \in \mathbb{R}_0^+ , \\ u(0) = \bar{u}_0, \end{cases} \\
\text{Plant:} & \begin{cases} \hat{x}(t) = f(x(t), u(t)), t \in \mathbb{R}_0^+ , \\ x(0) = x_0, \end{cases} \\
\text{Sensor:} & \begin{cases} \hat{y}_s = x(s\tau, x_0, u), s \in \mathbb{N}_0, \end{cases} \\
\text{Quantizer:} & y_s = [\hat{y}_s]_{\mu X}, s \in \mathbb{N}_0, \\
\text{Switch:} & w_k = y_s = M_k, k \in \mathbb{N}, \\
\text{Controller:} & \begin{cases} \xi_{k+1} = f_C(\xi_k, w_k), k \in \mathbb{N}, \\ \xi_k = f_C(\xi_k, w_k), \end{cases} \\
\end{cases}
\]

Due to possible different realizations of non-idealities, the model of NCS considered is non-deterministic. In the sequel we refer to the above NCS as \( \Sigma \).

Note that the definition of NCS given in this section allows taking into account different scheduling protocols and communication constraints: any protocol or set of protocols satisfying Assumptions (A.2—A.5), (A.6) and (A.8—A.11) can be used. For example, communication protocols designed for safety-critical control systems, such as Controller Area Network (CAN) [47] and Time Triggered Protocol (TTP) [48] used in vehicular and industrial applications, satisfy the assumptions above.
4 Symbolic Models for NCS

In this section we propose symbolic models that approximate NCS with arbitrarily good accuracy. The approximation scheme employed is based on the notions of alternating approximate simulation and bisimulation that are formally recalled in the Appendix. In Subsection 4.1, we provide a representation of NCS in terms of systems; this first step is instrumental in deriving symbolic models. In Subsection 4.2, we propose symbolic models that approximate NCS with plant admitting incremental forward complete Lyapunov functions, in the sense of alternating approximate simulation; finally, in Subsection 4.3 we show that the proposed symbolic models approximate the NCS in the sense of alternating approximate bisimulation when the plant enjoys the property of incremental stability.

4.1 NCS as systems

NCS are characterized by heterogeneous dynamics; while the plant is described by a differential equation, the controller can be easily represented as a finite state automaton. In order to deal with this heterogeneity, we use the notion of systems as a unified mathematical framework to describe control systems as well as symbolic controllers.

Definition 1 A system is a sextuple \( S = (X, X_0, U, \rho, Y, H) \) consisting of a set of states \( X \), a set of initial states \( X_0 \subseteq X \), a set of inputs \( U \), a transition relation \( \rho \subseteq X \times U \times X \), a set of outputs \( Y \) and an output function \( H : X \to Y \). A transition \( (x, u, x') \in \rho \) of \( S \) is denoted by \( x u \rho x' \). For such a transition, state \( x' \) is called a \( u \)-successor or simply a successor of state \( x \). We denote by \( \text{Post}_u(x) \) the set of \( u \)-successors of a state \( x \) and by \( U(x) \) the set of inputs \( u \in U \) for which \( \text{Post}_u(x) \) is nonempty.

System \( S \) is said to be symbolic (or finite), 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 \( x u \rho x' \) for some \( u \in U \); non-blocking, if \( U(x) \neq \emptyset \) for any \( x \in X \). The evolution of systems is captured by the notions of state and output runs. A state run of \( S \) is a (possibly infinite) sequence \( \{x_i\}_{i \in \mathbb{N}_0} \) such that for any \( i \in \mathbb{N}_0 \) there exists \( u_i \in U \) for which \( x_i u_i \rho x_{i+1} \). An output run is a (possibly infinite) sequence \( \{y_i\}_{i \in \mathbb{N}_0} \) such that there exists a state run \( \{x_i\}_{i \in \mathbb{N}_0} \) with \( y_i = H(x_i) \) for any \( i \in \mathbb{N}_0 \).

In order to give a representation of NCS in terms of systems, we first need to provide an equivalent formulation of NCS. We start by defining a sequence of discrete time-varying delays \( \{R_s\}_{s \in \mathbb{N}_0} \), where \( R_s = N_k \) for all \( s \in \mathbb{N}_0 \) satisfying \( M_k \leq s < M_{k+1} \). This sequence takes into account all delays introduced by the computing unit and the communication channel in the NCS \( \Sigma \). Given the NCS \( \Sigma \), define the system \( \Sigma_d \), which includes a single delay block taking into account all the delays in the NCS \( \Sigma \), in particular the delay \( \Delta_{k}^{\text{net,pc}} \) (before the symbolic
controller block) and the delay $\Delta^\text{net,cp}_k$ (after the symbolic controller block) in Fig. 1. System $\Sigma_d$ is depicted in Fig. 2 and its semantics is formally specified by the following equations:

$$\Sigma_d : \begin{cases} \text{Iteration delay:} & N_k \in [N_{\min}; N_{\max}], k \in \mathbb{N}, \\
\text{Sampling/holding time sequence:} & M_{k+1} = M_k + N_k, k \in \mathbb{N}, \\
& M_1 = 0, \\
\text{Switch:} & \bar{u}_s = v_k, s \in [M_k; M_k+1[, \\
\text{Discrete delay block:} & R_s = N_k, s \in [M_k; M_k+1[, \\
\text{Delayed input:} & \tilde{u}_s = \begin{cases} \bar{u}_s & s \geq N_1, \\
& \bar{u}_0 & \text{otherwise}, \end{cases} \\
\text{Sampled-data control system:} & P_d : \begin{cases} z_{s+1} = f(z_s, \tilde{u}_s), & s \in \mathbb{N}_0, \\
& \tilde{y}_s = z_s. \end{cases} \end{cases}$$

In equations (5), we abstracted the interconnection of blocks ZoH, Plant and Sensor into a nonlinear sampled-data control system $P_d$ which is the time discretization of the plant $P$ with sampling time $\tau$, namely $z_{s+1} = f(z_s, \tilde{u}_s) := x(\tau, z_s, \tilde{u}_s)$ for all $s \in \mathbb{N}_0$. A sequence $\{z_s\}_{s \in \mathbb{N}_0}$ is called a trajectory of the sampled-data control system $P_d$ if it satisfies the above equation for some $\tilde{u}_s$, for all $s \in \mathbb{N}_0$. Note that, since the symbolic controller $C$ in (2) is event-driven and not time-varying, and the discrete delay block in (6) introduces a cumulative delay equal to the iteration delay $N_k$ in $\Sigma$, the sequence of inputs $\{\hat{u}_s\}_{s \in \mathbb{N}_0}$ results to be the same in (2) and (6). As a consequence, for any initial condition and controller given, the corresponding sequences of states measured at the sensors of systems $\Sigma$ and $\Sigma_d$ coincide. We now have all the ingredients to provide a system representation of the control system $\Sigma_d$ in (6). To this purpose, we preliminarily define:

$$X_e = \bigcup_{N \in [N_{\min}; N_{\max}]} X^N.$$

**Definition 2** Given $\Sigma_d$, define the system $S(\Sigma_d) = (X, X_0, U, \tau, Y, H)$, where $X = (X_0 \times U) \cup \{x \in X \mid \exists \bar{u} \in U \text{ s.t. } x_{i+1} = f(x_i, \bar{u}) \forall i \in [1; N-1]\}$, $X_0 = X_0 \times U, x^1 = (x_1, x_2, ..., x_N, \bar{u}^1) \xrightarrow{\tau} x^2 = (x_1, x_2, ..., x_N, \bar{u}^2)$, if $\bar{u}^2 = u$ and

$$x^2_{i+1} = \begin{cases} \tilde{f}(x_{N_i}, \bar{u}^1), & \text{if } i = 0, \\
\tilde{f}(x_i, \bar{u}^2), & \text{if } i \in [1; N-1], \end{cases}$$

(6)
Figure 2: Illustration of Σ_{d}, which is formally described by the equations in (5).

The sequence \( \{\tilde{y}_s\}_{s \in \mathbb{N}_0} \) includes all output samples of the sampled-data control system \( P_d \). At each iteration \( k \), the quantized output \( w_k = y_s = [\tilde{y}_s]_{\mu X} \) for \( s = M_k \) reaches the controller and a control input value \( v_k \) is computed. Block Delay takes into account the total delay \( N_k \) of the NCS loop at iteration \( k \), after which the control input \( v_k \) reaches \( P_d \).

\[
\begin{align*}
Y_\tau &= X_0 \cup X_e \\
H_\tau (x_1, x_2, ..., x_N, \bar{u}) &= (x_1, x_2, ..., x_N), \text{ for all } (x_1, x_2, ..., x_N, \bar{u}) \in X_\tau.
\end{align*}
\]

Note that \( S(\Sigma_d) \) is non-deterministic because, depending on the values of \( N_2 \) in the transition relation, more than one \( u \)-successor of \( x^1 \) may exist. System \( S(\Sigma_d) \) can be regarded as a metric system with the metric \( d_{Y_\tau} \) on \( Y_\tau \) naturally induced by the metrics \( d_{X}(x_1, x_2) = \|x_1 - x_2\| \) on \( X \), as follows. Given any \( x^i = (x^i_1, x^i_2, ..., x^i_{N_i}, \bar{u}^i), i = 1, 2 \), we set \( d_{Y_\tau}(x^1, x^2) = \max_{i \in [1:N]} \|x^1_i - x^2_i\| \) if \( N_1 = N_2 = N \), and \( d_{Y_\tau}(x^1, x^2) = +\infty \), otherwise. Since the state vectors of \( S(\Sigma_d) \) are built from the trajectories of \( P_d \) in \( \Sigma_d \), it is readily seen that:

**Theorem 1** For any trajectory \( \{z_s\}_{s \in \mathbb{N}_0} \) of the sampled-data control system \( P_d \) in \( \Sigma_d \), there exists a state run

\[
\begin{align*}
\left( x_0, \bar{u}_0 \right) &\xrightarrow{u_1} \left( \bar{x}^1_1, \bar{u}_1 \right) \xrightarrow{u_2} \left( \bar{x}^2_1, \bar{u}_2 \right) \xrightarrow{u_3} \cdots \\
\end{align*}
\]

of \( S(\Sigma_d) \) such that:

\[
\{x_0, x^1_1, ..., x^1_{N_1}, x^2_1, ..., x^2_{N_2}, ... \} = \{z_s\}_{s \in \mathbb{N}_0}. \tag{7}
\]

Conversely, for any state run \( \{\bar{x}^i_s\}_{s \in \mathbb{N}_0} \) of \( S(\Sigma_d) \), there exists a trajectory \( \{z_s\}_{s \in \mathbb{N}_0} \) of the sampled-data control system \( P_d \) in \( \Sigma_d \) such that \( \tag{8} \) holds.
Although system $S(\Sigma_d)$ contains all the information of the NCS available at the sensor, it is not a finite model. Hence, in the following subsections, we illustrate the construction of finite systems approximating $S(\Sigma_d)$.

### 4.2 Symbolic models for possibly unstable NCS

In this section we propose symbolic models that approximate possibly unstable NCS in the sense of alternating approximate simulation, whose definition is formally recalled in the Appendix. Our results rely on the assumption of existence of an incremental forward complete (δ-FC) Lyapunov function for the plant control system of the NCS. More formally:

**Definition 3 [37]** A smooth function $V : \mathbf{X} \times \mathbf{X} \to \mathbb{R}_0^+$, is a δ-FC Lyapunov function for the plant control system of the NCS if there exist a real $\lambda \in \mathbb{R}$ and $\mathcal{K}_\infty$ functions $\alpha$ and $\pi$ such that, for any $x_1, x_2 \in \mathbf{X}$ and any $u \in \mathbf{U}$, the following conditions hold:

1. $\alpha(\|x_1 - x_2\|) \leq V(x_1, x_2) \leq \pi(\|x_1 - x_2\|),$
2. $\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).

In [37] it was shown that existence of δ-FC Lyapunov functions for a nonlinear control system is a sufficient condition for the control system to enjoy the so-called incremental forward completeness property. This notion requires that the distance between two arbitrary trajectories of a control system are 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 interested reader can refer to [37] for further details on this notion. In the following, we suppose the existence of a δ-FC Lyapunov function $V$ for the control system $P$ in the NCS $\Sigma$. Moreover, let $\gamma$ be a $\mathcal{K}_\infty$ function$^1$ such that $V(x, x') - V(x, x'') \leq \gamma(\|x' - x''\|)$, for every $x, x', x'' \in \mathbf{X}$. We assume that $V$ is symmetric, i.e. $V(x_1, x_2) = V(x_2, x_1)$ for all $x_1, x_2 \in \mathbf{X}$. This assumption can be given without loss of generality because for any δ-FC Lyapunov function $V : \mathbf{X} \times \mathbf{X} \to \mathbb{R}_0^+$, function $\bar{V} : \mathbf{X} \times \mathbf{X} \to \mathbb{R}_0^+$ defined by $\bar{V}(x_1, x_2) = V(x_1, x_2) + V(x_2, x_1)$, for all $x_1, x_2 \in \mathbf{X}$, is a δ-FC Lyapunov function and also symmetric.

We are now ready to introduce symbolic models approximating NCS. Given a design parameter $\eta \in \mathbb{R}^+$, define the system $S_*(\tilde{\Sigma_d}) := (X_*, X_0, U, \rightarrow_*, Y_*, H_*)$, where $X_* = ([X_0]_{\mu_X} \times U) \cup \{x_1^1, x_2^1, ..., x_N^1, \bar{u}_* \} \in [X_*]_{\mu_X} \times U : \exists \bar{u}_* \in U$ s.t. $V(\bar{f}(x_1^1, \bar{u}_*), x_2^1) \leq \gamma(\|w - z\|) = \left(\sup_{x, y \in \mathbf{X}} \|\partial\bar{V}/\partial x(y, z)\right) \|w - z\|$. 

1. Since $V$ is smooth and $\bar{X}$ is bounded, one can always choose $\gamma(\|w - z\|) = \left(\sup_{x, y \in \mathbf{X}} \|\partial\bar{V}/\partial x(y, z)\right) \|w - z\|$. 

**Proof 1** The proof of the above result follows directly from equations (5), defining $P_d$ and $\Sigma_d$, and from Definition 2 of $S(\Sigma_d)$. 

Proof 1: The proof of the above result follows directly from equations (5), defining $P_d$ and $\Sigma_d$, and from Definition 2 of $S(\Sigma_d)$.
\[ e^{\lambda \tau} \Delta (\eta) + \gamma (\mu_X) \forall i \in [1; N - 1]; \quad X_0, s = [X_0]_{\mu_X} \times U, \quad x^1 = (x_1^1, x_2^1, ..., x_{N_1}^1, \bar{u}_s^1) \xrightarrow{\mu_s} x^2 = (x_1^2, x_2^2, ..., x_{N_2}^2, \bar{u}_s^2), \text{if } \bar{u}_s^2 = u_s \text{ and} \]

\[
\begin{cases} 
V(\bar{f}(x_{N_i}^1, \bar{u}_s^1), x_{i+1}^2) \leq e^{\lambda \tau} \Delta (\eta) + \gamma (\mu_X), \\
V(\bar{f}(x_{i+1}^2, \bar{u}_s^1), x_{i+2}^2) \leq e^{\lambda \tau} \Delta (\eta) + \gamma (\mu_X), 
\end{cases} \quad i \in [1; N_2 - 1];
\]

\[ Y_s = Y_r, \text{and } H_s (x_1^1, x_2^2, ..., x_N^N, \bar{u}_s) = (x_1^1, x_2^2, ..., x_N^N), \text{for all } (x_1^1, x_2^2, ..., x_N^N, \bar{u}_s) \in X_s. \]

**Remark 1** The size of the set of states \( X_s \) scales exponentially with the bound \( N_{\max} \) of the time delay and, when \( N_{\max} \) is large, this can be problematic from the space complexity point of view. The motivation in the present formulation of \( X_s \) is that it makes the formal comparison between \( S_s(\Sigma_d) \) and \( S(\Sigma_d) \) easier, as we shall show in the sequel. However, for computational purposes, it is possible to give a more succinct representation of \( X_s \) by mapping any state \( x_s = (x_1^1, x_2^2, ..., x_N^N, \bar{u}_s) \) into \( (x_1^1, x_N^N, \bar{u}_s) \), where the intermediate components of the aggregate vector \( x_s \) are not stored, in order to save memory; when \( N_{\max} \) is large, this representation of states drastically reduces the space complexity, if compared with the formulation of \( X_s \) in \( S_s(\Sigma_d) \).

Since the set \( X \) is bounded, the set \([X]_{\mu_X}\) is finite, from which system \( S_s(\Sigma_d) \) is symbolic. Furthermore, it is metric when we regard the set \( Y_s \) as being equipped with the metric \( d_Y \). We can now present the following result that identifies in the existence of incremental forward complete Lyapunov functions a sufficient condition for the symbolic model \( S_s(\Sigma_d) \) to approximate \( S(\Sigma_d) \) in the sense of alternating approximate simulation\(^2\) with (any desired) accuracy \( \varepsilon \), i.e. \( S_s(\Sigma_d) \leq_{\text{alt}} S(\Sigma_d) \).

**Theorem 2** Consider \( \Sigma_d \) and suppose that there exists a \( \delta \)-FC Lyapunov function \( V \) for the control system \( P \) in the NCS \( \Sigma \). 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:

\[ \mu_X < \min\{\mu_X, \eta^{-1}(\alpha(\varepsilon))\} \leq \varepsilon, \]

we have \( S_s(\Sigma_d) \leq_{\text{alt}} S(\Sigma_d) \).

**Proof 2** Consider the relation \( R \subseteq X_s \times X_s \) defined by \((x^*, x) \in R \) if and only if \( x^* = (x_1^1, x_2^1, ..., x_N^N, \bar{u}_s), x = (x_1, x_2, ..., x_N, \bar{u}), \) for some \( N, V(x_i^1, x_i) \leq \alpha(\varepsilon) \) for all \( i \in [1; N] \), and \( \bar{u}_s = \bar{u} \). We first prove condition (i) of Definition\(^3\) in the Appendix. For any \( x^* = (x_s^*, \bar{u}_s) \in X_{0,s} \), choose \( x = (x_o, \bar{u}) \in X_{0,\tau} \), with \( x_o = x_s^* \) and \( \bar{u} = \bar{u}_s \), which implies that \( ||x_s^* - x_o|| \leq \mu_X \). Hence, from condition (i) in Definition\(^3\) and the inequality in \((10)\) one gets:

\[ V(x_s^*, x_0) \leq \alpha(\mu_X) \leq \alpha(\eta^{-1}(\alpha(\varepsilon))) = \alpha(\varepsilon), \]

\(^2\)For ease of notation in the sequel we refer to an alternating approximate simulation with accuracy \( \varepsilon \) by \( \text{AltA} \) simulation.
which concludes the proof of condition (i). We now consider condition (ii) of Definition 3. For any $(x^*, x) \in \mathcal{R}$, from the definition of the metric $d_{\gamma^1}$, the definition of $\mathcal{R}$ and condition (i) in Definition 3, one can write $d_{\gamma^1}(x^*, x) = \max_i \| x_i^* - x_i \| \leq \max_i \frac{1}{\alpha}(V(x_i^*, x_i)) \leq \frac{1}{\alpha}(\alpha(\varepsilon)) = \varepsilon$. We now show that condition (iii') in Definition 3 holds. Consider any $(x^*, x) \in \mathcal{R}$, with $x^* = (x_1^*, x_2^*, ..., x_N^*, u_*)$ and $x = (x_1, x_2, ..., x_N, \bar{u})$; then pick any $u = u_* \in \mathcal{U}$ and consider any transition $\bar{x} \xrightarrow{u} x$, with $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_N, u)$, for some $\bar{N}$. Pick $\bar{x}^* = (\bar{x}_1^*, \bar{x}_2^*, ..., \bar{x}_N^*, u_*)$ defined by $\bar{x}_i^* = [\bar{x}_i]_{\mu_X}$ for all $i \in [1; \bar{N}]$. We now prove that $x^* \xrightarrow{\bar{u}} \bar{x}^*$ is a transition of $S_*(\Sigma_d)$. First, from condition (i) in Definition 3, the definition of $\bar{x}^*$ and the first inequality in (12), one can write:

$$V(\bar{x}_i^*, \bar{x}_i) \leq \gamma(\mu_X) \leq \alpha^{-1}(\alpha(\varepsilon)) = \alpha(\varepsilon)$$

for all $i \in [1; \bar{N}]$. By using condition (ii) in Definition 3, one has:

$$\frac{\partial V}{\partial x_i} f(x_i^*, \bar{u}_i) + \frac{\partial V}{\partial x_N} f(x_N, \bar{u}) \leq \lambda V(x_i^*, x_N).$$

By the definitions of $\gamma$, $\mathcal{R}$ and $S(\Sigma_d)$, and by integrating the previous inequality, the following holds:

$$V(\bar{f}(x_i^*, \bar{u}_i), \bar{x}_i) \leq V(\bar{f}(x_i^*, \bar{u}_i), \bar{x}_i) + \gamma(||\bar{x}_i - \bar{x}_i^*||) \leq e^{\lambda \eta} V(x_i^*, x_N) + \gamma(||\bar{x}_i - \bar{x}_i^*||) \leq e^{\lambda \eta} \alpha(\varepsilon) + \gamma(\mu_X) = e^{\lambda \eta} \alpha(\varepsilon) + \gamma(\mu_X).$$

where condition $\varepsilon = \eta$ in (12) has been used in the last step. By similar computations, it is possible to prove that the inequality in (12) implies:

$$V(\bar{f}(x_i^*, \bar{u}_i), \bar{x}_i) \leq e^{\lambda \eta} \alpha(\varepsilon) + \gamma(\mu_X), \ i \in [1; \bar{N} - 1].$$

Hence, from the inequalities in (13)–(14) and from the definition of the transition relation of $S_*(\Sigma_d)$ in (3), the transition $x^* \xrightarrow{\bar{u}} \bar{x}^*$ is in $S_*(\Sigma_d)$, implying with (12) that $(\bar{x}^*, \bar{x}) \in \mathcal{R}$, which concludes the proof.

Remark 2 In some practical case studies, the accuracy $\mu_X$ of the quantizer may not be chosen arbitrarily small as requested in condition (11). If a lower bound $\mu_{X, \text{min}}$ to the accuracy of the quantizer is given, the attainable accuracy $\varepsilon$ in the above result is lower bounded by $\varepsilon_{\text{min}} = \alpha^{-1}(\alpha(\mu_{X, \text{min}}))$.

The result given above is important because it provides symbolic models that approximate possibly unstable nonlinear NCS with arbitrarily good accuracy. However, since the relationship between $S(\Sigma_d)$ and $S_*(\Sigma_d)$ is given in terms of alternating approximate simulation, if a symbolic controller, designed on the basis of $S_*(\Sigma_d)$ for enforcing a given specification, fails to exist, there is no guarantee that a controller, enforcing the same specification, does not exist for the original NCS model. When alternating approximate simulation is replaced by alternating approximate bisimulation, the above drawback is overcome. In the following subsection, we derive sufficient conditions under which alternatingly approximately bisimilar symbolic models can be constructed.
4.3 Symbolic models for incrementally stable NCS

In this section we suppose the existence of a symmetric $\delta$-FC Lyapunov function for the control system $P$, which satisfies the inequality (ii) in Definition 3 for some $\lambda < \epsilon$.

Theorem 3 Consider the NCS $\Sigma$ and suppose that there exists a symmetric $\delta$-FC Lyapunov function for the control system $P$ in the NCS $\Sigma$ satisfying the inequality (ii) in Definition 3 for some $\lambda < \epsilon$. Then for any desired precision $\varepsilon \in \mathbb{R}^+$, any sampling time $\tau \in \mathbb{R}^+$ and any state quantization $\mu_X$ satisfying the following inequality:

$$
\mu_X \leq \min \left\{ \gamma^{-1} \left( (1 - \epsilon^{\lambda \tau}) \mathcal{A}(\varepsilon) \right), \overline{\mathcal{P}}^{-1}(\mathcal{A}(\varepsilon)), \bar{\nu}_X \right\},
$$

systems $\Sigma_*(\overline{\Sigma}_d)$ and $S(\overline{\Sigma}_d)$ are alternatingly approximately bisimilar with accuracy $\varepsilon$.

Proof 3 Consider the relation (already used in the proof of Theorem 2) $\mathcal{R} \subseteq X_* \times X_*$ defined by $(x^*, x) \in \mathcal{R}$ if and only if $x^* = (x_1^*, x_2^*, ..., x_N^*, \bar{u}_*)$, $x = (x_1, x_2, ..., x_N, \bar{u})$, for some $N$, $V(x_i^*, x_i) \leq \mathcal{A}(\varepsilon)$ for all $i \in [1; N]$, and $\bar{u}_* = \bar{u}$. The proof of conditions (i)-(ii) of Definition 3 in the Appendix is the same as the one given in the proof of Theorem 2 since it is not affected by the modifications on the symbolic model $S_*(\overline{\Sigma}_d)$. Next we show that condition (iii) in Definition 3 holds. Consider any $(x^*, x) \in \mathcal{R}$, with $x^* = (x_1^*, x_2^*, ..., x_N^*, \bar{u}_*)$, $x = (x_1, x_2, ..., x_N, \bar{u})$; then pick any $u = u_* \in \mathcal{U}$ and consider any transition $\overline{x} = \overline{x}(x_1, \overline{x}_2, ..., \overline{x}_N, u)$, for some $N$. Pick the transition $\overline{y} = \overline{y}(\overline{x}_1, \overline{x}_2, ..., \overline{x}_N, u)$, for some $N$. Next pick the transition $\overline{y} = \overline{y}(\overline{x}_1, \overline{x}_2, ..., \overline{x}_N, u)$, for some $N$.
$x^* \xrightarrow{u} \bar{x}^*$, with $\bar{x}^* = (\bar{x}_1^*, \bar{x}_2^*, ..., \bar{x}_N^*, u_*)$, and define the state $\bar{x}_1^* := \bar{f}(x_N^*, \bar{u}_*)$.

By using condition (ii) in Definition 5 one gets:

$$\frac{\partial V}{\partial x_N} f(x_N^*, \bar{u}_*) + \frac{\partial V}{\partial x_N} f(x_N, \bar{u}) \leq \lambda V(x_N^*, x_N). \quad (17)$$

By the symmetry property of $V$, the definitions of $\gamma$, $R$, $S(\bar{\Sigma}_d)$ and $S_*(\bar{\Sigma}_d)$, and by integrating the previous inequality, the following holds:

$$V(\bar{x}_1^*, \bar{x}_1) \leq V(\bar{x}_1^*, \bar{x}_1) + \gamma(\|\bar{x}_1 - \bar{x}_1^*\|) \leq e^{\gamma \lambda} V(x_N^*, x_N) + \gamma(\|\bar{x}_1 - \bar{x}_1^*\|) \leq e^{\gamma \lambda} \Omega(\varepsilon) + \gamma(\mu_X) \leq \Omega(\varepsilon), \quad (18)$$

where condition (10) has been used in the last step. By similar computations, it is possible to prove by induction that $V(\bar{x}_i^*, \bar{x}_i) \leq \Omega(\varepsilon)$ implies $V(\bar{x}_{i+1}^*, \bar{x}_{i+1}) \leq \Omega(\varepsilon)$, for any $i \in [1; N - 1]$. Hence the inequality $V(\bar{x}_i^*, \bar{x}_i) \leq \Omega(\varepsilon)$ has been proven for any $i \in [1; \bar{N}]$, implying $(\bar{x}^*, \bar{x}) \in R$, which concludes the proof of condition (iii) of Definition 5. We complete the proof by showing that the conditions (i), (ii) and (iii') of Definition 5 hold for the relation $R^{-1}$. We first prove condition (i) of Definition 5. For any $x = (x_0, \bar{u}) \in X_{0, \tau}$, choose $x^* = (x_0^*, \bar{u}_*) \in X_{0, \tau}$, with $x_0^* = [x_0]_{\delta X}$ and $\bar{u}_* = \bar{u}$, which implies that $\|x_0 - x_0^*\| \leq \mu_X$. Hence the inequality in (17) holds, which concludes the proof of condition (i). The proof of condition (ii) of Definition 5 for the relation $R^{-1}$ is the same as the one for the relation $R$ and is not reported. Next we show that condition (iii') in Definition 5 holds for $R^{-1}$. Consider any $(x, x^*) \in R^{-1}$, with $x = (x_1, x_2, ..., x_N, \bar{u})$, $x^* = (x_1^*, x_2^*, ..., x_N^*, \bar{u}_*)$; then pick any $u = u_* \in U$ and consider any transition $x^* \xrightarrow{u} \bar{x}^*$, with $\bar{x}^* = (\bar{x}_1^*, \bar{x}_2^*, ..., \bar{x}_N^*, \bar{u}_*)$, for some $\bar{N}$. Now pick the transition $x \xrightarrow{u} \bar{x}$, with $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_N, u)$, and define the state $\bar{x}_1^* := \bar{f}(x_N^*, \bar{u}_*)$. After that, it is possible to rewrite exactly the same steps as in the proof of condition (iii') for $R$, in particular (18), implying that $V(\bar{x}_i^*, \bar{x}_i) \leq \Omega(\varepsilon)$ for any $i \in [1; \bar{N}]$; as a consequence $(\bar{x}^*, \bar{x}) \in R$, hence one gets $(\bar{x}, \bar{x}^*) \in R^{-1}$, concluding the proof.

The above theorem provides stronger results than Theorem 2 (A ϵ A bisimulation vs. A ϵ A simulation) at the expense of stronger assumptions (δ-GAS vs. existence of δ-FC Lyapunov functions).

**Remark 3** By Proposition 3.4 of [12], for control systems with compact state space, incremental global asymptotic stability (δ-GAS) and global asymptotic stability (GAS) are equivalent notions. Moreover in [12] it is shown that the existence of a δ-GAS Lyapunov function is equivalent to the GAS property. For this reason, the assumption of existence of a δ-GAS Lyapunov function in Theorem 5 can be replaced by the GAS property. However, since at present there are no constructive results available in the literature to derive a δ-GAS Lyapunov function for a GAS control system (as requested in the statement of Theorem 5 and in the definition of the symbolic model $S_*(\bar{\Sigma}_d)$), when the assumptions of Theorem 5 are replaced by the GAS property, the result obtained is only of existential nature.
5 NCS Symbolic Control Design

In this section, we address NCS symbolic control design with specifications expressed in terms of non-deterministic transition systems. We consider a control design problem where the NCS Σ has to satisfy a given specification Q while being robust with respect to the non-idealities of the communication network. Our specification Q is expressed in terms of a collection of transitions $Q \subseteq X_Q \times X_Q$, where $X_Q$ is a finite subset of $X$, and a set of initial states $X_Q^0 \subseteq X_Q$. For the forthcoming developments it is convenient to reformulate the specification Q in terms of the following system:

$$S(Q) = (X_Q, X_Q^0, U_q, \overset{q}{\longrightarrow}, Y_q, H_q),$$  \hspace{1cm} (19)

where $X_q = X_Q^0 \cup \{x = (x_1, x_2, ..., x_N) \in X_Q^N, N \in [N_{\min}; N_{\max}]\}$, $x \overset{q}{\rightarrow} x_{i+1}$, $i \in [1; N - 1]$, $U_q = \{u_q\}$, where $u_q$ is a dummy symbol, $x^1 \overset{u_q}{\rightarrow} x^2$, if $x^1 = (x_1^1, x_2^1, ..., x_N^1)$, $x^2 = (x_1^2, x_2^2, ..., x_N^2)$ and $x_N^1 \overset{q}{\rightarrow} x_N^2$, $Y_q = Y_\tau$, and $H_q(x) = x$, for all $x \in X_q$. We can now formally state the symbolic control problem considered.

**Problem 1** Consider the NCS Σ, the specification $S(Q)$ in (19) and a desired precision $\varepsilon \in \mathbb{R}^+$. Find a symbolic controller system $S_C$, a parameter $\theta \in \mathbb{R}^+$ and an AθA simulation relation $R$ from $S_C$ to $S(\Sigma_d)$ such that:

1. the $\theta$-approximate feedback composition of $S(\Sigma_d)$ and $S_C$, denoted $S(\Sigma_d) \times_R^\theta S_C$, is approximately simulated by $S(Q)$ with accuracy $\varepsilon$, i.e. $S(\Sigma_d) \times_R^\theta S_C \preceq_\varepsilon S(Q)$;
2. the system $S(\Sigma_d) \times_R^\theta S_C$ is non-blocking.

The above control design problem is known in the literature as approximate similarity game (see e.g. [21]), where condition (1) requires the state trajectories of the NCS to be close to the state run of the specification $S(Q)$ up to the accuracy $\varepsilon$ irrespective of the particular realization of the network non-idealities, and condition (2) prevents deadlocks in the interaction between the plant and the controller. In Problem 1 we considered a controller in the form of a symbolic system rather than a Mealy machine as in (2). In the end of this section we discuss how to derive a Mealy machine controller $C$ from the controller $S_C$. In order to solve Problem 1 some preliminary definitions and results are needed. Given two systems $S_i = (X_i, X_{0,i}, U_i, \overset{i}{\longrightarrow}, Y_i, H_i)$ ($i = 1, 2$), $S_1$ is a subsystem of $S_2$ if $X_1 \subseteq X_2$, $X_{0,1} \subseteq X_{0,2}$, $U_1 \subseteq U_2$, $\overset{i}{\longrightarrow} \subseteq \overset{2}{\longrightarrow}$, $Y_1 \subseteq Y_2$, and $H_1(x) = H_2(x)$ for any $x \in X_1$. Given two sub-systems $S_i = (X_i, X_{0,i}, U_i, \overset{i}{\longrightarrow}, Y_i, H_i)$ ($i = 1, 2$) of a system $S$, define the union system $S_1 \cup S_2$ as $(X_1 \cup X_2, X_{0,1} \cup X_{0,2}, U_1 \cup U_2, \overset{1}{\longrightarrow} \cup \overset{2}{\longrightarrow}, Y_1 \cup Y_2, H)$, where $H(x) = H_1(x)$

\footnote{The notions of approximate feedback composition and of approximate simulation are formally recalled in the Appendix.}
is \( x \in X_1 \) and \( H(x) = H_2(x) \) otherwise. Note that \( S_1 \sqcup S_2 \) is a sub-system of \( S \). It is easy to see that the union operator enjoys the associative property. We now have all the ingredients to introduce the controller \( S_{C^*} \) that will solve Problem 4.

**Definition 4** The symbolic controller \( S_{C^*} \) is the maximal non-blocking sub-system \( S_C \) of \( S_*(\Sigma_d) \) such that:

1. \( S_C \) is approximately simulated by \( S(Q) \) with accuracy \( \mu_X \), i.e. \( S_C \preceq_{\mu_X} S(Q) \);
2. \( S_C \) is alternatingly 0-simulated by \( S_*(\Sigma_d) \), i.e. \( S_C \preceq_{alt 0} S_*(\Sigma_d) \).

Condition (1) requires that for any state run \( r_c \) of \( S_C \) there exists a state run \( r_q \) in \( S(Q) \) such that \( r_c \) approximates \( r_q \) within the accuracy \( \mu_X \). Condition (2) ensures that the controller enforces the specification irrespective of the time-delay realization induced by the communication network. The following result holds.

**Proposition 1** The symbolic controller \( S_{C^*} \) is the union of all non-blocking sub-systems \( S_C \) of \( S_*(\Sigma_d) \) satisfying conditions (1) and (2) of Definition 4.

The proof of the above result is a direct consequence of the definition of the union operator and of Definition 4; it is therefore omitted. Since \( S(Q) \) and \( S_*(\Sigma_d) \) are symbolic systems, a symbolic (finite) controller \( S_{C^*} \) can be computed in a finite number of steps by adapting standard fixed point characterizations of bisimulation [50, 21]. We are now ready to provide the solution of Problem 4.

**Theorem 4** Consider the NCS \( \Sigma \) and the specification \( S(Q) \). Suppose that there exists a \( \delta \)-FC Lyapunov function \( V \) for the control system \( P \) in the NCS \( \Sigma \). For any desired precision \( \varepsilon \in \mathbb{R}^+ \), choose the parameters \( \theta, \mu_X, \eta \in \mathbb{R}^+ \) such that:

\[
\mu_X + \theta \leq \varepsilon, \tag{20}
\]

\[
\mu_X < \min\{\mu_X, \alpha^{-1}(\Theta(\theta))\} \leq \theta = \eta. \tag{21}
\]

Then a \( A\Theta A \) simulation relation \( \mathcal{R} \) from \( S_{C^*} \) to \( S(\Sigma_d) \) exists which solves Problem 4 with \( S_C = S_{C^*} \).

**Proof 4** By condition (2) in Definition 4 a (non-empty) \( A\Theta A \) simulation relation \( \mathcal{R}_1 \) from \( S_{C^*} \) to \( S_*(\Sigma_d) \) exists. Let \( \mathcal{R}_2 \) be a \( A\Theta A \) simulation relation from \( S_*(\Sigma_d) \) to \( S(\Sigma) \), which exists by the assumption on existence of a \( \delta \)-FC Lyapunov function for the plant \( P \) of the NCS in view of Theorem 3. Define the relation \( \mathcal{R} = \{(x^1, x^3) \in X_{C^*} \times X_* | \exists x^2 \in X_* \text{ s.t. } (x^1, x^2) \in \mathcal{R}_1 \text{ and } (x^2, x^3) \in \mathcal{R}_2\} \), where \( X_{C^*} \) is the set of states of controller \( S_{C^*} \). By Lemma 1 (ii), \( \mathcal{R} \) is a \( A\Theta A \)

---

Here maximality is defined with respect to the preorder induced by the notion of sub-system.
simulation relation from $S_{C^*}$ to $S(\bar{\Sigma}_d)$. We now prove condition (1) of Problem 4. From condition (2) in Definition 4,

$$S_{C^*} \preceq_{\mu_X} S(Q).$$

Furthermore from Theorem 1 the condition in (21) implies that

$$S_x(\bar{\Sigma}_d) \preceq_{\mu_X} S(\bar{\Sigma}_d).$$

Hence, from Lemma 1 (ii) in the Appendix, by combining (22) and (23) one gets $S_{C^*} \preceq_{\mu_X} S(\bar{\Sigma}_d)$, which, by Lemma 1 (iii) implies

$$S(\bar{\Sigma}_d) \times^R S_{C^*} \preceq_{\mu_X} S(\bar{\Sigma}_d),$$

since $R$ is a $A\theta A$ simulation relation from $S_{C^*}$ to $S(\bar{\Sigma}_d)$. By condition (1) in Definition 4,

$$S_{C^*} \preceq_{\mu_X} S(Q).$$

By Lemma 1 (ii) and condition (24) the similarity inclusions in (24) and (25) imply $S(\bar{\Sigma}_d) \times^R S_{C^*} \preceq_{\mu_X} S(Q)$, which concludes the proof of condition (1) of Problem 4. We now show that condition (2) holds. Consider any state $(x, x_e)$ of $S(\bar{\Sigma}_d) \times^R S_{C^*}$. Pick any $u_c \in U_c(x_e)$, which is a non-empty set because $S_{C^*}$ is non-blocking. Since $(x_e, x) \in R$, there exists $u$ such that for any $x \xrightarrow{u_c} x'$ in $S(\bar{\Sigma}_d)$ there exists $x_e \xrightarrow{u} x'_e$ in $S_{C^*}$ with $(x'_e, x') \in R$. Hence, from Definition 4 the transition $(x, x_e) \xrightarrow{u} (x', x'_e)$ is in $S(\bar{\Sigma}_d) \times^R S_{C^*}$, implying that $S(\bar{\Sigma}_d) \times^R S_{C^*}$ is non-blocking, which concludes the proof of condition (ii) in Problem 4.

**Remark 4** Note that the choice of $\theta$ and $\mu_X$ is not unique, provided they satisfy the conditions in Theorem 4. A larger $\theta$ results in a larger $A\theta A$-simulation relation in the $R$ from $S_{C^*}$ to $S(\bar{\Sigma}_d)$ in the controller; as a consequence, states in the plant can be mapped into states of the controller with a higher approximation, resulting in a less precise control action with respect to the choice of a smaller $\theta$. On the other hand, a smaller $\theta$ forces the choice of a smaller quantization $\mu_X$ in the symbolic controller, according to (21), resulting in a higher space complexity.

We conclude this section by deriving a controller $C^*$ in the form of (2). On the basis of the symbolic controller $S_{C^*}$, we first note that the controller $S_{C^*}$ is in general non-deterministic because it is obtained as a sub-system of the non-deterministic symbolic model $S_x(\bar{\Sigma}_d)$. In particular, multiple sequences of control inputs can solve the specification, even starting from the same initial condition. Since $S_{C^*}$ is a sub-system of $S_x(\bar{\Sigma}_d)$, from (9) the transitions of $S_{C^*}$ are in the form $x^1 = (x^1_1, x^1_2, ..., x^1_{N_1}, \bar{a}^1_{x^1}) \xrightarrow{u} x^2 = (x^2_1, x^2_2, ..., x^2_{N_2}, \bar{a}^2_{x^2})$. Starting from $S_{C^*}$, we define the controller $C^*$ in (2) by $\Xi = X_*$ and

$$\begin{cases} h_C(\xi, w) \in U(\xi), \\ f_C(\xi, w) = \text{Post}_{h_C(\xi, w)}(\xi). \end{cases}$$

(26)
for any \((ξ, w) ∈ Dom_C := \{(ξ, w) = ((x_1^*, ..., x_N^*, û), w) ∈ X × [X]_μ : \|x_N^* - w\| ≤ θ\}\), where \(U(ξ)\) and \(Post_{hc(ξ, w)}(ξ)\) are defined as in Definition 11 applied to system \(S_{C^*}\). Note from the first line in (26) that the controller \(S_{C^*}\) derived from a non-deterministic system \(S_{C^*}\) is not uniquely determined, since \(U(ξ)\) may not be a singleton. Moreover, the second line in (26) takes into account that \(x_N^*\) is the state of the aggregate vector \(x^*\) in \(ξ\) which is required to match the output sample \(w\), sent through the plant-to-controller branch of the network and reaching the controller (as illustrated in Section 5).

6 Application to Robot Motion Planning with Remote Control

In this section, we apply the results derived in the previous sections to an example in the context of robot motion planning with remote control. Symbolic techniques for robot motion planning and control have been greatly exploited in the literature, see e.g. [51] and the references therein. However, existing work does not consider the symbolic control of robot motion over non-ideal communication networks. In this section we exploit the remote control of an electric car-like robot, with limited power, sensing, computation and communication capabilities, whose goal is the surveillance of an area. The motion of the robot \(P\) is described by means of the following nonlinear control system:

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{bmatrix} =
\begin{bmatrix}
\frac{u_1 \cos(x_3 + \delta(u_2))}{\cos(\delta(u_2))} \\
\frac{u_1 \cos(x_3 + \delta(u_2))}{\sin(x_3 + \delta(u_2))} \\
\frac{u_1}{b} \tan(u_2)
\end{bmatrix},
\]

(27)

where \(\delta(u_2) = \arctan\left(\frac{a \tan(u_2)}{b}\right)\), \(a = 0.5\) is the distance of the center of mass from the rear axle and \(b = 1.5\) is the wheel base, see Fig. 5 (top left panel). The state quantities are the 2D-coordinates of the center of mass of the vehicle and its heading angle, while the inputs are the velocity of the rear wheel and the steering angle. Note that \(u_1\) is always nonnegative to guarantee that the vehicle does not move backwards. All the quantities are expressed in units of the International System (SI). We suppose that \(x ∈ X = X_0 = [−x_{1,\text{max}}, x_{1,\text{max}}] × [−x_{2,\text{max}}, x_{2,\text{max}}] × [−x_{3,\text{max}}, x_{3,\text{max}}]\) and \(u ∈ U = [0, u_{1,\text{max}}] × [−u_{2,\text{max}}, u_{2,\text{max}}]\), where \(x_{\text{max}} = [x_{1,\text{max}}, x_{2,\text{max}}, x_{3,\text{max}}]' = [50, 50, \pi]'\) and \(u_{\text{max}} = [u_{1,\text{max}}, u_{2,\text{max}}]' = [5, \pi]'\). The model above is known in the literature as single-track vehicle model and is widely used because, in spite of its simplicity, it well captures the major features of interest of the vehicle cornering behavior [32]. The robot \(P\) is remotely connected to a controller, implemented on a shared CPU, by means of a non-ideal communication network. The control loop forms a NCS, as the one in Fig. 1, whose network/computation parameters are \(B_{\text{min}} = 0.1\ \text{kbit/s}, B_{\text{max}} = 1\ \text{kbit/s}, τ = 1\ \text{s}, Δ_{\text{ctrl}}^{\text{min}} = 0.01\ \text{s}, Δ_{\text{ctrl}}^{\text{max}} = 0.1s, Δ_{\text{req}}^{\text{min}} = 0.05s, Δ_{\text{req}}^{\text{max}} = 0.2s, Δ_{\text{net}}^{\text{min}} = 0.1s, Δ_{\text{net}}^{\text{max}} = 0.25s\). The state quantization, assumed to be different (in absolute values) for each compo-
The component of the state, is equal to $x_{i,\text{max}}/100$ for the state $x_i$ ($i = 1, 2, 3$), so that we have 201 quantization values for each state component. We assume the input quantization to be equal to $u_{i,\text{max}}/5$ for the input $u_i$ ($i = 1, 2$) and the network protocols to introduce a relative overhead which is bounded by the 20% of the total number of data bits ($N_{\text{cp}}^+ = N_{\text{pc}}^+ = 0.2$). This implies $|[X]_{\mu_X}| = 201^3$ and $|U| = 66$, hence $\Delta_{\text{min}}^{B_{\text{pc}}} = 0.0275s$, $\Delta_{\text{max}}^{B_{\text{pc}}} = 0.275s$, $\Delta_{\text{min}}^{B_{\text{cp}}} = 0.0073s$, $\Delta_{\text{max}}^{B_{\text{cp}}} = 0.073s$. We assume there may be packet dropouts, with the constraint that two consecutive dropouts are not allowed ($N_{\text{pd}} = 1$). The motion planning problem considered here is described in the following. We require that the robot leaves its support (HOME location) and visits (in the exact order) two buildings, denoted by $B1$ and $B2$, to then reach an outlet where it possibly powers up the battery (CHARGE location). Finally, the vehicle returns HOME. During the whole path, the robot is requested to avoid some obstacles, such as walls and other buildings. We denote the union of the obstacles locations as the UNSAFE location. We now start applying the results in Section 4 regarding the design of a symbolic model for the given NCS. According to the definition of $\Sigma_d$ in Subsection 4.1, the minimum and maximum delays in a single iteration of the network amount to $\Delta_{\text{min}} = 0.34s$ and $\Delta_{\text{max}} = 2.70s$, respectively. From (3), this results in $N_{\text{min}} = 1$, $N_{\text{max}} = 3$. In order to have a uniform quantization in the state space and in the input space, we apply the results to a normalized plant $\hat{P}$, whose state and input are those of $P$, but component-wise normalized with respect to $x_{\text{max}}$ and $u_{\text{max}}$. According to the previous description of the NCS, this results in $\mu_X = 0.005$ and $\mu_U = 0.1$. We assume that the normalized signals are sent through the network and the static blocks implementing the coordinate change from $P$ to $\hat{P}$ and vice versa (omitted in the general scheme) are physically connected to the sensor and to the ZoH, respectively. It is possible to show that the quadratic Lyapunov-like function $V(x, x') = 0.5 \|x - x'\|^2_2$ is $\delta$-FC for control system (27), with $\lambda = \frac{2u_{\text{max}}}{\text{const}(d_{u_{\text{max}}})}$, $\alpha(r) = 0.5r^2$, $\overline{\alpha}(r) = 1.5r^2$ and $\gamma(r) = 6r$; hence Theorem 2 can be applied. Further details are omitted because, as it will be discussed in the sequel, the explicit construction of the symbolic model is not needed to solve the control design problem. In the symbolic control design step, we apply the results illustrated in Section 5 and we consider a finite automaton encoding all the trajectories satisfying the given specification. Although a covering specification can be repeated many times, we consider a single surveillance round, which can be coded into a finite-time specification by means of the following co-safe LTL formula [53]:

$$\phi = \text{HOME} \land (\neg\text{UNSAFE} \cup \text{HOME}) \land (\neg\text{HOME} \cup (B1 \land \diamond(B2 \land \diamond\text{CHARGE})))$$

(28)

where $\neg$ and $\land$ are the logical connectives of negation (not) and conjunction (and), while $\cup$ and $\diamond$ are the temporal operators of until and eventually, respectively. The formula in (28) is the logical conjunction of two formulas, where the first one requires that the vehicle goes back to the location HOME before visiting the locations $B1$, $B2$ and CHARGE, in the exact order. We assume
that the robot starts from HOME.

For a precision $\varepsilon = 0.025$, starting from a specification $Q$ encoding point-to-point trajectories fulfilling the formula in (28), for the choice of the parameters $\theta = \eta = 0.0125$, Theorem 4 holds and the controller $S_C^*$ in Definition 4 solves the control problem. Estimates of the space complexity in constructing $S_C^*$ indicate $4 \cdot 10^{13}$ 32-bit integers. Because of the large computational complexity in building the controller and the specification automaton, we do not construct the whole models but solve the motion control problem by means of the procedure illustrated in [45] for the on-the-fly NCS control design, generalizing the integrated control design technique developed in [54] for nonlinear systems to the case of non-determinism and unstable plants. The total memory occupation and time required to construct $S_C^*$ are respectively 3742 32-bit integers and 2833 s. The computation has been performed on the Matlab suite through an Apple MacBook Pro with 2.5GHz Intel Core i5 CPU and 16 GB RAM. In Fig. 3 (bottom panel), we show a sample path of the NCS (blue solid line), for a particular realization of the network uncertainties, compared to the trajectory of the system controlled through an ideal network (black dashed line). As described before, the robot visits the regions $B_1$, $B_2$ and CHARGE (in yellow), while avoiding the obstacles (in red), to finally go back HOME (in green).

Each time delay $N_k$ is sampled from a discrete uniform random distribution over $[N_{\min}; N_{\max}]$. As a result, the NCS used just 59 control samples, in spite of the 94 control samples (one at each $\tau$) used in the ideal case. The plot of the NCS input function and of the realization of time delays are in Fig. 3 (top right panel). Note that, although the behavior of the NCS is not as regular as in the ideal case, the specifications are indeed met.

7 Conclusions

In this paper we proposed a symbolic approach to the control design of nonlinear NCS, where the most important non-idealities in the communication channel are taken into account. Under the assumption of existence of incremental forward complete Lyapunov functions, we derived symbolic models that approximate NCS in the sense of alternating approximate simulation. Under the assumption of incremental global asymptotic stability, alternatingly approximately bisimilar symbolic models are constructed. NCS symbolic control design, where specifications are expressed in terms of transition systems, was then solved and applied to an example in the context of robot motion planning. The results presented in this paper represent a first step in solving complex control problems where non-idealities in communication infrastructures and computing units are taken into account. However, some simplifying assumptions have to be dropped to make the proposed results applicable to more realistic industrial cases and more complex control objectives. In particular, multiple control and measurement packets (with out-of-order packet management) within each network iteration can be considered, thereby improving the control performance at the expense of additional formal complexity. Moreover, specifications expressed in terms of
Figure 3: Overhead view of the robot dynamics (top left panel). Control input and realization of the network delays (top right panel) in the NCS Σ. Space trajectory of the vehicle (bottom panel).
Linear Temporal Logic formulae can be taken into account.

Acknowledgements

The authors are grateful to Pierdomenico Pepe for fruitful discussions on the topic of this article.

References

[1] R. Murray, K. Astrom, S. Boyd, R. Brockett, and G. Stein, “Control in an information rich world,” *IEEE Control Systems Magazine*, vol. 23, no. 2, pp. 20–33, April 2003.

[2] 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.

[3] M. B. Cloosterman, L. Hetel, N. Van De Wouw, W. Heemels, J. Daafouz, and H. Nijmeijer, “Controller synthesis for networked control systems,” *Automatica*, vol. 46, no. 10, pp. 1584–1594, 2010.

[4] M. García-Rivera and A. Barreiro, “Analysis of networked control systems with drops and variable delays,” *Automatica*, vol. 43, no. 12, pp. 2054–2059, 2007.

[5] 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.

[6] P. Naghshtabrizi, J. P. Hespanha, and A. R. Teel, “Stability of delay impulsive systems with application to networked control systems,” *Transactions of the Institute of Measurement and Control*, vol. 32, no. 5, pp. 511–528, 2010.

[7] W. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nesic, “Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance,” *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781–1796, 2010.

[8] D. Nesic and A. R. Teel, “Input-output stability properties of networked control systems,” *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1650–1667, 2004.

[9] J. Hespanha, P. Naghshtabrizi, and X. Yonggang, “A survey of recent results in networked control systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, January 2007.
[10] W. Heemels, N. van de Wouw, R. Gielen, M. Donkers, L. Hetel, S. Olaru, M. Lazar, J. Daafouz, and S. Niculescu, “Comparison of overapproximation methods for stability analysis of networked control systems,” in Hybrid Systems: Computation and Control, ser. Lecture Notes in Computer Science, K. Johansson and W. Yi, Eds. Berlin: Springer Verlag, 2010, vol. 6174, pp. 181–191.

[11] D. Nesic and D. Liberzon, “A unified framework for design and analysis of networked and quantized control systems,” IEEE Transactions on Automatic Control, vol. 54, no. 4, pp. 732–747, 2009.

[12] P. Naghshtabrizi and J. P. Hespanha, “Designing an observer-based controller for a network control system,” in 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC’05. IEEE, 2005, pp. 848–853.

[13] A. Chaillet and A. Bicchi, “Delay compensation in packet-switching networked control systems,” in 47th IEEE Conference on Decision and Control, 2008. CDC 2008. IEEE, 2008, pp. 3620–3625.

[14] M. Donkers, W. Heemels, N. Van De Wouw, and L. Hetel, “Stability analysis of networked control systems using a switched linear systems approach,” IEEE Transactions on Automatic Control, vol. 56, no. 9, pp. 2101–2115, 2011.

[15] W. P. M. H. Heemels, D. Nesic, A. Teel, and N. Van de Wouw, “Networked and quantized control systems with communication delays,” in Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009, Dec 2009, pp. 7929–7935.

[16] R. Alur, A. D’Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss, “Compositional modeling and analysis of multi-hop control networks,” IEEE Transactions on Automatic control, vol. 56, no. 10, pp. 2345–2357, 2011.

[17] D. J. Antunes, J. P. Hespanha, and C. J. Silvestre, “Volterra integral approach to impulsive renewal systems: Application to networked control,” IEEE Transactions on Automatic Control, vol. 57, no. 3, pp. 607–619, 2012.

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

[19] N. van de Wouw, D. Nešić, and W. Heemels, “A discrete-time framework for stability analysis of nonlinear networked control systems,” Automatica, vol. 48, no. 6, pp. 1144–1153, 2012.
[20] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas, “Discrete abstractions of hybrid systems,” *Proceedings of the IEEE*, vol. 88, pp. 971–984, 2000.

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

[22] R. Alur and D. L. Dill, *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science. Berlin: Springer, April 1990, vol. 443, ch. Automata for modeling real-time systems, pp. 322–335.

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

[24] G. Lafferriere, G. J. Pappas, and S. Sastry, “O-minimal hybrid systems,” *Math. Control Signal Systems*, vol. 13, pp. 1–21, 2000.

[25] T. Brihaye and C. Michaux, “On the expressiveness and decidability of o-minimal hybrid systems,” *Journal of Complexity*, vol. 21, no. 4, pp. 447–478, 2005.

[26] P. E. Caines and Y. J. Wei, “Hierarchical hybrid control systems: A lattice-theoretic formulation,” *Special Issue on Hybrid Systems, IEEE Transaction on Automatic Control*, vol. 43, no. 4, pp. 501–508, April 1998.

[27] X. D. Koutsoukos, P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon, “Supervisory control of hybrid systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1026–1049, July 2000.

[28] T. Moor, J. Raisch, and S. D. O’Young, “Discrete supervisory control of hybrid systems based on l-complete approximations,” *Journal of Discrete Event Dynamic Systems*, vol. 12, pp. 83–107, 2002.

[29] D. Forstner, M. Jung, and J. Lunze, “A discrete-event model of asynchronous quantised systems,” *Automatica*, vol. 38, pp. 1277–1286, 2002.

[30] A. Bicchi, A. Marigo, and B. Piccoli, “On the reachability of quantized control systems,” *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 546–563, 2002.

[31] P. Tabuada and G. Pappas, “Linear time logic control of discrete-time linear systems,” *IEEE Transactions of Automatic Control*, vol. 51, no. 12, pp. 1862–1877, 2006.

[32] L. Habets, P. Collins, and J. V. Schuppen, “Reachability and control synthesis for piecewise-affine hybrid systems on simplices,” *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 938–948, 2006.
[33] C. Belta and L. Habets, “Controlling a class of nonlinear systems on rectangles,” *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1749–1759, 2006.

[34] O. Junge, “A set oriented approach to global optimal control,” *ESAIM: Control, optimisation and calculus of variations*, vol. 10, no. 2, pp. 259–270, 2004.

[35] G. Reißig, “Computation of discrete abstractions of arbitrary memory span for nonlinear sampled systems,” in *Proc. of 12th Int. Conf. Hybrid Systems: Computation and Control (HSCC)*, vol. 5469, pp. 306–320, April 2009.

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

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

[38] 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.

[39] A. Borri, G. Pola, and M. D. Di Benedetto, “Symbolic models for nonlinear control systems affected by disturbances,” *International Journal of Control*, vol. 88, no. 10, pp. 1422–1432, September 2012.

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

[41] G. Pola, P. Pepe, M. Di Benedetto, and P. Tabuada, “Symbolic models for nonlinear time-delay systems using approximate bisimulations,” *Systems and Control Letters*, vol. 59, pp. 365–373, 2010.

[42] G. Pola, P. Pepe, and M.D. Di Benedetto, “Symbolic models for time-varying time-delay systems via alternating approximate bisimulation,” *International Journal of Robust and Nonlinear Control*, 2014, DOI: 10.1002/rnc.3204, http://arxiv.org/abs/1011.5835. To appear.

[43] A. Girard and G. Pappas, “Approximate bisimulation: a bridge between computer science and control theory,” *European Journal of Control*, vol. 17, no. 5–6, pp. 568–578, 2011.

[44] 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. [Online]. Available: http://doi.acm.org/10.1145/2185632.2185670
In this appendix, we recall some notions of approximate equivalence and composition that are used in the paper.

Definition 5 [38] 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) $\forall x_1 \in X_{1,0} \ \exists x_2 \in X_{0,2} \ \text{such that} \ (x_1, x_2) \in \mathcal{R}$, and (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 $\varepsilon$-approximate simulation relation from $S_1$ to $S_2$ if it enjoys conditions (i), (ii) and the following one: (iii) $\forall (x_1, x_2) \in \mathcal{R}$ if $x_1 \xrightarrow{u_1} x'_1$ then $\exists x_2 \xrightarrow{u_2} x'_2$ such that $(x'_1, x'_2) \in \mathcal{R}$. System $S_1$ is $\varepsilon$-simulated by $S_2$ or $S_2$ $\varepsilon$-simulates $S_1$, denoted $S_1 \preceq_\varepsilon S_2$, if there exists an $\varepsilon$-approximate simulation relation from $S_1$ to $S_2$. Relation $\mathcal{R}$ is an $\varepsilon$-approximate bisimulation relation between $S_1$ and $S_2$ if $\mathcal{R}$ is an $\varepsilon$-approximate
Definition 6 [21] Consider a pair of metric systems \( S_i = (X_i, X_{0,i}, U_i, Y_i, H_i) \) \( (i = 1, 2) \) with the same output sets \( Y_1 = Y_2 \) and metric \( d \), and let \( \varepsilon \in \mathbb{R}_0^+ \) be a given precision. Let \( \mathcal{R} \) be an \( \varepsilon \)-simulation relation from \( S_2 \) to \( S_1 \). The \( \varepsilon \)-approximate feedback composition of \( S_1 \) and \( S_2 \), with composition relation \( \mathcal{R} \), is the system \( S_1 \times^R S_2 = (X, X_0, U, Y, H) \), where \( X = \mathcal{R}^{-1} \), \( X_0 = X \cap (X_{0,1} \times X_{0,2}) \), \( U = U_1 \), \( (x_1, x_2) \xrightarrow{u_2} (x'_1, x'_2) \) if \( x_1 \xrightarrow{u_1} x_1' \) and \( x_2 \xrightarrow{u_2} x_2' \) for some \( u_2 \in U_2 \), \( Y = Y_1 \), and \( H(x_1, x_2) = H_1(x_1) \) for any \( (x_1, x_2) \in X \).

For details on the above notions, see [21, 38]. Interaction between systems is formalized hereafter.

Lemma 1 [21] Let \( S_i = (X_i, X_{0,i}, U_i, Y_i, H_i) \) \( (i = 1, 2, 3) \) be metric systems with the same output sets \( Y_1 = Y_2 = Y_3 \) and metric \( d \). Then, the following statements hold: (i) for any \( \varepsilon_1 \leq \varepsilon_2 \), \( S_1 \preceq_{x_1} S_2 \) implies \( S_1 \preceq_{x_2} S_2 \); (ii) if \( S_1 \preceq_{x_1} S_2 \) and \( S_2 \preceq_{x_2} S_3 \) then \( S_1 \preceq_{x_{12}+x_{23}} S_3 \); (iii) for any \( \varepsilon \in \mathbb{R}_0^+ \) and any \( \varepsilon \)-A simulation relation \( \mathcal{R} \) from \( S_2 \) to \( S_1 \), \( S_1 \times^R S_2 \preceq_{x} S_2 \).