Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

On symbolic control design of nonlinear systems with dynamic regular language specifications [★]

Tommaso Masciulli [*] Giordano Pola [*]

[*] Department of Information Engineering, Computer Science and Mathematics, Center of Excellence for Research DEWS, University of L'Aquila, 67100, L'Aquila, Italy, (e-mails tommaso.masciulli@graduate.univaq.it, giordano.pola@univaq.it)

Abstract: Formal methods are becoming rather popular in the research community working on hybrid systems because they provide a systematic approach to design complex and heterogeneous systems of interest in e.g. industrial world. In this paper we consider a control problem where the plant is a nonlinear system, the controller is a finite state machine, easily implementable in digital devices, and the specification is a regular language and, it is dynamic. The motivation for considering dynamic specifications comes from some relevant and concrete applications where environment, external to the plant, may change in time and therefore designed controllers need to timely reconfigure to properly deal with new scenario. We propose an approach to reduce online computations for controller reconfiguration which exhibits gain in terms of time computational complexity. The results we present are based on the use of symbolic models and on regular language theory.

Keywords: nonlinear systems, quantized systems, symbolic control, regular languages, dynamic specifications.

1. INTRODUCTION

In the last twenty years, researchers working in the area of hybrid systems have explored formal methods as a tool for addressing control design of complex and heterogeneous systems arising in many applications of interest. Main advantage of this approach is twofold: (i) its capability to handle logic specifications, relevant in many concrete applications, which are difficult to enforce by using traditional control design techniques and, (ii) controllers designed are provably–correct in the sense that they can take into account non–idealities at the software and hardware implementation level. Central to this approach is the construction of symbolic models that approximate purely continuous or hybrid plants. A symbolic model is an abstract description of a purely continuous or hybrid system where each state corresponds to an aggregate of continuous/hybrid states and each label to an aggregate of continuous/hybrid inputs. The literature on symbolic models is very rich, see e.g. (Tabuada (2009); Belta et al. (2017); Pola and Di Benedetto (2019)) and the references therein. The use of symbolic models for control design purposes has been investigated, among many others, in the following papers: (Tabuada and Pappas (2006); Gol et al. (2014)) consider discrete–time (d.t.) linear control systems and linear temporal logic (LTL) specifications; (Reissig and Rungger (2014)), continuous–time (c.t.) nonlinear systems and general behavioral specifications; (Tabuada (2008)), c.t. nonlinear systems and regular language specifications; (Pola et al. (2012)), c.t. nonlinear systems and transition systems specifications, design of efficient on–the–fly inspired control algorithms is also explored; (Girard (2012, 2013)), c.t. nonlinear switched systems and safety and reachability specifications; (Yordanov et al. (2012)), d.t. piecewise affine systems and LTL specifications; (Pola and Di Benedetto (2014)), d.t. piecewise affine systems and transition systems specifications; (Borri et al. (2019)), networked c.t. nonlinear systems and transition systems specifications; (Pola et al. (2018); Dallal and Tabuada (2015); Meyer et al. (2015); Kim et al. (2015)), decentralized symbolic control of networks of nonlinear systems. To the best of our knowledge, current results always consider specifications that are not dynamic. However, there are applications of interest as for example autonomous driving where external environment may change over time, due e.g. to moving obstacle. This environment change can be well modeled by dynamic specifications. In this paper we consider a plant described by a continuous–time nonlinear system and a controller described by a finite state machine that then, is easily implementable in digital devices. Specifications are expressed as regular languages that, as also stressed in (Tabuada (2008)), are relevant in the control design of many concrete applications. We model dynamic specifications as follows. We consider a nominal specification $\mathbb{L}_{nom}$ and the environment change is captured by two specifications $\mathbb{L}_{-}$ and $\mathbb{L}_{+}$ describing, the behavior of $\mathbb{L}_{nom}$ that at some time t becomes illegal and respectively, the behavior not included in $\mathbb{L}_{nom}$ that

[★] The research leading to these results has been partially supported by the Center of Excellence DEWS.

at time t becomes legal. Resulting specification at time t is then given by $\mathbb{L}_{new} = (\mathbb{L}_{nom} \smallsetminus \mathbb{L}-) \cup \mathbb{L}_+$. We suppose that controller $C(\mathbb{L}_{nom})$ enforcing $\mathbb{L}_{nom}$ is computed offline before the controlled plant is initialized. The control problem we consider is then to design a controller $C(\mathbb{L}_{new})$ enforcing $\mathbb{L}_{new}$ on the basis of information on controllers $C(\mathbb{L}_{nom})$, $C(\mathbb{L}_-)$ and $C(\mathbb{L}_+)$, where $C(\mathbb{L}_-)$ and $C(\mathbb{L}_+)$ enforce $\mathbb{L}_-$ and $\mathbb{L}_+$, respectively. From this problem set–up, for computing $C(\mathbb{L}_{new})$ only computation of $C(\mathbb{L}_-)$ and $C(\mathbb{L}_+)$ is needed at time t, which could be doable at run–time, provided that sizes of $\mathbb{L}_-$ and $\mathbb{L}_+$ are small, as it may be the case in many realistic scenarios of interest. This is important because, in this way, controller can reconfigure in response to external environment change that is unpredictable and hence, cannot be modeled before it happens. A computational complexity analysis is included which shows the benefits of the approach taken. Results proposed are based on the use of symbolic models and regular language theory.

This paper is organized as follows. In Section 2 we introduce notation and preliminary definitions. In Section 3 we introduce the control problem. Section 4 recalls preliminary results that are instrumental to solve our control problem in Section 5. Section 6 presents a computational complexity analysis.

2. NOTATION AND PRELIMINARY DEFINITIONS

2.1 Notation

The symbol $\mathrm{card}(X)$ denotes the cardinality of a set X. The symbols $\mathbb{N}_0$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{R}_0^+$ denote the set of nonnegative integer, integer, real, positive real, and nonnegative real numbers, respectively. Given $a, b \in \mathbb{Z}$, we denote $[a;b] = [a,b] \cap \mathbb{Z}$. Given a set X, the symbol 2^X denotes the power set of X. Given a pair of sets X and Y, let $X \smallsetminus Y = \{x \in X | x \notin Y\}$. Given a pair of sets X and Y and a relation $\mathcal{R} \subseteq X \times Y$, the symbol $\mathcal{R}^{-1}$ denotes the inverse relation of $\mathcal{R}$, i.e. $\mathcal{R}^{-1} = \{(y,x) \in Y \times X : (x,y) \in \mathcal{R}\}$. Given $X' \subseteq X$ and $Y' \subseteq Y$, we denote $\mathcal{R}(X') = \{y \in Y | \exists x \in X' \text{ s.t. } (x,y) \in \mathcal{R}\}$ and $\mathcal{R}^{-1}(Y') = \{x \in X | \exists y \in Y' \text{ s.t. } (x,y) \in \mathcal{R}\}$. Given a function $f : X \to Y$, the symbol $f^{-1} : Y \to 2^X$ denotes the inverse map of f, i.e., $f^{-1}(y) = \{x \in X : y = f(x)\}$ for all y in the co-domain of f. Given f and $X' \subseteq X$ the symbol $f(X')$ denotes the image of X' through f, i.e. $f(X') = \{y \in Y | \exists x \in X' \text{ s.t. } y = f(x)\}$. 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 $\gamma(0) = 0$; function γ is said to belong to class $\mathcal{K}_\infty$ if $\gamma \in \mathcal{K}$ and $\gamma(r) \to \infty$ as $r \to \infty$. Symbol $|a|$ denotes absolute value of $a \in \mathbb{R}$. Given a vector $x \in \mathbb{R}^n$ we denote by $x(i)$ the i–th element of x and by $\|x\|$ the infinity norm of x. Given $a \in \mathbb{R}$ and $X \subseteq \mathbb{R}^n$, the symbol aX denotes the set $\{y \in \mathbb{R}^n | \exists x \in X \text{ s.t. } y = ax\}$. Given $\theta \in \mathbb{R}^+$ and $x \in \mathbb{R}^n$, we denote

$$\mathcal{B}^n_{[-\theta,\theta[}(x) = \{y \in \mathbb{R}^n | y(i) \in [-\theta + x(i), \theta + x(i)[, i \in [1;n]\}.$$

Note that for any $\theta \in \mathbb{R}^+$, the collection of $\mathcal{B}^n_{[-\theta,\theta[}(x)$ with x ranging in $2\theta\mathbb{Z}^n$ is a partition of $\mathbb{R}^n$. We now define the quantization function. Given a positive $n \in \mathbb{N}_0$ and quantization parameter $\theta \in \mathbb{R}^+$, the quantizer in $\mathbb{R}^n$ with accuracy θ is a function $[\,\cdot\,]^n_\theta : \mathbb{R}^n \to 2\theta\mathbb{Z}^n$, associating to any $x \in \mathbb{R}^n$ the unique vector $[x]^n_\theta \in 2\theta\mathbb{Z}^n$ such that $x \in \mathcal{B}^n_{[-\theta,\theta[}([x]^n_\theta)$. Definition above naturally extends to sets $X \subseteq \mathbb{R}^n$ when $[X]^n_\theta$ is interpreted as the image of X through function $[\cdot]^n_\theta$.

2.2 Systems, approximate relations and regular languages

We start with the following

Definition 1. A system is a tuple

$$S = (X, X_0, U, \longrightarrow, X_m, Y, H), \tag{1}$$

consisting of a set of states X, a set of initial states $X_0 \subseteq X$, a set of inputs U, a transition relation $\longrightarrow \subseteq X \times U \times X$, a set of marked states $X_m \subseteq X$, a set of outputs Y and an output function $H : X \to Y$.

A transition $(x, u, x') \in \longrightarrow$ of S is denoted by $x \xrightarrow{u} x'$. Given S and $x \in X$ define set $\mathrm{Post}_S(x)$ as the collection of states $x' \in X$ for which there exists a transition $x \xrightarrow{u} x'$. The evolution of systems is captured by the notions of state, input and output runs. Given a sequence of transitions of S

$$x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} ... \xrightarrow{u_{l-1}} x_l \tag{2}$$

with $x_0 \in X_0$, the sequences

$$r_X : x_0\, x_1 \,...\, x_l,$$

$$r_U : u_0\, u_1 \,...\, u_{l-1}, \tag{3}$$

$$r_Y : H(x_0)\, H(x_1) \,...\, H(x_l), \tag{4}$$

are called a *state run*, an *input run* and an *output run* of S, respectively. System S is said to be: *empty*, if $X_0 = \varnothing$; *symbolic*, if X and U are finite sets; *metric*, if Y is equipped with a metric $\mathbf{d} : Y \times Y \to \mathbb{R}_0^+$; *deterministic*, if for any $x \in X$ and $u \in U$ there exists at most one transition $x \xrightarrow{u} x^+$ and *nondeterministic*, otherwise; *output deterministic*, if for any pair of different transitions $x \xrightarrow{u} x^+$ and $x \xrightarrow{u} z^+$ we have $H(x^+) \neq H(z^+)$ and *output nondeterministic*, otherwise; *accessible*, if for any $x \in X$ there exists a state run ending in x; *co–accessible*, if for any $x \in X$ there exists a sequence of transitions starting from x and ending in a marked state; *nonblocking*, if for any transitions sequence (2) of S with $x_0 \in X_0$ either $x_l \in X_m$ or there exists a continuation $x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} ... \xrightarrow{u_{l-1}} x_l \xrightarrow{u_l} ... \xrightarrow{u_{l'-1}} x_{l'}$ of (2) such that $x_{l'} \in X_m$, and *blocking*, otherwise. Note that if S is co–accessible, it is nonblocking while the converse is not true, in general. Given two systems

$$S_i = (X_i, X_{0,i}, U_i, \xrightarrow[i]{}, X_{m,i}, Y_i, H_i),\ i = 1,2, \tag{5}$$

system S_1 is a *sub-system* of S_2, denoted $S_1 \sqsubseteq S_2$, if all entities defining S_1 except for H_1 are subsets of corresponding entities defining S_2 except for H_2 and $H_1(x) = H_2(x)$ for any $x \in X_1$. The accessible part of a system S, denoted $\mathrm{Ac}(S)$, is the unique maximal[1] subsystem S' of S such that for any state x' of S' there exists a state run of S' ending in x'. By definition, if S is nonempty, $\mathrm{Ac}(S)$ is accessible. The co–accessible part of a system S, denoted $\mathrm{Coac}(S)$, is the unique maximal[1] subsystem S' of S such that for any state $x' \in X'$ there

[1] Here, maximality is given with respect to the pre–order naturally induced by the binary operator $\sqsubseteq$.

exists a transition sequence of S' starting from x' and ending in a marked state of S'. By definition, $\mathrm{Coac}(S)$, if not empty, is co–accessible. The trim of a system S, denoted $\mathrm{Trim}(S)$, is defined as $\mathrm{Trim}(S) = \mathrm{Coac}(\mathrm{Ac}(S)) = \mathrm{Ac}(\mathrm{Coac}(S))$. By definition, $\mathrm{Trim}(S)$, if not empty, is accessible and co–accessible, and hence, also nonblocking. In order to provide approximations of a continuous system describing a plant, we need to recall the following notion.

Definition 2. (Borri et al. (2019)) Let $S_i = (X_i, X_{0,i}, U_i, \xrightarrow[i]{}, X_{m,i}, Y_i, H_i)$ $(i = 1, 2)$ be metric systems with the same input set $U_1 = U_2$, output set $Y_1 = Y_2$ and metric $\mathbf{d}$, and let $\theta \in \mathbb{R}_0^+$ be a given accuracy. A relation $\mathcal{R} \subseteq X_1 \times X_2$ is said a strong θ-approximate simulation relation from S_1 to S_2 if it enjoys the following conditions:

(i) for every $x_1 \in X_{0,1}$ and $x_2 \in X_2$ s.t. $(x_1, x_2) \in \mathcal{R}$, it holds that $x_2 \in X_{0,2}$;
(ii) for every $x_1 \in X_{m,1}$ and $x_2 \in X_2$ s.t. $(x_1, x_2) \in \mathcal{R}$, it holds that $x_2 \in X_{m,2}$;
(iii) $\forall (x_1, x_2) \in \mathcal{R}$, $\mathbf{d}(H_1(x_1), H_2(x_2)) \leq \theta$;
(iv) $\forall (x_1, x_2) \in \mathcal{R}$, if $x_1 \xrightarrow[1]{u} x_1'$ then there exists $x_2 \xrightarrow[2]{u} x_2'$ such that $(x_1', x_2') \in \mathcal{R}$.

Relation $\mathcal{R}$ is a strong θ-approximate bisimulation relation between S_1 and S_2 if $\mathcal{R}$ is a strong θ-approximate simulation relation from S_1 to S_2 and $\mathcal{R}^{-1}$ is a strong θ-approximate simulation relation from S_2 to S_1. Systems S_1 and S_2 are strongly θ-bisimilar, denoted $S_1 \cong_\theta S_2$, if there exists a strong θ-approximate bisimulation relation $\mathcal{R}$ between S_1 and S_2.

We conclude this section by recalling from e.g. (Cassandras and Lafortune (1999)) some notions on formal language theory. Let Y be a finite set representing the alphabet. A word over Y is a finite sequence $y : y_1\, y_2 \ldots y_l$ of symbols in Y. The empty word is denoted by ε. The symbol Y^* denotes the Kleene closure of Y, that is the collection of all words over Y including ϵ. A language L over Y is a subset of Y^*. The concatenation of two words $y_1\, y_2 \ldots y_l$ and $y_{l+1}\, y_{l+2} \ldots y_{l'}$ is the word $y_1\, y_2 \ldots y_l\, y_{l+1}\, y_{l+2} \ldots y_{l'}$. The empty word is the identity element of concatenation, i.e. $\varepsilon y = y \varepsilon = y$. Similarly one can define the concatenation yL of a word y with a language L, or the concatenation $L_1 L_2$ of two languages L_1 and L_2. Given three words p, t and s and their concatenation pts, p is said a prefix of pts, t a substring of pts, and s a suffix of pts. We will use the notation s/t (read "s after t") to denote the suffix of s after its prefix t; If t is not a prefix of s, then s/t is not defined. Similarly, L/t denotes the language obtained as the collection of suffixes s of words in L after their prefix t. The prefix closure of a language L, denoted $\overline{L}$, is the collection of all prefixes of words of L. Given a system S, the *input language* (resp. *output language*) generated by S is the collection of all its input runs (resp. output runs) and is denoted as $\mathcal{L}^u(S)$ (resp. $\mathcal{L}^y(S)$). The *marked input language* (resp. *marked output language*) of S, denoted as $\mathcal{L}_m^u(S)$ (resp. $\mathcal{L}_m^y(S)$), is the collection of all input runs r_U in (3) (resp. output runs r_Y in (4)) such that the corresponding transitions sequence in (2) is with ending state $x_l \in X_m$. Following e.g. (Cassandras and Lafortune (1999)), a language L over a finite set U is *regular* if there exists a symbolic system S with input set U such that $L = \mathcal{L}_m^u(S)$.

3. CONTROL PROBLEM FORMULATION

The control scheme we consider in this paper consists of a plant P, a controller C, and a Zero order Holder (ZoH). The plant P is described by a nonlinear control system:

$$P : \begin{cases} \dot{x}(t) = f(x(t), u(t)), \\ x(t) \in \mathbf{X} \subseteq \mathbb{R}^n, \\ x(0) \in \mathbf{X}_0 \subseteq \mathbf{X}, \\ u(t) \in \mathbf{U} \subseteq \mathbb{R}^m, t \in \mathbb{R}_0^+, \end{cases} \tag{6}$$

where $x(t)$ and $u(t)$ denote, respectively, the state and the control input at time $t \in \mathbb{R}_0^+$; $\mathbf{X}$ is the state space; $\mathbf{X}_0$ is the set of initial states; $\mathbf{U}$ is the input set; $f : \mathbf{X} \times \mathbf{U} \to \mathbb{R}^n$ is a continuous map satisfying the following Lipschitz assumption: for every compact set $K \subseteq \mathbf{X}$, there exists a constant $\kappa \in \mathbb{R}^+$ such that $\|f(x, u) - f(x', u)\| \leq \kappa \|x - x'\|$ for all $x, x' \in K$ and all $u \in \mathbf{U}$. Since we are interested in controlling the plant P through a digital and quantized controller, we assume that the set $\mathbf{U}$ is finite and denote by $\mathcal{U}$ the set of piecewise constant functions u from $\mathbb{R}_0^+$ to $\mathbf{U}$ such that $u(t) = u(k\tau)$ for all $t \in [k\tau, (k+1)\tau[$ and $k \in \mathbb{N}_0$, where $\tau \in \mathbb{R}^+$ is the clock period of the microprocessor implementing the controller. Given $t_f \in \mathbb{R}^+$, a function $x : [0, t_f] \to \mathbb{R}^n$ is said to be a *state trajectory* of P if there exists $u \in \mathcal{U}$ satisfying $\dot{x}(t) = f(x(t), u(t))$, for almost all $t \in [0, t_f]$. We also denote by $\mathbf{x}(t, x_0, u)$ the state reached at time t under the input u from initial condition x_0; this state is uniquely determined, since the assumptions on f ensure existence and uniqueness of trajectories. Control system P is said to be forward complete if a trajectory exists for any initial state $x(0)$ and any time horizon $t_f \in \mathbb{R}^+$. Controller C is given in the form of a symbolic system in the sense of Definition 1, as follows:

$$C = (X_c, X_{c,0}, \mathbf{U}, \xrightarrow[c]{}, X_{c,m}, Y_c, H_c). \tag{7}$$

From the definition above, controller C is nondeterministic, in general. A Zero order Holder (ZoH) block is placed in between P and C and is described by:

$$u(t) = u_k, t \in [k\tau, (k+1)\tau[, \, k \in \mathbb{N}_0, \tag{8}$$

where

$$\mathrm{Trans}_c : \, x_{c,0} \xrightarrow[c]{u_0} x_{c,1} \xrightarrow[c]{u_1} \ldots \tag{9}$$

is a (not unique) sequence of transitions of the controller C with $x_{c,0} \in X_{c,0}$. We denote by P^C the controlled system obtained by coupling the Eqns. (6), (7) and (8). A trajectory of P^C is then given by a pair $(x(\cdot), \mathrm{Trans}_c)$ satisfying (6), (8) and (9). If controller C is nondeterministic, controlled system P^C is nondeterministic, as well. A typical symbolic control problem with regular language specifications is recalled hereafter. Consider a finite subset $\mathcal{Y}$ of the state space $\mathbf{X}$ of P and a specification expressed as a regular language $L \subseteq \mathcal{Y}^*$, where we recall $\mathcal{Y}^*$ is the Kleene closure of $\mathcal{Y}$.

Problem 3. Given the plant P in (6), $\tau \in \mathbb{R}^+$, the specification L and a desired accuracy $\theta \in \mathbb{R}^+$, find a quantization parameter $\eta \in \mathbb{R}^+$, a controller C as in (7) and a relation $\mathcal{R}^0 \subseteq \mathbf{X}_0 \times X_{c,0}$ such that for any trajectory $(x(\cdot), \mathrm{Trans}_c)$ of P^C with pair of initial states $(x(0), x_{c,0}) \in \mathcal{R}^0$, there exist $k_f \in \mathbb{N}_0$ and a word $q_0 q_1 \ldots q_{k_f} \in L$ such that $\|x(k\tau) - q_k\| \leq \theta$, for all $k \in [0; k_f]$.

Problem above has been solved in (Pola et al. (2018)) for incrementally stable discrete–time nonlinear control systems. The solution to the control problem above is specified in the sequel by a triplet $(\eta(L), C(L), \mathcal{R}^0(L))$ where we emphasize its dependence on the specification L. For later purposes we also set:

$C(L) = (X_c(L), X_{c,0}(L), \mathbf{U}, \xrightarrow[c(L)]{}, X_{c,m}(L), Y_c(L), H_c(L))$.

We can now introduce the control problem we consider in this paper. Consider a regular language specification $L_{nom} \subseteq \mathcal{Y}^*$, representing the nominal specification that we want to enforce on the plant P. Let $C(L_{nom})$ be the controller that solves Problem 3 with $L = L_{nom}$ and suppose that it has been computed offline before the controlled system $P^{C(L_{nom})}$ is initialized. Suppose at some time $\mathbf{t} = \mathbf{k}\tau$, $\mathbf{k} \in \mathbb{N}_0$, word $\mathbf{w}: \ w_0 w_1 ... w_{\mathbf{k}} \in \overline{L_{nom}}$ (where we recall $\overline{L_{nom}}$ is the prefix closure of regular language L_{nom}) has been enforced by the controller $C(L_{nom})$ on the plant P, meaning that $\|x(k\tau) - w_k\| \leq \theta, \forall k \in [0;\mathbf{k}]$, where $x(.)$ is the state trajectory of the plant in $P^{C(L_{nom})}$. Suppose that at time $\mathbf{t}$ external environment changes and: some words $\mathbf{w}\,\mathbf{q}_-$ of L_{nom}, obtained by the concatenation of $\mathbf{w}$ and $\mathbf{q}_- \in \mathcal{Y}^*$, become illegal; some words $\mathbf{w}\,\mathbf{q}_+ \in \mathcal{Y}^*$ that were illegal, i.e. $\mathbf{w}\,\mathbf{q}_+ \notin L_{nom}$, become legal. The resulting new specification L_{new} at time $\mathbf{t}$ can be formalized by $L_{new} = ((L_{nom}/\mathbf{w}) \smallsetminus L_-) \cup L_+$, where (we recall that $L_{nom}/\mathbf{w}$ is the language collecting suffixes of words in L_{nom} after their prefix $\mathbf{w}$, and):

- $L_- \subseteq \mathcal{Y}^*$ is such that regular language $\mathbf{w}L_-$ collects all words of L_{nom} that are illegal at time $\mathbf{t}$;
- $L_+ \subseteq \mathcal{Y}^*$ is such that regular language $\mathbf{w}L_+$ collects all words of $\mathcal{Y}^* \smallsetminus L_{nom}$ that are legal at time $\mathbf{t}$.

One could then find in principle the controller $C(L_{new})$ to solve Problem 3 with $L = L_{new}$. However, this approach is not efficient from the computational complexity point of view because in many practical applications, the change of environment causes small modifications to the nominal specification L_{nom} for which the controller $C(L_{nom})$ is supposed to be already available. For this reason in this paper we consider the following control problem:

Problem 4. Find $C(L_{new})$ on the (only) basis of knowledge of $C(L_{nom})$, $C(L_-)$, $C(L_+)$ and the state $x_{c,\mathbf{k}}$ reached by the controller $C(L_{nom})$ in $P^{C(L_{nom})}$ at step $\mathbf{k}$.

We point out that in the problem above, only controllers $C(L_-)$ and $C(L_+)$ need to be computed at time $t = \mathbf{k}\tau$. Provided that sizes of L_- and L_+ are small enough, such a computations could be performed at run–time. This is important because, in this way, controller can reconfigure in response to external environment change that is unpredictable and hence, cannot be modeled before it happens. Solution to Problem 4 requires solution to Problem 3 that is addressed in the next section.

4. SOLUTION TO PROBLEM 3

Results reported in this section are adapted from (Pola et al. (2018)). We start by providing a symbolic system approximating the plant P. To this purpose we first proceed with a time discretization and then with a state space quantization of P. Given the clock period $\tau \in \mathbb{R}^+$, define $S_\tau(P) = (X_\tau, X_{0,\tau}, U_\tau, \xrightarrow[\tau]{}, X_{m,\tau}, Y_\tau, H_\tau)$, where $X_\tau = X_{0,\tau} = X_{m,\tau} = \mathbf{X}$, $U_\tau = \mathcal{U}$, transition $x \xrightarrow[\tau]{u} x'$ if $x' = \mathbf{x}(\tau, x, u)$, $Y_\tau = \mathbf{X}$ and $H_\tau(x) = x$ for all $x \in X_\tau$. We now proceed with the state space quantization of $S_\tau(P)$. Given P, $\tau \in \mathbb{R}^+$ and a state space quantization $\eta \in \mathbb{R}^+$, define $S_{\tau,\eta}(P) = (X_{\tau,\eta}, X_{0,\tau,\eta}, U_{\tau,\eta}, \xrightarrow[\tau,\eta]{}, X_{m,\tau,\eta}, Y_{\tau,\eta}, H_{\tau,\eta})$, where $X_{\tau,\eta} = X_{0,\tau,\eta} = X_{m,\tau,\eta} = [\mathbf{X}]^n_\eta$, $U_{\tau,\eta} = \mathcal{U}$, $\xi \xrightarrow[\tau,\eta]{u} \xi'$ if $\xi' = [\mathbf{x}(\tau, \xi, u)]^n_\eta$, $Y_{\tau,\eta} = \mathbf{X}$ and $H_{\tau,\eta}(x) = x$ for all $x \in X_{\tau,\eta}$. Intuitively, this definition corresponds to replacing any state of $S_\tau(P)$ by its quantization. System $S_{\tau,\eta}(P)$ is countable and becomes symbolic when the set $\mathbf{X}$ is bounded. We now recall the following notion.

Definition 5. A smooth function $V: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+_0$ is an incrementally globally asymptotically stable (δ–GAS) *Lyapunov function* for plant P, if there exist $\kappa \in \mathbb{R}^+$ and $\mathcal{K}_\infty$ functions α_1 and α_2 such that for any $x, x' \in \mathbb{R}^n$ and any $u \in U$:
(i) $\alpha_2(\|x - x'\|) \leq V(x, x') \leq \alpha_1(\|x - x'\|)$;
(ii) $\frac{\partial V}{\partial x} f(x, u) + \frac{\partial V}{\partial x'} f(x', u) < -\kappa V(x, x')$.

We also assume the existence of a $\mathcal{K}_\infty$ function γ such that

$$\forall x, y, z \in \mathbb{R}^n, \ |V(x, y) - V(x, z)| \leq \gamma(\|y - z\|). \quad (10)$$

Note that γ is not a function of the variable x. This assumption is not restrictive provided that state space $\mathbf{X}$ of P is bounded, as it is the case in concrete applications. We can now recall the following result.

Theorem 6. Consider the control system P and suppose it admits a δ–GAS Lyapunov function V and, as such, satisfying conditions of Definition 5, for some $\kappa \in \mathbb{R}^+$ and $\mathcal{K}_\infty$ functions α_1 and α_2 and (10) for some $\mathcal{K}_\infty$ function γ. Then, for any desired accuracy $\theta \in \mathbb{R}^+$ and any $\tau \in \mathbb{R}^+$, select quantization parameter $\eta \in \mathbb{R}^+$ satisfying:

$$\eta \leq \min\left\{\gamma^{-1}((1 - e^{-\kappa\tau})\alpha_1(\theta)), (\alpha_2^{-1} \circ \alpha_1)(\theta)\right\}. \quad (11)$$

Then, relation $\mathcal{R}_\theta \subseteq X_\tau \times X_{\tau,\eta}$ specified by $(x, \xi) \in \mathcal{R}_\theta$ iff $V(x, \xi) \leq \alpha_1(\theta)$ is a strong θ–approximate bisimulation relation between $S_\tau(P)$ and $S_{\tau,\eta}(P)$. Consequently, $S_\tau(P)$ and $S_{\tau,\eta}(P)$ are strongly θ–bisimilar.

We now use system $S_{\tau,\eta}(P)$ to derive a solution to Problem 3. To this purpose, we first need to reformulate the specification L. Since language L is regular there exists a symbolic system $S'_L = (X'_L, X'_0, Y_L, \xrightarrow[\prime,L]{}, X'_{L,m}, Y'_L, H'_L)$ such that its marked input language coincides with L, i.e., $\mathcal{L}^u_m(S'_L) = L$. Without loss of generality, S'_L is chosen as deterministic, accessible and nonblocking, see e.g. (Cassandras and Lafortune (1999)). It is useful to define the dual symbolic system S_L of system S'_L, where states of S_L are transitions of S'_L and vice versa, as follows:

Definition 7. Given system S'_L, define the dual system

$$S_L = (X_L, X_{L,0}, U_L, \xrightarrow[L]{}, X_{L,m}, \mathbf{X}, H_L), \quad (12)$$

where: X_L coincides with the set $\xrightarrow[\prime,L]{}$ of transitions of S'_L; $X_{L,0}$ is the collection of states $x'_L \xrightarrow[\prime,L]{u'_L} x'^{,+}_L$ in X_L with $x'_L \in X'_{L,0}$; $U_L = \{u_D\}$, where u_D is a dummy input; $\xrightarrow[L]{}$ is the collection of transitions

$$\left(x^1_L \xrightarrow[\prime,L]{u^{12}_L} x^2_L\right) \xrightarrow[L]{u_L} \left(x^3_L \xrightarrow[\prime,L]{u^{34}_L} x^4_L\right)$$

with $x_L^2 = x_L^3$; $X_{L,m}$ is the collection of states $x'_L \xrightarrow[\prime,L]{u'_L} x_L^{\prime,+}$ in X_L with $x_L^{\prime,+} \in X'_{L,m}$; $H_L(x'_L \xrightarrow[\prime,L]{u'_L} x_L^{\prime,+}) = u'_L$ for any state $x'_L \xrightarrow[\prime,L]{u'_L} x_L^{\prime,+}$ in X_L.

For ease of notation, we denote a state of S_L by x_L and a transition of S_L as $x_L \xrightarrow[L]{u_D} x_L^+$. Since S'_L is accessible and nonblocking, it is easy to see that also S_L is accessible and nonblocking. Moreover, since S'_L is deterministic then S_L is output deterministic. From the definitions above, it is easy to see that $\mathcal{L}_m^y(S_L) = \mathcal{L}_m^u(S'_L) \smallsetminus \{\varepsilon\}$; when $X_{L,0} \cap X_{L,m} = \varnothing$, which occurs in many realistic situations, we have $\mathcal{L}_m^y(S_L) = \mathcal{L}_m^u(S'_L)$. We now select transitions from S_L which can be followed by system $S_{\tau,\eta}(P)$, up to accuracy η. Given the dual system S_L in (12) define symbolic system $S''_L = (X_L, X_{0,L}, \mathbf{U}, \xrightarrow[\prime\prime,L]{}, X_{L,m}, \mathbf{X}, H_L)$, where $x_L \xrightarrow[\prime\prime,L]{u} x_L^+$ if $[H_L(x_L)]_\eta^n \xrightarrow[\tau,\eta]{u} [H_L(x_L^+)]_\eta^n$. We now have all the ingredients to give the solution to Problem 3. Given S''_L define:

$$C(L) = \mathrm{Trim}(S''_L), \tag{13}$$

$$\mathcal{R}^0(L) = \left\{ \begin{array}{l} (x_0, x_{c,0}) \in \mathbf{X}_0 \times X_{c,0}(L) | \\ (x_0, [H_\mathrm{T}(x_{c,0})]_\eta^n) \in \mathcal{R}_\theta \end{array} \right\}, \tag{14}$$

and suppose that the following holds

Assumption 8. $C(L)$ and $\mathcal{R}^0(L)$ are not empty.

The following result shows formal correctness of the procedure above to solve Problem 3.

Theorem 9. Suppose that plant P admits a δ–GAS Lyapunov function V and as such, satisfies conditions of Definition 5, for some $\kappa \in \mathbb{R}^+$ and $\mathcal{K}_\infty$ functions α_1 and α_2 and (10) for some $\mathcal{K}_\infty$ function γ. For any desired accuracy $\theta \in \mathbb{R}^+$ and clock period $\tau \in \mathbb{R}^+$ select $\mu \in \mathbb{R}^+$ and $\eta(L) \in \mathbb{R}^+$ satisfying (11) with $\eta = \eta(L)$ and $\mu + \eta(L)/2 \leq \theta$. Suppose that Assumption 8 holds. Then, parameter $\eta(L)$, controller $C(L)$ in (13) and relation $\mathcal{R}^0(L)$ in (14) solve Problem 3.

For later purposes we now give the following:

Definition 10. Language $\mathbf{L}(P^{C(L)})$ is the collection of all words $q_0 q_1 ... q_{t_f} \in L$ for which there exists a trajectory $(x(\cdot), \mathrm{Trans}_c)$ of $P^{C(L)}$ with $(x(0), x_{c,0}) \in \mathcal{R}^0(L)$ and satisfying $\|x(k\tau) - q_k\| \leq \theta$ for all times $t \in [0; t_f]$.

By the definition above, $\mathbf{L}(P^{C(L)})$ represents the part of L that can be enforced by $C(L)$ on P. The following result holds as a direct consequence of Theorem 9.

Corollary 11. Suppose that assumptions of Theorem 9 hold and select η as required in Theorem 9. Suppose that Assumption 8 holds. Then, $\mathbf{L}(P^{C(L)}) = \mathcal{L}_m^y(C(L))$.

It is easy to see that there can be the case that $\mathbf{L}(P^{C_1(L)}) = \mathbf{L}(P^{C_2(L)})$ for controllers $C_1(L)$ and $C_2(L)$ with $C_1(L) \neq C_2(L)$, meaning that while controllers are different, they implement the same part of the specification L. When $\mathbf{L}(P^{C_1(L)}) = \mathbf{L}(P^{C_2(L)})$ we write $C_1(L) \sim_L C_2(L)$.

5. MAIN RESULT

In this section we provide the solution to Problem 4.

5.1 Preliminary results

In this section we introduce two results concerning systems marking the union of regular languages and systems marking the set difference of regular languages. Consider a pair of output deterministic systems $S_i = (X_i, X_{0,i}, U_i, \xrightarrow[i]{}, X_{m,i}, Y_i, H_i)$, $i = 1, 2$, with $X_1 \cap X_2 = \varnothing$ and let $\mathcal{L}_m^y(S_i) = L_i$, $i = 1, 2$. We start with the following definition of union of systems:

Definition 12. The union of S_1 and S_2, denoted $S_1 \sqcup S_2$, is specified by system S as in (1), where: $X = X_1 \cup X_2$; $X_0 = X_{0,1} \cup X_{0,2}$; $\xrightarrow{} = \xrightarrow[1]{} \cup \xrightarrow[2]{}$; $X_m = X_{m,1} \cup X_{m,2}$; $Y = Y_1 \cup Y_2$; $H(x) = H_1(x)$ if $x \in X_1 \wedge H(x) = H_2(x)$ if $x \in X_2$.

The following result holds:

Proposition 13. $\mathcal{L}_m^y(S_1 \sqcup S_2) = \mathcal{L}_m^y(S_1) \cup \mathcal{L}_m^y(S_2)$.

We now address system difference. We start with the following:

Definition 14. The system difference of S_1 by S_2, denoted

$$S_1 \smallsetminus_s S_2,$$

is $\mathrm{Trim}(S)$, where S is system as in (1), with:

- $X = (X_1 \times \{x_D\}) \cup X'$, where $X' = \{(x_1, x_2) \in X_1 \times X_2 \mid H_1(x_1) = H_2(x_2)\}$ and x_D is a dummy state;
- $X_0 \subseteq X_{0,1} \times (X_{0,2} \cup \{x_D\})$ defined as follows:
    - · $\forall x_1 \in X_{0,1}, x_2 \in X_{0,2}$, if $H_1(x_1) = H_2(x_2)$ then $(x_1, x_2) \in X_0$;
    - · $\forall x_1 \in X_{0,1}$, if $\nexists x_2 \in X_{0,2}$ such that $H_1(x_1) = H_2(x_2)$ then $(x_1, x_D) \in X_0$;
- $\xrightarrow{} \subseteq X \times U_1 \times X$ defined as follows: for any $x_1 \xrightarrow[1]{u_1} x_1^+$
    - · $(x_1, x_2) \xrightarrow{u_1} (x_1^+, x_2^+)$, if $x_2 \xrightarrow[2]{u_2} x_2^+$,
    - · $(x_1, x_D) \xrightarrow{u_1} (x_1^+, x_D)$,
    - · $(x_1, x_2) \xrightarrow{u_1} (x_1^+, x_D)$, if there does not exist $x_2 \xrightarrow[2]{u_2} x_2^+$ with $(x_1^+, x_2^+) \in X'$;
- $X_m = (X_{m,1} \times \{x_D\}) \cup X'_m$ where $X'_m = (X_{m,1} \times (X_2 \smallsetminus X_{m,2})) \cap X'$, in other words $(x_1, x_2) \in X_m$ if $(x_1, x_2) \in X'$ and $x_1 \in X_{m,1} \wedge x_2 \notin X_{m,2}$;
- $Y = Y_1$;
- $H(x_1, x_2) = H_1(x_1)$.

The following result holds:

Proposition 15. $\mathcal{L}_m^y(S_1 \smallsetminus_s S_2) = \mathcal{L}_m^y(S_1) \smallsetminus \mathcal{L}_m^y(S_2)$.

5.2 Solution to Problem 4

In this section we provide the solution to Problem 4. Without loss of generality we make the following:

Assumption 16. The sets of states $X(L_{nom})$, $X(L_+)$, $X(L_-)$, respectively of controllers $C(L_{nom})$, $C(L_+)$ and $C(L_-)$, have empty intersection.

Let C_{next} be a controller coinciding with $C(L_{nom})$ except for the set of initial states, that is $X_{0,next} = \mathrm{Post}_{C(L_{nom})}(x_{c,\mathbf{k}})$. Define regular language $L_{next} = \{l = l_0\, l_1 \ldots, l_J \in \mathcal{Y}\mathcal{Y}^* \mid \mathbf{w}l \in L_{nom} \wedge [w_\mathbf{k}]_\eta^n \xrightarrow[\tau,\eta]{u} [l_0]_\eta^n\}$, where we recall that $\mathbf{w} = w_0\, w_1 \ldots w_\mathbf{k}$.

Proposition 17. $C_{next} \sim_{L_{next}} C(L_{next})$

Without loss of generality we make the following:

Assumption 18. Set $\widetilde{L} = L_{next} \cup L_{+}$. The sets of states $X(\widetilde{L} \smallsetminus L_{-})$, $X(L_{-} \smallsetminus \widetilde{L})$ and $X(\widetilde{L} \cap L_{-})$ respectively of controllers $C(\widetilde{L} \smallsetminus L_{-})$, $C(L_{-} \smallsetminus \widetilde{L})$ and $C(\widetilde{L} \cap L_{-})$, have empty intersection.

We can now give the main result of this paper.

Theorem 19. Controller $(C_{next} \sqcup C(L_{+})) \smallsetminus_{s} C(L_{-})$ solves Problem 4.

6. COMPUTATIONAL COMPLEXITY ANALYSIS

In this section we provide an analysis of space computational complexity (S.c.c.) and time computational complexity (T.c.c.) associated with Problem 4. In the sequel, S.c.c. of a symbolic system S as in (1), is evaluated as $\mathrm{card}(\xrightarrow{\quad})$. When S.c.c. does not depend on the size of the entities involved we say that it is constant. We start with the following:

Proposition 20. S.c.c. and T.c.c. for finding $C(L)$ of Problem 3 are $\mathrm{card}(\xrightarrow[{}^{\prime},L]{}) + \mathrm{card}(\xrightarrow[\tau,\eta]{})$ and $\mathrm{card}(\xrightarrow[{}^{\prime},L]{})\,\mathrm{card}(\xrightarrow[\tau,\eta]{})$, respectively.

We can now give the main result of this section.

Theorem 21. S.c.c. for solving Problem 4 is:

$$\begin{aligned} 2\,\mathrm{card}(\xrightarrow[c_{next}]{}) + \mathrm{card}(\xrightarrow[L_{-}]{}) + \\ \mathrm{card}(\xrightarrow[L_{+}]{}) + \mathrm{card}(\xrightarrow[\tau,\eta]{}). \end{aligned} \tag{15}$$

T.c.c. for solving Problem 4 is:

$$\begin{aligned} (\mathrm{card}(\xrightarrow[L_{-}]{}) + \mathrm{card}(\xrightarrow[L_{+}]{}))\,\mathrm{card}(\xrightarrow[\tau,\eta]{}) + \\ \mathrm{card}(\xrightarrow[c_{next}]{})\,\mathrm{card}(\xrightarrow[c(L_{-})]{}). \end{aligned} \tag{16}$$

A traditional approach to solve Problem 4 consists in solving Problem 3 with $L = L_{new}$ whose S.c.c. and T.c.c., by Proposition 20, are, respectively

$$\mathrm{card}(\xrightarrow[L_{new}]{}) + \mathrm{card}(\xrightarrow[\tau,\eta]{}), \tag{17}$$

$$\mathrm{card}(\xrightarrow[L_{new}]{})\,\mathrm{card}(\xrightarrow[\tau,\eta]{}). \tag{18}$$

A comparison between the two approaches follows. First note that:

$$\mathrm{card}(\xrightarrow[L_{new}]{}) \leq \mathrm{card}(\xrightarrow[\tau,\eta]{}). \tag{19}$$

Suppose:

$$\mathrm{card}(\xrightarrow[L_{-}]{}), \mathrm{card}(\xrightarrow[L_{+}]{}) << \mathrm{card}(\xrightarrow[L_{new}]{}). \tag{20}$$

Condition above corresponds to the case where environment change is very small with respect to the nominal situation, which is the case in many realistic scenarios. Since (20) implies $\mathrm{card}(\xrightarrow[L_{-}]{}), \mathrm{card}(\xrightarrow[L_{+}]{}) << \mathrm{card}(\xrightarrow[L_{nom}]{})$ and $\mathrm{card}(\xrightarrow[L_{nom}]{}) \sim \mathrm{card}(\xrightarrow[L_{new}]{}) \sim \mathrm{card}(\xrightarrow[c(L_{new})]{}) \sim \mathrm{card}(\xrightarrow[c_{next}]{})$, by comparing (15) and (17), we obtain that S.c.c. in the two approaches is of the same order of computational complexity. By comparing (16) and (18), under (19) and (20), we obtain that T.c.c. of Problem 4 is smaller than the one obtained by using a traditional approach.

REFERENCES

Belta, C., Yordanov, B., and Aydin Gol, E. (2017). *Formal Methods for Discrete-Time Dynamical Systems.* Springer, 2017.

Borri, A., Pola, G., and Di Benedetto, M.D. (2019). Design of symbolic controllers for networked control systems. *IEEE Transactions on Automatic Control,*, 64, 1034–1046.

Cassandras, C. and Lafortune, S. (1999). *Introduction to Discrete Event Systems.* Kluwer Academic Publishers.

Dallal, E. and Tabuada, P. (2015). On compositional symbolic controller synthesis inspired by small-gain theorems. In *54th IEEE Conference on Decision and Control*, 6133–6138. Osaka, Japan.

Girard, A. (2012). Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5), 947–953.

Girard, A. (2013). Low-complexity quantized switching controllers using approximate bisimulation. *Nonlinear Analysis: Hybrid Systems*, 10, 34–44.

Gol, E., Lazar, M., and Belta, C. (2014). Language–guided controller synthesis for linear systems. *IEEE Transactions of Automatic Control*, 59(5), 1163–1176.

Kim, E., Arcak, M., and Seshia, S. (2015). Compositional controller synthesis for vehicular traffic networks. In *54th IEEE Conference on Decision and Control*, 6165–6171. Osaka, Japan.

Meyer, P., Girard, A., and Witrant, E. (2015). Safety control with performance guarantees of cooperative systems using compositional abstractions. In *Analysis and Design of Hybrid Systems ADHS*, 317–322. Atlanta, GA, USA.

Pola, G., Borri, A., and Di Benedetto, M.D. (2012). Integrated design of symbolic controllers for nonlinear systems. *IEEE Transactions on Automatic Control*, 57(2), 534 –539. doi:10.1109/TAC.2011.2164740.

Pola, G. and Di Benedetto, M.D. (2014). Symbolic models and control of discrete–time piecewise affine systems: An approximate simulation approach. *IEEE Transactions of Automatic Control*, 59(1), 175–180.

Pola, G. and Di Benedetto, M.D. (2019). Control of cyber-physical-systems with logic specifications: A formal methods approach. *Annual Reviews in Control*, 47, 178–192.

Pola, G., Pepe, P., and Di Benedetto, M.D. (2018). Decentralized approximate supervisory control of networks of nonlinear control systems. *IEEE Transactions on Automatic Control*, 63, 2803–2817.

Reissig, G. and Rungger, M. (2014). Feedback refinement relations for symbolic controller synthesis. In *53rd IEEE Conference on Decision and Control*, 88–94.

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

Tabuada, P. (2009). *Verification and Control of Hybrid Systems: A Symbolic Approach.* Springer, 2009.

Tabuada, P. and Pappas, G. (2006). Linear time logic control of discrete-time linear systems. *IEEE Transactions of Automatic Control*, 51(12), 1862–1877.

Yordanov, B., Tumova, J., Cerna, I., Barnat, J., and Belta, C. (2012). Temporal logic control of discrete-time piecewise affine systems. *IEEE Transactions of Automatic Control*, 57(6), 1491–1504.