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

# On data–driven controller synthesis with regular language specifications ⋆

Giordano Pola * Tommaso Masciulli * Elena De Santis *
Maria Domenica Di Benedetto *

* *Department of Information Engineering, Computer Science and Mathematics, Center of Excellence for Research DEWS, University of L'Aquila, 67100, L'Aquila, Italy, (e-mails {giordano.pola, elena.desantis, mariadomenica.dibenedetto}@univaq.it, tommaso.masciulli@student.univaq.it)*

**Abstract:** Data–driven control design for general systems with regular language specifications is addressed. We consider a discrete–time control system described as an abstract system i.e. as a collection of input–state functions. The abstract system is assumed to be suffix and concatenation closed, causal, deterministic and time–invariant. State variables are known but their dynamics are not, apart from a finite set of experiments. Given a specification expressed as a regular language defined over an alphabet consisting of a finite set of states of the plant, we design a controller based on the finite set of experiments, that guarantees that the specification is met, up to an error that can be chosen as small as desired. We also present results on maximality, convergence and adaptivity of the controller as the set of experiments increases.

*Keywords:* data–driven control, general systems, unknown systems, regular languages, transition systems, formal methods.

## 1. INTRODUCTION

There has been a growing interest in data-driven techniques for monitoring and controlling complex systems. This approach may be relevant when deriving a mathematical model of a system may be infeasible, for example in human-in-the-loop applications, or when physical models may be too complex for monitoring and control purposes. In general, system identification is widely used as a tool for controller synthesis based on data. A different approach consists in synthesizing controllers directly on the basis of data, without assuming an a–priori structure of the unknown model. The literature on data–driven control is vast. The interested reader is referred to e.g. (Bazanella et al. (2011); Hou and Wang (2013); De Persis and Tesi (2019)) and the references therein for data-driven control methods.

In this paper we address data–driven control design for general systems with regular language specifications. We consider an abstract discrete–time control system described by a collection of pairs of input and state functions. We assume that the control system only satisfies suffix and concatenation closures, causality, determinism and time–invariance. The set of states is supposed to be endowed with a metric. As such, the class of systems we consider encompasses many classes of control systems, as e.g. linear and nonlinear control systems, metric systems, hybrid systems, and infinite dimensional systems. We assume that state variables are known but their dynamics are not.

The specifications we consider are assumed to be given in terms of a desired regular language that is defined over an alphabet consisting of a finite set of states of the plant. Regular languages provide a rich framework in the control design of discrete–event systems, purely continuous and hybrid systems. In fact, they are able to represent key specifications such as reachability, safety, motion planning and collision avoidance, periodic orbits, state-based switching, and specifications involving sequences of smaller tasks that need to be performed according to a given order (see e.g. Tabuada (2009); Pola and Di Benedetto (2019)). Moreover, operators known for regular languages, and for automata recognizing them, as for example concatenation, union, intersection, and complement, see e.g. (Cassandras and Lafortune (1999)), provide a useful mean for assisting the designer in properly modeling desired complex specifications. In this paper, controller design is based only on a finite set of experiments collected on the plant, and the proposed controller enforces the specification on the control system, up to an error that can be chosen as small as desired. Results concerning maximality, convergence and adaptivity of the controller as the set of experiments gets bigger are also presented. Proofs of the results presented are omitted for lack of space. To the best of our knowledge, this is the first contribution on data–driven control design where there is no *a priori* assumption on the specific class the unknown not-identified system belongs to, and regular language specifications are considered. The existing literature concerning control design for enforcing logic specifications, as e.g. regular languages, linear temporal logic, always assumes full knowledge of the model, see e.g. (Tabuada (2009); Belta et al. (2017); Pola and Di Benedetto (2019)) and the references therein.

⋆ This research has been partially supported by the Center of Excellence DEWS.

The paper is organized as follows. In Section 2 we introduce notation and preliminary definitions. In Section 3 we formulate the control problem. In Section 4 we present the solution to the control problem. In Section 5 we address maximality of the controller, its convergence and adaptivity properties. Section 6 offers some concluding remarks.

## 2. NOTATION AND PRELIMINARY DEFINITIONS

The symbol $\vee$ denotes the logic disjunction. Symbols $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{R}_0^+$ denote the set of natural, integer, real, positive real and non–negative real numbers, respectively. Given $n_1, n_2 \in \mathbb{N}$ with $n_2 \geq n_1$, symbol $[n_1; n_2]$ denotes $\{n_1, n_1+1, ..., n_2\}$. Given two sets $X$, $[n_1; n_2] \subset \mathbb{N}$, a function $x : [n_1; n_2] \rightarrow X$ and $n \in [n_1; n_2]$, we denote by $x|_{[n_1;n]}$ the restriction of $x$ to $[n_1; n]$ that is $x|_{[n_1;n]}(i) = x(i)$, for all $i \in [n_1; n]$. Given $x \in \mathbb{R}$ and $\eta \in \mathbb{R}^+$, we denote by $[x]_\eta$ the unique real number in $\eta\mathbb{Z}$ such that $x \in [[x]_\eta - \eta/2, [x]_\eta + \eta/2[$. Given two functions $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ we denote by $f \circ g : X \rightarrow Z$ the composition function of $f$ and $g$, defined by $f \circ g(x) = f(g(x))$, for all $x \in X$. We now recall from e.g. Cassandras and Lafortune (1999) some notions on formal language theory that are used in the sequel to define logic specifications for control systems. Let $Y$ be a finite set representing the alphabet. A word over $Y$ is a finite sequence $y_1\, y_2 \ldots y_l$ of symbols in $Y$. 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 denoted by $\varepsilon$. The symbol $Y^*$ denotes the Kleene closure of $Y$, that is the collection of all words over $Y$ including $\varepsilon$. Similarly, given a word $y$ over $Y$, the symbol $\{y\}^*$ denotes the Kleene closure of word $y$, that is the collection of all words, including the empty word, obtained by concatenating $y$ with itself, an arbitrary but finite number of times. A language $L$ over $Y$ is a subset of $Y^*$. The notion of transition system that is used in the sequel to define regular languages and controllers slightly extends the one given in Tabuada (2009) to transition systems with marked states:

*Definition 1.* A transition system is a tuple

$$S = (X, X_0, U, \longrightarrow, X_m, Y, H),$$

consisting of

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

A transition $(x, u, x') \in \longrightarrow$ of $S$ is denoted by $x \xrightarrow{u} x'$. Transition system $S$ is empty if $X_0 = \varnothing$. The evolution of transition systems is captured by the notions of state, input and output runs. Given a sequence of transitions of $S$

$$x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} \ldots \xrightarrow{u_{l-1}} x_l \qquad (1)$$

with $x_0 \in X_0$, the sequences

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

$$r_U : u_0\, u_1 \ldots u_{l-1}, \qquad (2)$$

$$r_Y : H(x_0)\, H(x_1) \ldots H(x_l), \qquad (3)$$

are called a *state run*, an *input run* and an *output run* of $S$, respectively.

*Definition 2.* Transition system $S$ is said to be:

- *symbolic*, if $X$ and $U$ are finite sets;
- *metric* if $Y$ is equipped with a metric $\mathbf{d} : Y \times Y \rightarrow \mathbb{R}_0^+$;
- *deterministic*, if for any $x \in X$ and any $u \in U$ there exists at most one transition $x \xrightarrow{u} x'$ and *nondeterministic*, otherwise;
- *nonblocking*, if for any transitions sequence (1) of $S$ with $x_0 \in X_0$, either $x_l \in X_m$ or there exists a continuation

  $$x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} \ldots \xrightarrow{u_{l-1}} x_l \xrightarrow{u_l} \ldots \xrightarrow{u_{l'-1}} x_{l'}$$

  of it such that $x_{l'} \in X_m$, and *blocking*, otherwise.

The *input language* (resp. *output language*) of $S$, denoted $\mathcal{L}^u(S)$ (resp. $\mathcal{L}^y(S)$), is the collection of all its input runs (resp. output runs). 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 (2) (resp. output runs $r_Y$ in (3)) such that the corresponding transitions sequence in (1) is with ending state $x_l \in X_m$. A language $L$ over a finite set $U$ is said *regular* if there exists a symbolic transition system $S$ with input set $U$ such that $L = \mathcal{L}_m^u(S)$. By the Kleene's Theorem, any regular language can be equivalently reformulated as a regular expression, and vice versa, see e.g. Cassandras and Lafortune (1999). We also recall some operations on transition systems, naturally adapted from the ones given for discrete–event systems, see e.g. Cassandras and Lafortune (1999).

A transition system

$$S' = (X', X_0', U', \longrightarrow', X_m', Y', H') \qquad (4)$$

is said to be a *sub–transition system* of a transition system

$$S = (X, X_0, U, \longrightarrow, X_m, Y, H), \qquad (5)$$

denoted $S' \sqsubseteq S$, if $X' \subseteq X$, $X_0' \subseteq X_0$, $U' \subseteq U$, $\longrightarrow' \subseteq \longrightarrow$, $X_m' \subseteq X_m$, $Y' \subseteq Y$ and $H'(x) = H(x)$ for all $x \in X'$. Transition system $S'$ in (4) is a *strict sub–transition system* of $S$ in (5), denoted $S' \sqsubset S$, if $S' \sqsubseteq S$ and $(X' \subset X) \vee (X_0' \subset X_0) \vee (\longrightarrow' \subset \longrightarrow)$.

Consider two sub–transition systems $S_i = (X_i, X_{0,i}, U_i, \underset{i}{\longrightarrow}, X_{m_i}, Y_i, H_i)$, $i = 1, 2$, of a transition system $S$. The union between $S_1$ and $S_2$ is the transition system

$$S_1 \sqcup S_2 = (X, X_0, U, \longrightarrow, X_m, Y, H),$$

where $X = X_1 \cup X_2$, $X_0 = X_{0,1} \cup X_{0,2}$, $U = U_1 \cup U_2$, $\longrightarrow = \underset{1}{\longrightarrow} \cup \underset{2}{\longrightarrow}$, $X_m = X_{m,1} \cup X_{m,2}$, $Y = Y_1 \cup Y_2$, and $H(x) = x$ for all $x \in X$. The accessible part of a transition system $S$, denoted $\mathrm{Ac}(S)$, is the union of all sub–transition systems $S'$ 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 $S$, denoted $\mathrm{Coac}(S)$, is the union of all sub–transition systems $S'$ of $S$ such that for any state $x' \in X'$ there 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 nonblocking.

The trim of $S$, 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 nonblocking.

## 3. SYSTEM DEFINITION AND CONTROL PROBLEM STATEMENT

In this section we first define the abstract system. Then we formulate the control problem.
Let

$$\mathcal{T} = \{(T_1, T_2) \in \mathbb{N} \times \mathbb{N} \,|\, T_1 < T_2\}.$$

Let $\mathcal{U}$ be the set of input values and $\mathcal{U}^{[T_1;T_2]}$ the set of all input functions $u : [T_1; T_2] \to \mathcal{U}$, with $(T_1, T_2) \in \mathcal{T}$. Let $\mathcal{X}$ be the set of state values and $\mathcal{X}^{[T_1;T_2]}$ the set of all functions $x : [T_1; T_2] \to \mathcal{X}$, denoting the state evolutions. We consider a system described by pairs of input–state functions, as follows:

*Definition 3.* A system $P$ is a relation

$$P \subseteq \bigcup_{(T_1,T_2)\in\mathcal{T}} \left(\mathcal{U}^{[T_1;T_2-1]} \times \mathcal{X}^{[T_1;T_2]}\right) \tag{6}$$

satisfying the following properties:

- *(Suffix closure)* If $(u, x) \in P \cap (\mathcal{U}^{[T_1;T_2-1]} \times \mathcal{X}^{[T_1;T_2]})$ for some $(T_1, T_2) \in \mathcal{T}$, then any suffix $(u', x')$ of $(u, x)$ is in $P$, i.e. $(u', x') \in P \cap (\mathcal{U}^{[T_3;T_2-1]} \times \mathcal{X}^{[T_3;T_2]})$ for any $T_3 \in [T_1; T_2 - 1]$, where $u'(t) = u(t)$ for any $t \in [T_3; T_2 - 1]$ and $x'(t) = x(t)$ for any $t \in [T_3; T_2]$;
- (*Causality*) for any $T_3 \in [T_1 + 1; T_2]$ and any $u \in \mathcal{U}^{[T_1;T_3-1]}$, let
$$\mathcal{U}(u) = \{v \in \mathcal{U}^{[T_1;T_2-1]} | v(t) = u(t), \forall t \in [T_1; T_3 - 1]\}.$$
Then
$$P \cap (\{u\} \times \mathcal{X}^{[T_1;T_3]}) = \left(\mathcal{U}(u)|_{[T_1;T_3-1]}\right) \times \left(\mathcal{X}^{[T_1;T_2]}\Big|_{[T_1;T_3]}\right),$$
where
$$\mathcal{U}(u)|_{[T_1;T_3-1]}$$
denotes the collection of functions $v \in \mathcal{U}(u)$, restricted to the time interval $[T_1; T_3 - 1]$, and
$$\mathcal{X}^{[T_1;T_2]}\Big|_{[T_1;T_3]}$$
denotes the collection of functions $x \in \mathcal{X}^{[T_1;T_2]}$, restricted to the time interval $[T_1; T_3]$;
- (*Concatenation closure*) for any pairs
$$(u, x) \in P \cap \left(\mathcal{U}^{[T_1;T_2-1]} \times \mathcal{X}^{[T_1;T_2]}\right),$$
$$(u', x') \in P \cap \left(\mathcal{U}^{[T_2;T_3-1]} \times \mathcal{X}^{[T_2;T_3]}\right),$$
for some $(T_1, T_2), (T_2, T_3) \in \mathcal{T}$, and satisfying $x(T_2) = x'(T_2)$, the pair
$$(u'', x'') \in P \cap (\mathcal{U}^{[T_1;T_3-1]} \times \mathcal{X}^{[T_1;T_3]}),$$
where $u''(t) = u(t)$ for any $t \in [T_1; T_2 - 1]$, $x''(t) = x(t)$ for any $t \in [T_1; T_2]$, $u''(t) = u'(t)$ for any $t \in [T_2; T_3 - 1]$, and $x''(t) = x'(t)$ for any $t \in [T_2; T_3]$.

This definition follows the general notion of abstract systems given in System Theory, see e.g. Ruberti and Isidori (1979). System $P$ in general may be nondeterministic. We suppose that our control plant is represented by the system $P$ and satisfies the following:

*Assumption 4.* System $P$ is metric, i.e. we suppose that its set of states $\mathcal{X}$ is endowed with a metric

$$\mathbf{d} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+_0. \tag{7}$$

The system $P$ is assumed to be unknown, apart from a finite set of input-state functions $\mathcal{E}$ contained in $P$, and collected in some time intervals in $\mathcal{T}$. In the sequel we call elements of $\mathcal{E}$, as experiments. Our aim is to design a controller $C$ for $P$ on the basis of the experiments, in such a way that the controlled systems satisfies some given specifications. Since the time when the controller is designed and applied to $P$ is different from the time when the experiments in $\mathcal{E}$ have been collected, $P$ is supposed to be time–invariant:

*Assumption 5.* System $P$ is time–invariant, i.e. for any

$$(u, x) \in P \cap (\mathcal{U}^{[T_1;T_2-1]} \times \mathcal{X}^{[T_1;T_2]})$$

and any $t' \in \mathbb{N}$ we suppose

$$(u', x') \in P \cap (\mathcal{U}^{[T_1+t';T_2+t'-1]} \times \mathcal{X}^{[T_1+t';T_2+t']}),$$

where $u'(t) = u(t - t')$ and $x'(t) = x(t - t')$ for any $t \in [T_1 + t'; T_2 + t']$.

As a consequence, we simplify notations and in the sequel we consider input and state functions in $P$ starting from time 0.
Finally, we suppose that

*Assumption 6.* System $P$ is deterministic, i.e. for any $u \in \mathcal{U}^{[0;T]}$ with $(0, T) \in \mathcal{T}$ and any $\bar{x} \in \mathcal{X}$ there exists at most one $x \in \mathcal{X}^{[0;T]}$ such that $x(0) = \bar{x}$ and $(u, x) \in P$.

It follows from the definition of system and the assumptions above that $P$ is a discrete–time, time–invariant and deterministic control system. Many classes of discrete–time systems are included in this framework, as e.g. linear, nonlinear and hybrid systems, infinite dimensional systems, or quantized and sampled data control systems where the original plant is a continuous–time process.
For later purposes, we denote by $\mathcal{X}_0 \subseteq \mathcal{X}$, the set of initial states of $P$, defined as the collection of all states $x_0 \in \mathcal{X}$ for which there exists a pair $(u, x) \in P$ such that $x_0 = x(0)$.
We now define the controller $C$ in the form of a transition system in the sense of Definition 1:

$$C = (X_c, X_{c,0}, U_c, \underset{c}{\longrightarrow}, X_{c,m}, Y_c, H_c). \tag{8}$$

The plant $P$ controlled by $C$, denoted by $P^C$, is described by the collection of pairs $(u, x) \in P$, where control input $u \in \mathcal{U}^{[0;T-1]}$ is such that $u(t) = u_t, t \in [0; T - 1]$, where sequence $\{u_t\}_{t\in[0;T-1]}$ is composed of input labels in a state run

$$x_{c,0} \overset{u_0}{\underset{c}{\longrightarrow}} x_{c,1} \overset{u_1}{\underset{c}{\longrightarrow}} \ldots \overset{u_{T-1}}{\underset{c}{\longrightarrow}} x_{c,T} \tag{9}$$

of the controller $C$, ending in a marked state of $C$, i.e. $x_{c,T} \in X_{c,m}$. We say that control input $u \in \mathcal{U}^{[0;T-1]}$ is marked by $C$ and that the state run in (9) *marks* vector $(u_0, u_1, ..., u_{T-1})$.
We consider as specification a regular language $Q$ defined over a finite alphabet set $\mathbb{X} \subseteq \mathcal{X}$, where $\mathcal{X}$ is the set of states of $P$, which may represent key specifications such as e.g. reachability, safety, motion planning and collision avoidance (see e.g. Tabuada (2009); Pola and Di Benedetto (2019)). The data–driven controller synthesis problem can be stated as follows:

*Problem 7.* Consider a system $P$, a finite collection of experiments

$$\mathcal{E} \subseteq P$$

and a desired accuracy $\theta \in \mathbb{R}^+_0$. Find a controller $C$ as in (8) and a relation $\mathcal{R}_0 \subseteq \mathcal{X}_0 \times X_{c,0}$, both depending on the specification $Q$ and on the set of experiments $\mathcal{E}$ (but not on $P$ which is unknown, apart from the set of experiments $\mathcal{E}$), which enforce specification $Q$ on $P$ up to accuracy $\theta$, i.e. such that for any pairs

$$(u,x) \in P^C \cap (\mathcal{U}^{[0;T-1]} \times \mathcal{X}^{[0;T]}),$$

for some $(0,T) \in \mathcal{T}$, with $(x(0), x_{c,0}) \in \mathcal{R}_0$, where $x_{c,0}$ is the initial state of a state run (9) of $C$ marking $u$, there exists a word $q_0\, q_1\, ... q_T \in Q$ such that:

$$\mathbf{d}(x(t), q_t) \leq \theta, \; \forall t \in [0;T]. \tag{10}$$

*Remark 8.* Note that since $P$ is suffix closed, any suffix of a pair $(u,x) \in \mathcal{E}$ is in $P$ and can then be considered as an additional experiment. As a consequence, the set of experiments $\mathcal{E}$ can be enlarged with its suffixes which can then be useful in enforcing a larger part of the specification. However, in order to simplify notation, in the sequel we will implicitly assume that set $\mathcal{E}$ is already suffix closed, i.e. it contains all suffixes of its elements.

*Remark 9.* Assumption 6 of $P$ being deterministic is not far from being necessary to solve Problem 7, in the following sense. Suppose for simplicity that the desired accuracy $\theta$ is set to 0. Suppose now that $P$ is nondeterministic, i.e. there exist $(u, x_1), (u, x_2) \in P$ with $x_1(0) = x_2(0)$ and $x_1 \neq x_2$. Suppose also that $(u, x_1) \in \mathcal{E}$ and $(u, x_2) \notin \mathcal{E}$ and that $x_1 \in Q$ and $x_2 \notin Q$. On the basis of the collection of experiments $\mathcal{E}$, one would consider $u$ as a control input enforcing the specification because indeed $(u, x_1) \in \mathcal{E}$ and $x_1 \in Q$. On the other hand, since $P$ is nondeterministic, it can evolve starting from initial state $x_1(0) = x_2(0)$ and with control input $u$, either with state evolution $x_1$ which is in $Q$ or with state evolution $x_2$ which is not only not in $Q$ but also unknown (recall $(u, x_2) \notin \mathcal{E}$). Hence, in a nondeterministic setting, it is not possible in general to use a finite collection of experiments to design controllers enforcing regular language specifications.

For later purposes, we give the following

*Definition 10.* Let $Q(C, \mathcal{R}_0)$ be the set collecting all words $q_0\, q_1\, ... q_T \in Q$ for which there exists a pair

$$(u,x) \in P^C \cap (\mathcal{U}^{[0;T-1]} \times \mathcal{X}^{[0;T]}),$$

for some $(0,T) \in \mathcal{T}$, with $(x(0), x_{c,0}) \in \mathcal{R}_0$, where $x_{c,0}$ is the initial state of a state run (9) of $C$ marking $u$ such that (10) holds.

By the definition above, the set $Q(C, \mathcal{R}_0)$ is the part of the regular language specification $Q$ that is enforced by $C$ and $\mathcal{R}_0$ on $P$. Then, the following holds:

*Lemma 11.*

- If $C \sqsubseteq C'$ then $Q(C, \mathcal{R}_0) \subseteq Q(C', \mathcal{R}_0)$;
- If $\mathcal{R}_0 \subseteq \mathcal{R}'_0$ then $Q(C, \mathcal{R}_0) \subseteq Q(C, \mathcal{R}'_0)$.

## 4. MAIN RESULT

In this section we provide the solution to Problem 7. To this purpose, we first need to reformulate the specification $Q$ in terms of transition systems, as in Definition 1. Since $Q$ is a regular language, there exists a symbolic transition system

$$S'_Q = (X'_Q, X'_{0,Q}, \mathcal{X}, \xrightarrow[\prime,Q]{}, X'_{Q,m}, Y'_Q, H'_Q),$$

such that its marked input language coincides with $Q$, i.e.,

$$\mathcal{L}^u_m(S'_Q) = Q.$$

Note that in the definition above the output function can be chosen arbitrarily since it plays no role in ensuring $\mathcal{L}^u_m(S'_Q) = Q$. Without loss of generality, $S'_Q$ can be chosen as deterministic, accessible and nonblocking, see e.g. Cassandras and Lafortune (1999). Construction of $S'_Q$ can be done by resorting to standard algorithms available in the literature, see e.g. Lawson (2004). Automatic tools for constructing $S'_Q$ are also well known, see e.g. Caugherty (1990). For later purposes, it is useful to define the dual symbolic transition system $S_Q$ of system $S'_Q$, where states of $S_Q$ are transitions of $S'_Q$ and vice versa:

*Definition 12.* Pola et al. (2018) Given transition system $S'_Q$, define the dual transition system

$$S_Q = (X_Q, X_{Q,0}, U_Q, \xrightarrow[Q]{}, X_{Q,m}, \mathbb{X}, H_Q) \tag{11}$$

where:

- $X_Q$ coincides with the set $\xrightarrow[\prime,Q]{}$ of transitions of $S'_Q$;
- $X_{Q,0}$ is the collection of states $x'_Q \xrightarrow[\prime,Q]{u'_Q} x'^{,+}_Q$ in $X_Q$ with $x'_Q \in X'_{Q,0}$;
- $U_Q = \{u_Q\}$, where $u_Q$ is a dummy input;
- $\xrightarrow[Q]{}$ is the collection of transitions
$$\left(x^1_Q \xrightarrow[\prime,Q]{u'_Q} x^2_Q\right) \xrightarrow[Q]{u_Q} \left(x^3_Q \xrightarrow[\prime,Q]{u''_Q} x^4_Q\right)$$
with $x^2_Q = x^3_Q$;
- $X_{Q,m}$ is the collection of states $x'_Q \xrightarrow[\prime,Q]{u'_Q} x'^{,+}_Q$ in $X_Q$ with $x'^{,+}_Q \in X'_{Q,m}$;
- $H_Q(x'_Q \xrightarrow[\prime,Q]{u'_Q} x'^{,+}_Q) = u'_Q$ for any state $x'_Q \xrightarrow[\prime,Q]{u'_Q} x'^{,+}_Q$ in $X_Q$.

When specialized to Finite State Automata (FSA), the construction above coincides with the one of dual FSA in Gol et al. (2014). From the definitions above, it follows that

$$\mathcal{L}^y(S_Q) = \mathcal{L}^u(S'_Q) \setminus \{\varepsilon\} \text{ and } \mathcal{L}^y_m(S_Q) = \mathcal{L}^u_m(S'_Q) \setminus \{\varepsilon\}.$$

Moreover, $S_Q$ is symbolic, accessible and nonblocking. Since $\mathbb{X} \subseteq \mathcal{X}$, transition system $S_Q$ is metric with metric (7). For ease of notation, we denote a state $x'_Q \xrightarrow[\prime,Q]{u'_Q} x'^{,+}_Q$ of $X_Q$ by $x_Q$ and a transition $x_Q \xrightarrow[Q]{u_Q} x^+_Q$ of $S_Q$ by $x_Q \xrightarrow[Q]{} x^+_Q$.

Next step consists in encoding the experiments in $\mathcal{E}$ in the transition system

$$S(\mathcal{E}) = (X_e, X_{e,0}, U_e, \xrightarrow[e]{}, X_{e,m}, Y_e, H_e), \tag{12}$$

where:

- $X_e$ is the collection of states $z \in \mathcal{X}$ for which there exists a pair $(u,x) \in \mathcal{E}$ and $t \in \mathbb{N}$ such that $z = x(t)$;
- $X_{e,0} = X_e \cap \mathcal{X}_0$;
- $U_e = \mathcal{U}$;
- $z \xrightarrow[e]{v} z^+$, if there exists $(u,x) \in \mathcal{E}$ and $t \in \mathbb{N}$ such that $z = x(t)$, $v = u(t)$ and $z^+ = x(t+1)$;
- $X_{e,m} = X_e$;
- $Y_e = \mathcal{X}$;
- $H_e(x) = x$ for any $x \in X_e$.

The following statement highlights some properties of transition system $S(\mathcal{E})$.

*Proposition 13.* Transition system $S(\mathcal{E})$ is:
i) deterministic;
ii) symbolic;
iii) metric, with metric (7) for any $x, x' \in X_e$.

Moreover,

*Proposition 14.* Let $\mathcal{E}_1, \mathcal{E}_2 \subseteq P$. If $\mathcal{E}_1 \subseteq \mathcal{E}_2$, then $S(\mathcal{E}_1)$ is a sub–transition system of $S(\mathcal{E}_2)$, i.e. $S(\mathcal{E}_1) \sqsubseteq S(\mathcal{E}_2)$.

In order to solve Problem 7, we need to select transitions of $S(\mathcal{E})$ that match transitions of the specification $Q$ up to accuracy $\theta$. To this purpose, we define the following transition system:

$$C' = (X'_c, X'_{c,0}, U'_c, \underset{c,'}{\longrightarrow}, X'_{c,m}, \mathcal{X}, H'_c), \tag{13}$$

where:

- $X'_c$ is the collection of pairs $(x_e, x_Q) \in X_e \times X_Q$ such that
$$\mathbf{d}(H_e(x_e), H_Q(x_Q)) \leq \theta,$$
where we recall that $\theta$ is the desired accuracy in Problem 7;
- $X'_{c,0} = X'_c \cap (X_{e,0} \times X_{Q,0})$;
- $U'_c$ is the collection of input values $v \in \mathcal{U}$ for which there exists $(u, x) \in \mathcal{E}$ and $t \in \mathbb{N}$ such that $v = u(t)$;
- $(x_e, x_Q) \underset{c,'}{\overset{u}{\longrightarrow}} (x_e^+, x_Q^+)$ if $x_e \underset{e}{\overset{u}{\longrightarrow}} x_e^+$ and $x_Q \underset{Q}{\longrightarrow} x_Q^+$;
- $X_{c,m} = X_e \times X_{Q,m}$;
- $H'_c(x_e, x_Q) = H_Q(x_Q)$ for any $(x_e, x_Q) \in X'_c$.

Transition system $C'$ can be viewed as the product composition of $S(\mathcal{E})$ and $S_Q$ in an approximating sense. A similar notion of approximate product composition appeared previously in Tabuada (2008). In the sequel, we may write

$$S(\mathcal{E}) \times_\theta S_Q,$$

instead of $C'$ to emphasize the dependence of $C'$ on $S(\mathcal{E})$, $S_Q$ and $\theta$.

Transition system $C'$ is blocking in general. Since the controllers in $P^C$ are required to fulfill condition (10), we need to extract from $C'$, a sub–transition system exhibiting nonblocking behavior. This is accomplished by computing the transition system

$$C = \mathrm{Trim}(C'), \tag{14}$$

later on specified by the tuple

$$(X_c, X_{c,0}, U_c, \underset{c}{\longrightarrow}, X_{c,m}, \mathcal{X}, H_c),$$

which is indeed nonblocking.

Since $X_c \subseteq X_e \times X_Q$ and sets $X_e$ and $X_Q$ are finite, then $X_c$ is a finite set. Since $U_c \subseteq U'_c$ and $U'_c$ is finite, then $U_c$ is a finite set. As a consequence, controller $C$ is finite.

We now have all the ingredients to present the main result of this paper.

*Theorem 15.* Controller $C$ in (14) and relation $\mathcal{R}_0$ defined as

$$\mathcal{R}_0 = \{(x, (x_e, x_Q)) \in \mathcal{X}_0 \times X_{c,0} | x = x_e\}, \tag{15}$$

solve Problem 7.

A direct consequence of the result above is that the part of the specification that can be enforced by controller $C$ and the relation of initial states $\mathcal{R}_0$ coincides with the output marked language of $C$:

*Corollary 16.* $Q(C, \mathcal{R}_0) = \mathcal{L}_m^y(C)$.

## 5. MAXIMALITY, CONVERGENCE AND ADAPTIVITY OF THE SOLUTION

In this section, we discuss maximality of the solution of Problem 7, convergence of the solution as the set of experiments decreases, and adaptivity of the proposed controller. The controllers in (13) and (14) will be denoted respectively by $C'(\mathcal{E}, \theta)$ and $C(\mathcal{E}, \theta)$ to point out their dependence on the set of experiments $\mathcal{E}$ and the accuracy $\theta$. Similarly, $\mathcal{R}_0(\mathcal{E}, \theta)$ will denote the relation of initial states in (15), where the dependence on $\mathcal{E}$ and $\theta$ is specified.

We first show that the controller and relation of initial states solving Problem 7 given in Theorem 15 enforce the largest possible part of the specification on the controlled plant.

*Proposition 17.* For any controller $C''$ and relation $\mathcal{R}''_0$ solving Problem 7 with $C = C''$ and $\mathcal{R}_0 = \mathcal{R}''_0$, we have:

$$Q(C'', \mathcal{R}''_0) \subseteq Q(C(\mathcal{E}, \theta), \mathcal{R}_0(\mathcal{E}, \theta)).$$

The next result shows monotonicity of our solution to Problem 7 with respect to increasing accuracies.

*Proposition 18.* Let $\theta_1, \theta_2 \in \mathbb{R}_0^+$ be a pair of accuracies and let the finite set of experiments $\mathcal{E} \subseteq P$ be given. If $\theta_1 \leq \theta_2$ then

$$C(\mathcal{E}, \theta_1) \sqsubseteq C(\mathcal{E}, \theta_2); \tag{16}$$

$$\mathcal{R}_0(\mathcal{E}, \theta_1) \subseteq \mathcal{R}_0(\mathcal{E}, \theta_2); \tag{17}$$

$$Q(C(\mathcal{E}, \theta_1), \mathcal{R}_0(\mathcal{E}, \theta_1)) \subseteq Q(C(\mathcal{E}, \theta_2), \mathcal{R}_0(\mathcal{E}, \theta_2)). \tag{18}$$

The intuition behind the result above is that as accuracy parameter increases, our solution to Problem 7 may find more transitions of $S(\mathcal{E})$ that match transitions of $S_Q$.

The next result shows monotonicity of our solution to Problem 7 with respect to increasing sets of experiments.

*Proposition 19.* Let $\mathcal{E}_1 \subseteq P$ and $\mathcal{E}_2 \subseteq P$ be a pair of finite collections of experiments on $P$ and let the accuracy $\theta \in \mathbb{R}_0^+$ be given. If $\mathcal{E}_1 \subseteq \mathcal{E}_2$ then

$$C(\mathcal{E}_1, \theta) \sqsubseteq C(\mathcal{E}_2, \theta); \tag{19}$$

$$\mathcal{R}_0(\mathcal{E}_1, \theta) \subseteq \mathcal{R}_0(\mathcal{E}_2, \theta); \tag{20}$$

$$Q(C(\mathcal{E}_1, \theta), \mathcal{R}_0(\mathcal{E}_1, \theta)) \subseteq Q(C(\mathcal{E}_2, \theta), \mathcal{R}_0(\mathcal{E}_2, \theta)). \tag{21}$$

The intuition behind the result above is that as the set of experiments increases (in the sense of sets inclusion), our solution to Problem 7 may find more transitions of $P$ that match transitions of $S_Q$.

The following result establishes convergence properties of our solution to Problem 7.

*Proposition 20.* Consider a sequence $\mathcal{E}_{seq} = \{\mathcal{E}_i\}_{i \in \mathbb{N}}$ of finite sets of experiments on $P$ and suppose that

$$\mathcal{E}_i \subseteq \mathcal{E}_{i+1} \subseteq P,$$

for any $i \in \mathbb{N}$. Then, there exists $i(\mathcal{E}_{seq}) \in \mathbb{N}$ such that for any $i \geq i(\mathcal{E}_{seq})$:

$$Q(C(\mathcal{E}_i, \theta), \mathcal{R}_0(\mathcal{E}_i, \theta)) = Q(C(\mathcal{E}_{i(\mathcal{E}_{seq})}, \theta), \mathcal{R}_0(\mathcal{E}_{i(\mathcal{E}_{seq})}, \theta)). \tag{22}$$

Intuitively, the result above shows that as the set of experiments gets bigger, there is a step $i(\mathcal{E}_{seq})$ after which the corresponding part of the specification $Q$ enforced cannot "increase" anymore (in the sense of inclusion in

(21)). This is a consequence of the fact that regular languages specifications can be encoded by transition systems that are symbolic, i.e. their sets of states and inputs have finite cardinality.

We then obtain the following result:

*Corollary 21.* Consider two sequences $\mathcal{E}_{seq} = \{\mathcal{E}_i\}_{i\in\mathbb{N}}$ and $\mathcal{E}'_{seq} = \{\mathcal{E}'_i\}_{i\in\mathbb{N}}$ of sets of experiments on $P$ and suppose that

$$\mathcal{E}_i \subseteq \mathcal{E}_{i+1} \subseteq P, \quad \mathcal{E}'_i \subseteq \mathcal{E}'_{i+1} \subseteq P, \quad \forall i \in \mathbb{N} \tag{23}$$

and that

$$\bigcup_{i\in\mathbb{N}} \mathcal{E}_i = P, \quad \bigcup_{i\in\mathbb{N}} \mathcal{E}'_i = P. \tag{24}$$

Then,

$$Q(C(\mathcal{E}_{i(\mathcal{E}_{seq})},\theta), \mathcal{R}_0(\mathcal{E}_{i(\mathcal{E}_{seq})},\theta)) = Q(C(\mathcal{E}'_{i(\mathcal{E}'_{seq})},\theta), \mathcal{R}_0(\mathcal{E}'_{i(\mathcal{E}'_{seq})},\theta)).$$

This result shows that independently of the sets of experiments $\mathcal{E}_{seq}$ and $\mathcal{E}'_{seq}$ and provided that these sets satisfy conditions (23) and (24), the parts of the specification $Q$ that can be enforced by the corresponding solutions to Problem 7 coincide asymptotically.

We conclude this section by showing the adaptivity of the solution to Problem 7 with respect to increasing sequences of experiments.

*Proposition 22.* $C(\mathcal{E}_1 \cup \mathcal{E}_2, \theta) = \mathrm{Trim}(C'(\mathcal{E}_1,\theta) \sqcup C'(\mathcal{E}_2,\theta))$.

The result above is significant from a computational point of view since it shows that, once the solution to Problem 7 has been found with respect to a set of experiments $\mathcal{E}_1$, if another set of experiments $\mathcal{E}_2$ is collected, the solution to Problem 7 with respect to the set of experiments $\mathcal{E}_1 \cup \mathcal{E}_2$ does not need to be recomputed but can be obtained by combining the solution already available for the set $\mathcal{E}_1$ with the solution to be determined for the set $\mathcal{E}_2$.

*Remark 23.* One could wonder whether the result above can be strengthened to the following equality

$$C(\mathcal{E}_1 \cup \mathcal{E}_2, \theta) = C(\mathcal{E}_1,\theta) \sqcup C(\mathcal{E}_2,\theta).$$

The equality above is not true because, for two transition systems $S_1$ and $S_2$, while inclusion $\mathrm{Trim}(S_1) \sqcup \mathrm{Trim}(S_2) \sqsubseteq \mathrm{Trim}(S_1 \sqcup S_2)$ is true, equality $\mathrm{Trim}(S_1) \sqcup \mathrm{Trim}(S_2) = \mathrm{Trim}(S_1 \sqcup S_2)$ is not true, in general.

## 6. CONCLUSIONS

In this paper we addressed data–driven control design of an abstract discrete–time control system with specifications expressed in terms of regular languages. We first derived the solution to the control problem. Then we addressed maximality of the controller, convergence of the controller as the number of experiments increases and adaptive–type control design. In future work we plan to design efficient algorithms for the synthesis of the proposed controllers.

## ACKNOWLEDGEMENTS

We thank Carlos Canudas-de-Wit and Claudio De Persis for fruitful discussions on data–driven control design.

## REFERENCES

Bazanella, A.S., Campestrini, L., and Eckhard, D. (2011). *Data-driven controller design: the $H_2$ approach.* Springer Science & Business Media.

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

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

Caugherty, D. (1990). JFLAP: An interactive formal languages and automata package. Available online at http://www.jflap.org/.

De Persis, C. and Tesi, P. (2019). Formulas for data-driven control: Stabilization, optimality and robustness. *ArXiv*, abs/1903.06842.

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

Hou, Z.S. and Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235, 3–35.

Lawson, M. (2004). *Finite Automata.* CRC Press.

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 supervisory control of networks of nonlinear control systems. *IEEE Transactions on Automatic Control*, 63(9), 2803–2817.

Ruberti, A. and Isidori, A. (1979). *Teoria dei sistemi.* Bollati Boringhieri.

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.