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

Regulation Control in Interpreted Petri Nets Under Partial Observation ⋆

Italia Jiménez-Ochoa * **Daniel Guevara-Lozano** *
C. Renato Vázquez ** **Antonio Ramírez-Treviño** *

* *CINVESTAV, Av. del Bosque 1145, 45019, Zapopan, Jalisco, México (e-mail: {mjimenez, dguevara, art}@gdl.cinvestav.mx)*
** *Tecnologico de Monterrey, Av. Ramón Corona 2514, 45201, Zapopan, Jalisco, México (e-mail: cr.vazquez@tec.mx)*

Abstract: This paper addresses the regulation control problem for discrete event systems (DES) under partial information. In this approach, the system to be controlled, named the plant, and the required behavior, named the specification, are both represented as Petri nets (PNs) with input and output symbols. The goal is to synthesize a controller that indicates input symbols to the plant in order to reach a state where the output is equal to that of the specification. To achieve this goal, the only information available to the controller is the plant output, i.e., the controller does not know the exact state of the plant. In this work, a control methodology is proposed to synthesize regulation controllers under this setting.

Keywords: Petri nets, Event-based control, Supervisory-control.

1. INTRODUCTION

The control of discrete event systems (DES) has been widely studied in the last decades (Wonham et al. (2018); Giua and Silva (2018)). The most studied control approach for DES is the Supervisory Control theory (Ramadge and Wonham (1982, 1987); Moody and Antsaklis (1998)), initially proposed for finite state automata and later extended to Petri nets (PNs), in which the specification is a language that describes all the event-sequences that are allowed in the closed-loop system. In this framework, the supervisor disables key events in order to constrain the system behaviour into the specification language. The Generalized Mutual Exclusions approach (Giua et al. (1992); Basile et al. (2013)) is another well-known control strategy designed for PNs, in which bounds on the weighted sum of tokens of particular places are imposed to avoid unsafe or deadlock states. This technique has been extensively studied for liveness-enforcing of particular classes of PNs (Chen et al. (2011); Li et al. (2012)).

The regulation control approach was introduced in Santoyo et al. (2001); Sánchez-Blanco et al. (2004) in order to develop efficient and intuitive controllers for DES, in which the goal is to drive the system to states where some required output sensors are activated. The regulation framework has been recently revisited for its application in industrial automation systems by Vázquez et al. (2018); Guevara-Lozano et al. (2019). In this framework, the system to be controlled, named the *plant*, is represented by a PN where input symbols are associated to transitions and output symbols are associated to places, representing actuators and sensors, respectively. Similarly, the required behavior is also represented as a PN, named *specification*, where the input symbols describe guards and the output symbols describe the output signals required from the plant. The regulation control problem consists in the synthesis of an external agent (controller) that provides input symbols to the plant to reach a state where the plant output is equal to the specification output.

The control regulation problem is related to the Supervisory Control. However, there are relevant differences. In the regulation framework, the user does not specify neither a required language nor required states. Instead, the user specifies sequences of required input/output symbols as a PN (i.e., activation of sensors/actuators). The controller synthesis does involve the computation of particular states and event-sequences to fulfill the specification, but this process can be fully automatized. Moreover, rather than enclosing the plant behavior, the regulation control enforces the computed event-sequences. The advantage of this approach is that the user does not require neither a deep knowledge of the plant nor a training on DES theory to describe a specification. Moreover, the practical complexity can be dramatically reduced.

In this work, it is assumed that the activation of the plant output symbols is the only information available to the controller, then there exist several possible plant markings for the same sensor signals combination. The contribution of this work is to propose an approach to deal with this issue, in this, one controller is designed for each possible initial marking, and a dynamic mechanism is proposed in order to select a controller that drives the plant to a convenient marking without enabling unplanned sequences.

⋆ This work has been supported by program SNI Conacyt. The research leading to these results has received support from the Conacyt Fondo Sectorial de Investigación para la Educación, project number 288470.

This paper is organized as follows. Basic definitions are provided in Section 2. Section 3 describes the regulation control problem. The synthesis of a controller is introduced in Section 4 for the case in which the plant state is available to the controller. The case in which only the plant output is available is addressed in Section 5. Finally, some conclusions and future work are provided in Section 6.

2. DEFINITIONS

In the sequel, the i-th column (resp. row) of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}(\bullet, i)$ (resp. $\mathbf{A}(i, \bullet)$). Given sets of indexes S_r and S_c, $\mathbf{A}(S_r, S_c)$ denotes the sub-matrix built with the entries of $\mathbf{A}$ at the rows and columns indicated by S_r and S_c, respectively.

Definition 2.1. A PN structure is a bipartite digraph represented by the 4-tuple $G = \langle P, T, I, O\rangle$, where $P = \{p_1, p_2, ..., p_n\}$ is the finite set of places, $T = \{t_1, t_2, ..., t_m\}$ is the finite set of transitions, $I : P \times T \to \mathbb{Z}^+$ is a function representing the weighted arcs going from places to transitions, and $O : T \times P \to \mathbb{Z}^+$ is a function representing the weighted arcs going from transitions to places, $\mathbb{Z}^+$ is the set of nonnegative integers. Graphically, places are represented by circles, transitions by rectangles and arcs by arrows. The *incidence matrix* of G is defined as $\mathbf{C} = [c_{ij}]$, where $c_{ij} = O(t_j, p_i) - I(p_i, t_j)$. See Silva (1993); David and Alla (2010) for further information about PNs.

As usual, ${}^\bullet t_j$ (resp. ${}^\bullet p_i$) denotes the set of all places p_i (resp. transitions t_j) such that $I(p_i, t_j) \neq 0$ (resp. $O(p_i, t_j) \neq 0$). Similarly, $t_j^\bullet$ (resp. $p_i^\bullet$) denotes the set of all places p_i (resp. transitions t_j) such that $O(p_i, t_j) \neq 0$ (resp. $I(p_i, t_j) \neq 0$). A PN is a *state machine* if each transition has only one input and one output place, i.e., $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$ and the Petri net graph is strongly connected. The *marking* function $M : P \to \mathbb{Z}^+$ is a mapping representing the number of tokens residing into each place. The marking of a PN is expressed as an n-entry column vector $\mathbf{M}$. Graphically, tokens are represented as dots inside places.

Definition 2.2. A PN system is the pair $\langle G, \mathbf{M}_0\rangle$, where G is a PN structure and $\mathbf{M}_0$ is an initial token distribution. A transition t_j is enabled at marking $\mathbf{M}_k$ iff $\forall p_i \in {}^\bullet t_j$, $\mathbf{M}_k(p_i) \geq I(p_i, t_j)$, this is denoted as $\mathbf{M}_k \xrightarrow{t_j}$. A transition t_j can fire (the corresponding event can occur) if it is enabled, in such case, the system reaches a new marking $\mathbf{M}_{k+1}$ that can be computed with the so-called PN *fundamental equation* $\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{C}\mathbf{v}_k$, where $\mathbf{v}_k(i) = 0$ for $i \neq j$ and $\mathbf{v}_k(j) = 1$. This is denoted as $\mathbf{M}_k \xrightarrow{t_j} \mathbf{M}_{k+1}$.

Definition 2.3. A firing sequence of a PN $\langle G, \mathbf{M}_0\rangle$ is a sequence of transitions $\sigma = t_i t_j ... t_k$ such that $\mathbf{M}_0 \xrightarrow{t_i} \mathbf{M}_1 \xrightarrow{t_j} ... \mathbf{M}_w \xrightarrow{t_k}$. The Parikh vector $\boldsymbol{\sigma}$ of a firing sequence σ is defined as a column vector of length $|T|$ such that $\boldsymbol{\sigma}(j) = k$ if t_j is fired k times in the sequence σ. In this way, the marking $\mathbf{M}'$ reached after the firing of σ from a marking $\mathbf{M}$ can be computed with the fundamental equation

$$\mathbf{M}' = \mathbf{M} + \mathbf{C}\boldsymbol{\sigma}$$

This is denoted as $\mathbf{M} \xrightarrow{\sigma} \mathbf{M}'$, σ is said to be *fireable* from $\mathbf{M}$ and $\mathbf{M}'$ is said to be *reachable* from $\mathbf{M}$.

The reachability set of a PN is the set of all the reachable markings from $\mathbf{M}_0$, and it is denoted as $R(G, \mathbf{M}_0)$. A PN system is said *safe* if $\forall \mathbf{M} \in R(G, \mathbf{M}_0)$ and $\forall p \in P$ it holds $\mathbf{M}(p) \leq 1$. A vector $\mathbf{b} \neq \mathbf{0}$ such that $\mathbf{b}^T\mathbf{C} = \mathbf{0}$ is said to be a *P-flow*. The matrix $\mathbf{B}_y$ denotes a basis for the *P-flows* of the net. Nonnegative P-flows are called *P-semiflows*. Every reachable marking $\mathbf{M} \in R(G, \mathbf{M}_0)$ satisfies $\mathbf{b}^T\mathbf{M} = \mathbf{b}^T\mathbf{M}_0$. In this work, it will be assumed that all the P-flows are initially marked, thus $\mathbf{b}^T\mathbf{M}_0 > 0$.

Definition 2.4. (Ramírez-Trevino et al. (2003)). An *Interpreted Petri Net* (IPN) system is a 5-tuple $Q = \langle G, \mathbf{M}_0, \Sigma_I, \lambda, \boldsymbol{\varphi}\rangle$ where:

- $\langle G, \mathbf{M_0}\rangle$ is a PN system;
- Σ_I is the input alphabet of the PN system, where each element of the set Σ_I is an input symbol;
- $\lambda : T \to \Sigma_I \cup \{\epsilon\}$ is the labeling function of transitions with the restriction that nondeterministic inputs are not allowed, i.e., $\forall t_j, t_k \in T, j \neq k$, if $I(p_i, t_j) = I(p_i, t_k) \neq 0$ and both $\lambda(t_j), \lambda(t_k) \neq \epsilon$, then $\lambda(t_j) \neq \lambda(t_k)$. Here, ϵ represents a system's internal event;
- $\boldsymbol{\varphi}$ is an $q \times n$ matrix whose entries belong to $\{0, 1\}$, where q is the number of outputs and n is the number of places. The IPN output vector at marking $\mathbf{M}_k$ is defined as $\mathbf{y}_k = \boldsymbol{\varphi} \cdot \mathbf{M}_k$.

In this work it is assumed that each place is associated to at most one distinct output, i.e. $\varphi(i, \bullet)$ and $\varphi(\bullet, i)$ are elementary vectors. If $\lambda(t_j) \neq \epsilon$, the transition t_j is said to be controllable. Otherwise, it is uncontrollable. The set of uncontrollable transitions is denoted as T_{un}. A place $p_j \in P$ is said to be measurable if the $j-th$ column of φ is not null. i.e. $\varphi(\bullet, j) \neq \mathbf{0}$. Otherwise, it is nonmeasurable.

The evolution of an IPN is similar to that of the PN system, where the following aspects are also considered for the transitions firing.

- The input symbols are said to be *indicated* when an external controller requires the firing of the corresponding transitions. If $\lambda(t_j) = a_i \neq \epsilon$ is indicated and t_j is enabled then t_j *must* fire. If t_j is enabled by the marking, but the symbol $\lambda(t_j)$ is not indicated, then t_j cannot fire. If $\lambda(t_j) = \epsilon$ and t_j is enabled, then t_j *can* fire at any moment.
- At any reachable marking $\mathbf{M}_k$, an external observer reads the output vector $\mathbf{y}_k = \boldsymbol{\varphi} \cdot \mathbf{M}_k$.

Definition 2.5. An IPN Q is event-detectable if the firing of any transition $t_i \in T$ can be detected and distinguished from the knowledge of the indicated input symbols and a change in the output vector, i.e., $\forall\, t_i \in T$

- $\boldsymbol{\varphi} C(\bullet, i) \neq \mathbf{0}$
- $\forall t_j \in T \setminus \{t_i\}$, $\lambda(t_i) \neq \lambda(t_j)$ or $\boldsymbol{\varphi}\mathbf{C}(\bullet, i) \neq \boldsymbol{\varphi}\mathbf{C}(\bullet, j)$.

3. REGULATION FRAMEWORK

Following the Control Theory terminology, the IPN system to be controlled is named *Plant*, and the desired behavior of the plant is described by an IPN system named *Specification*.

Definition 3.1. The plant is a safe event-detectable IPN $\langle Q_p, \mathbf{M}_0\rangle = \langle G, \mathbf{M}_0, \Sigma_I, \lambda, \boldsymbol{\varphi}\rangle$ that models the discrete event system to be controlled.

Definition 3.2. A specification model $\langle Q_s, \breve{\mathbf{M}}_0\rangle = \langle \breve{G}, \breve{\mathbf{M}}_0, \breve{\Sigma}_I, \breve{\lambda}, \breve{\boldsymbol{\varphi}}\rangle$ is a safe state machine IPN, where $\breve{\lambda}$ is a bijection (each transition is associated to a different symbol), and the system and the specification models have the same output vector dimension.

The regulation problem has been addressed in Santoyo et al. (2001); Sánchez-Blanco et al. (2004); Guevara-Lozano et al. (2019) for the case in which the plant marking is known. The problem can be formulated as the synthesis of a controller function H that provides the input symbols that have to be indicated to the plant in order to produce an output equal to that of the specification.

Definition 3.3. Let $\langle Q_p, \mathbf{M}_0\rangle$ and $\langle Q_s, \breve{\mathbf{M}}_0\rangle$ be the IPN models of the plant and the specification, respectively. The regulation problem under complete information (i.e., assuming the plant marking is always known) consists in the computation of a *controller* function

$$H : R(Q_p, \mathbf{M}_0) \times R(Q_s, \breve{\mathbf{M}}_0) \times \breve{T} \to \Sigma_I,$$

such that $\forall \mathbf{M}_i \in R(Q_p, \mathbf{M}_0)$, $\forall \breve{\mathbf{M}}_j \in R(Q_s, \breve{\mathbf{M}}_0)$, the indication of the input symbols $H(\mathbf{M}_i, \breve{\mathbf{M}}_j, \breve{t}_k)$, where $\breve{t}_k$ is the specification transition previously fired, will eventually lead the plant to a marking $\mathbf{M}_j$ such that $\boldsymbol{\varphi}\mathbf{M}_j = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_j$.

In this work, the regulation problem under partial information is considered. In this case, instead of the plant marking, the controller will use the plant output to compute and provide appropriate control actions. For this, the control problem is split in two stages. In the first one, the plant is driven from its initial marking, which is assumed to belong to the set of markings that agree with the initial observation $\mu_0 = \{\mathbf{M}_i \in \{0,1\}^{|P|} | \boldsymbol{\varphi}\mathbf{M}_i = \mathbf{y}_0, \mathbf{B}_y^T\mathbf{M}_i = \mathbf{1}\}$, to a particular marking $\mathbf{M}_1$ such that $\boldsymbol{\varphi}\mathbf{M}_1 = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_1$, where $\breve{\mathbf{M}}_1$ is the specification marking that can be reached after the first firing in the specification. During this stage, it is required to update each possible marking from μ_0 according to the detected firings in the plant, otherwise stated, when a firing sequence σ_k is detected a set μ_k of possible current markings is computed as $\mu_k = \{\mathbf{M}_k^i | \mathbf{M}_0^i \xrightarrow{\sigma_k} \mathbf{M}_k^i, \mathbf{M}_0^i \in \mu_0\}$.

Definition 3.4. Let $\langle Q_s, \breve{\mathbf{M}}_0\rangle$ be the IPN specification and let Q_p be the IPN plant structure. Assume $\mathbf{M}_0 \in \mu_0$. The first stage of the regulation problem under partial information consists in the computation of a collection of *controller* functions $F_k : \mu_k \to \Sigma_I \qquad$ for $k \in \{0, 1, 2, ...\}$, where F_k represents the control actions that can be indicated after the k-th firing in the plant in order to drive the plant to a particular known marking $\mathbf{M}_1$, where $\boldsymbol{\varphi}\mathbf{M}_1 = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_1$.

Under the assumption that the plant is safe, event-detectable and deterministic, it holds that $|\mu_0| \geq |\mu_1| \geq ... \geq |\mu_k|$, thus, if the controller is non-blocking, there will be some step r such that $|\mu_r| = 1$, since the controller will lead all the possible markings to $\mathbf{M}_1$. Safeness and event-detectability also implies that once the marking is known, it can be uniquely determined in the future evolution based on the input-output information, thus, the second stage becomes equivalent to the regulation problem under complete information.

4. CONTROLLER DESIGN WITH COMPLETE INFORMATION

The design of a regulation controller under complete information was already advanced in Santoyo et al. (2001); Sánchez-Blanco et al. (2004); Guevara-Lozano et al. (2019). Let us recall some basic ideas. For the controller synthesis, three main steps are required: 1) each specification reachable marking $\breve{\mathbf{M}} \in R(Q_s, \breve{\mathbf{M}}_0)$ is associated to a plant reachable marking $\mathbf{M} \in R(Q_p, \mathbf{M}_0)$ such that $\boldsymbol{\varphi}\mathbf{M} = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}$, this association can be described by a linear function $\boldsymbol{\Pi} : R(Q_s, \breve{\mathbf{M}}_0) \to R(Q_p, \mathbf{M}_0)$, defined such that $\boldsymbol{\Pi}\breve{\mathbf{M}} = \mathbf{M}$ (the computation of such function can be performed by an LPP (Santoyo et al. (2001); Sánchez-Blanco et al. (2004))); 2) each specification transition $\breve{t} \in \breve{T}$ is associated to a controllable plant firing sequence σ such that $\mathbf{M}_i \xrightarrow{\sigma} \mathbf{M}_j$, where $\breve{\mathbf{M}}_i \xrightarrow{\breve{t}} \breve{\mathbf{M}}_j$, $\boldsymbol{\Pi}\breve{\mathbf{M}}_i = \mathbf{M}_i$ and $\boldsymbol{\Pi}\breve{\mathbf{M}}_j = \mathbf{M}_j$ (the sequences can be computed by exploring the plant reachability graph, for instance by using the $A^\star$ algorithm (Santoyo et al. (2001); Sánchez-Blanco et al. (2004)); 3) based on the knowledge of the mapping $\boldsymbol{\Pi}$ and the controllable sequences σ, a controller can be synthesized as an agent that indicates suitable plant input symbols so the sequences σ are enforced when required by the specification. The first two steps are described in Santoyo et al. (2001); Sánchez-Blanco et al. (2004). The third step is implemented in the following algorithm in order to synthesize a controller function as defined above.

Algorithm 4.1. Calculation of H.

1: **Input** Reachability sets of the plant $R(Q_p, \mathbf{M}_0)$ and the specification $R(Q_s, \breve{\mathbf{M}}_0)$, specification's transitions set $\breve{T}$, function $\boldsymbol{\Pi}$, controllable firing sequences σ_a computed for each $\breve{t}_a \in T$.
2: **Output** Controller H, complement $\overline{H}$.
3: **for** all $\breve{\mathbf{M}}_i \in R(Q_s, \breve{\mathbf{M}}_0)$ **do**
4: **for** all $\breve{t}_a \in T$ **do**
5: **if** $\exists \breve{\mathbf{M}}_j \in R(Q_s, \breve{\mathbf{M}}_0)$ such that $\breve{\mathbf{M}}_i \xrightarrow{\breve{t}_a} \breve{\mathbf{M}}_j$ **then**
6: Denote as $\sigma_a = t_{a1}...t_{am}$ the firing sequence such that $\boldsymbol{\Pi}\breve{\mathbf{M}}_i \xrightarrow{t_{a1}} \mathbf{M}_{a2} \xrightarrow{t_{a2}} ... \xrightarrow{t_{am}} \boldsymbol{\Pi}\breve{\mathbf{M}}_j$
7: Define the controller H:

$$\begin{aligned} &H(\boldsymbol{\Pi}\breve{\mathbf{M}}_i, \breve{\mathbf{M}}_j, \breve{t}_a) = \lambda(t_{a1}) \\ &H(\mathbf{M}_{a2}, \breve{\mathbf{M}}_j, \breve{t}_a) = \lambda(t_{a2}) \\ &\vdots \\ &H(\mathbf{M}_{am}, \breve{\mathbf{M}}_j, \breve{t}_a) = \lambda(t_{am}) \end{aligned}$$

8: Define the complement function $\overline{H}$:

$$\begin{aligned} &\overline{H}(\boldsymbol{\Pi}\breve{\mathbf{M}}_i, \breve{\mathbf{M}}_j, \breve{t}_a) = \\ &\quad \{\lambda(t) | t \in T_c \setminus \{t_{a1}\},\ \boldsymbol{\Pi}\breve{\mathbf{M}}_i \xrightarrow{t}\} \\ &\overline{H}(\mathbf{M}_{a2}, \breve{\mathbf{M}}_j, \breve{t}_a) = \\ &\quad \{\lambda(t) | t \in T_c \setminus \{t_{a2}\},\ \mathbf{M}_{a1} \xrightarrow{t}\} \\ &\vdots \\ &\overline{H}(\mathbf{M}_{am}, \breve{\mathbf{M}}_j, \breve{t}_a) = \\ &\quad \{\lambda(t) | t \in T_c \setminus \{t_{am}\},\ \mathbf{M}_{am} \xrightarrow{t}\} \end{aligned}$$

9: **end if**
10: **end for**

11: **end for**
12: For the rest of elements in the domain $(\mathbf{M}, \breve{\mathbf{M}}, \breve{t}) \in R(Q_p, \mathbf{M}_0) \times R(Q_s, \breve{\mathbf{M}}_0) \times \breve{T}$, set $H(\mathbf{M}, \breve{\mathbf{M}}, \breve{t}) = \epsilon$, $\overline{H}(\mathbf{M}, \breve{\mathbf{M}}, \breve{t}) = \emptyset$.

In this way, after the firing of a transition $\breve{t}_a$ in the specification, leading to a marking $\breve{\mathbf{M}}_j$, the controller H indicates input symbols in order to enforce the plant sequence σ_a, leading to $\mathbf{\Pi}\breve{\mathbf{M}}_j$. The controllability of σ_a (explained in Santoyo et al. (2001); Sánchez-Blanco et al. (2004)) ensures that it is the only sequence that can occur when the controller H indicates the corresponding symbols. Notice that if an uncontrollable transition t_k belongs to the sequence, the function H will correctly indicate the symbol $\lambda(t_k) = \epsilon$, meaning that the controller does not indicate a control action and it must wait for the firing of t_k.

The complement function $\overline{H}$ provides the symbols of controllable transitions that are enabled but not indicated. In the sequel, we will refer to those symbols as *disabled* by the controller H.

5. REGULATION UNDER PARTIAL INFORMATION

The regulation controller already introduced requires the knowledge of the plant's marking to provide suitable control actions. When such information is not available, a question that arises is whether the controller can properly work using the marking estimated by a marking observer, instead of the true marking of the plant that is not completely known. Nevertheless, from simple examples we can conclude that the IPN controller introduced above does not properly work under partial information, even when an observer is used (since the observer may not provide a suitable estimate from the initial marking).

The forthcoming analysis provides tools in order to design new controllers that operate under partial information. In this case, the initial marking of the plant is unknown, but it is assumed that the plant and the specification provide the same initial output. Thus, given an initial marking $\breve{\mathbf{M}}_0$ of the specification system that produces an output $\breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_0$, there might be several possible initial markings in the plant $\mathbf{M}_0^1$, $\mathbf{M}_0^2$, ..., $\mathbf{M}_0^i$ producing the same output, i.e., $\breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_0 = \boldsymbol{\varphi}\mathbf{M}_0^1 = \boldsymbol{\varphi}\mathbf{M}_0^2 = ... = \boldsymbol{\varphi}\mathbf{M}_0^i$. Consequently, the controller will not know which is the true marking of the plant. To overcome this problem, a controller is designed for each possible initial marking. In particular, if $\breve{t}_a$ is the only enabled transition at the specification initial marking leading the marking from $\breve{\mathbf{M}}_0$ to $\breve{\mathbf{M}}_1$, then a set of functions $\mathbf{\Pi}^1$, ... ,$\mathbf{\Pi}^i$ and their corresponding controllers H^1,...,H^i are designed such that, $\forall j \in \{1, 2, ..., i\}$, H^j drives the plant from $\mathbf{M}_0^j$ to a marking $\mathbf{M}_1$, with $\mathbf{M}_1 = \mathbf{\Pi}^j\breve{\mathbf{M}}_1$ and $\boldsymbol{\varphi}\mathbf{M}_1 = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_1$. In this way, the marking to be reached in the plant is the same for all the controllers. An algorithm is used to decide which of the control actions given by the controllers must be performed during the evolution of the plant.

Next, after reaching $\mathbf{M}_1$, the marking will be known. Moreover, safeness and even-detectability implies that once the marking is known, it can be uniquely determined in the future evolution based on the input-output information. Thus, the regulation problem under partial information is only required to be solved for the first firing of a specification transition.

5.1 Calculation of Possible Initial Markings

The Algorithm 5.1 is proposed to compute the possible initial markings in the plant that do not enable uncontrollable transitions and agree with both the observed output and the P-flows loads.

Algorithm 5.1. Calculation of possible initial markings.

1: **Input** Initial output of the system $\mathbf{y}_0$, $\mathbf{B}_y^T$, the set of uncontrollable transitions T_{uc}.
2: **Output** The set of possible initial markings μ_0.
3: Initialize $Restriction\_Set = \emptyset$
4: Initialize $\mu_0 = \emptyset$
5: Initialize $Flag = 1$, $k = 1$
6: **while** Flag = 1 **do**
7: Solve the following LPP:
8: $min \sum_{i=1}^{n} \mathbf{M}_k(p_i)$ subject to
9: $0 \leq \mathbf{M}_k(p_i) \leq 1$
10: $\boldsymbol{\varphi}\mathbf{M}_k = \mathbf{y}_0$
11: $\mathbf{B}_y^T\mathbf{M}_k = \mathbf{1}$
12: $\forall t_j \in T_{uc}: \quad \sum_{p_i \in {}^\bullet t_j} \mathbf{M}_k(p_i) \leq |{}^\bullet t_j| - 1$
13: $\forall r \in \{1, ..., k-1\}: \quad \mathbf{M}_r^T\mathbf{M}_k \leq s - 1$,
14: where s is the number of P-flows.
15: **if** LPP has solution **then**
16: $\mu_0 = \mu_0 \cup \{\mathbf{M}_k\}$
17: $k = k + 1$
18: **else**
19: Set Flag=0
20: **end if**
21: **end while**

Lines 10, 11 and 12 stand for ensuring that the potential initial marking meets the observed output, the P-semiflows loads and that no uncontrollable transition is enabled, respectively. Line 13 ensures that a different initial marking is computed each time the LPP is solved.

In the worst case, the number of possible initial markings would grow exponentially w.r.t. the number of minimal P-semiflows, hence their computation runs time-exponentially, and the memory required to store these marking would grow space-exponentially. Nonetheless, previous algorithm runs off-line, thus it will be not used during the on-line execution of the controller.

5.2 Calculation of partial controllers

The following algorithm computes the controllers for each possible initial marking. For this, let us denote as $\breve{t}_a$ the only enabled specification transition, i.e., $\breve{\mathbf{M}}_0 \xrightarrow{\breve{t}_a} \breve{\mathbf{M}}_1$. Moreover, consider a marking of the plant $\mathbf{M}_1$ such that

$\boldsymbol{\varphi}\mathbf{M}_1 = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_1$.

Algorithm 5.2. Calculation of controllers for μ_0.

1: **Input** IPN plant structure $(Q_p, \mathbf{M}_0)$ and μ_0. Target marking $\mathbf{M}_1$.
2: **Output** Controller H^i for each $\mathbf{M}_0^i \in \mu_0$.
3: **for** each $\mathbf{M}_0^i \in \mu_0$ **do**
4: Define the function $\boldsymbol{\Pi}^i$ as $\boldsymbol{\Pi}^i\breve{\mathbf{M}}_0 = \mathbf{M}_0^i$ and $\boldsymbol{\Pi}^i\breve{\mathbf{M}}_1 = \mathbf{M}_1$.
5: Compute a controllable sequence σ_a^i such that $\mathbf{M}_0^i \xrightarrow{\sigma_a^i} \mathbf{M}_1$, considering the constraint $\boldsymbol{\sigma}_a^i \geq \mathbf{1}$, where $\mathbf{1}$ is a column vector of dimension $|T|$ whose entries are 1's.
6: Compute the controller H^i and its complement $\overline{H}^i$ by using the Algorithm 4.1, but only for the specification marking $\breve{\mathbf{M}}_1$ in line 3 and the transition $\breve{t}_a$ in line 4.
7: **end for**

The computed sequences intentionally involve all the transitions, thus only one sequence can be fired to reach $\mathbf{M}_1$. The relation of the resulting functions $\boldsymbol{\Pi}^i$ and sequences σ_a^i is illustrated in fig. 1.

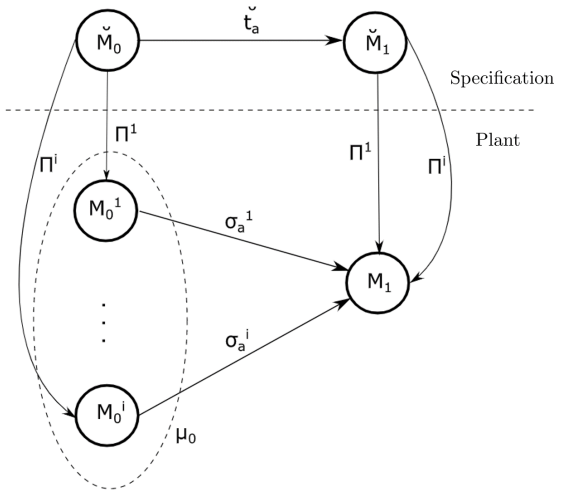

Fig. 1. Functions $\boldsymbol{\Pi}^i$ for the first stage control problem.

5.3 Common events matrix and control function F

Under partial information, the controller may not know the current marking, instead of that, a set of different possible current markings is calculated. Under this scenario, a symbol must not be indicated if it is disabled by a controller complement of a possible current marking. To facilitate this task, a matrix describing the indicated and disabled symbols for each possible current marking is computed, after each transition firing.

Let us firstly introduce some notation. Given a set of possible initial markings μ_0, when a firing sequence σ_k of length k is detected a set μ_k of updated markings is computed as $\mu_k = \{\mathbf{M}_k^i | \mathbf{M}_0^i \xrightarrow{\sigma_k} \mathbf{M}_k^i, \mathbf{M}_0^i \in \mu_0\}$. The detected firings will allow to discard possible initial markings. A set of indexes E_k will be used to record the current markings that are consistent with the observed input-output symbols, i.e., $i \in E_k$ iff $\mathbf{M}_k^i \in \mu_k$ is a possible current marking based on the observed input-output relation. In this way, the set of possible current markings is $\mu_k[E_k] = \{\mathbf{M}_k^i \in \mu_k | i \in E_k\}$.

Definition 5.1. Let $\langle Q_p, \mathbf{M}_0\rangle$ and $\langle Q_s, \breve{\mathbf{M}}_0\rangle$ be the IPN models of the plant and the specification, respectively. Let $\boldsymbol{\Pi}^i$ and H^i be the mapping function and the controller computed for an initial condition $\mathbf{M}_0^i \in \mu_0$, respectively. Let μ_k be the set of possible current markings at the plant. Let $\breve{\mathbf{M}}_1$ be the specification marking reached after the firing of $\breve{t}_a$, the only enabled transition from $\breve{\mathbf{M}}_0$. The matrix of common events is a square matrix of dimension $|\mu_k| \times |\mu_k|$, whose entries are defined as

$$\boldsymbol{\Omega}_{\mu_k}(i,j) = |\{H^i(\mathbf{M}_k^i, \breve{\mathbf{M}}_1, \breve{t}_a)\} \cap \overline{H}^j(\mathbf{M}_k^j, \breve{\mathbf{M}}_1, \breve{t}_a)|$$

If $\boldsymbol{\Omega}_{\mu_k}(i,j) > 0$ it means that the symbol to be enforced by the i-th controller must be avoided by the j-th controller complement. Thus, if it is not known whether the system is at $\mathbf{M}_k^i$ or at $\mathbf{M}_k^j$, the controller scheme should not indicate the symbol given by the i-th controller $H^i(\mathbf{M}_k^i, \breve{\mathbf{M}}_1)$, otherwise there would exist the possibility to fire an incorrect sequence. A transition $H^i(\mathbf{M}_k^i, \breve{\mathbf{M}}_1, \breve{t}_a)$ can safely fire only if all the entries of $\boldsymbol{\Omega}_{\mu_k}(i, E_k)$ are null.

Therefore, the first stage controller function F (Definition 3.4) is computed as

$$F(\mu_k, E_k) = \{H^i(\mathbf{M}_k^i, \breve{\mathbf{M}}_1, \breve{t}_a) | i \in E_k,\ \boldsymbol{\Omega}_{\mu_k}(i, E_k) = \mathbf{0}\}$$

5.4 Regulation algorithm

The first stage of the regulation algorithm is implemented by the forthcoming Algorithm 5.3, assuming $\mathbf{M}_0 \in \mu_0$, which drives the plant to a marking $\mathbf{M}_1$. Since the plant is safe and event-detectable then the plant's marking can be uniquely determined after any further firing. Thus, the second stage, which controls the plant for the subsequent specification firings, consists in a controller synthesized assuming complete information, as described in Section 4 but considering the initial markings of the plant and the specification as $\mathbf{M}_1$ and $\breve{\mathbf{M}}_1$.

Algorithm 5.3. Regulation based on partial information. Stage 1.

1: **Input** The plant and specification IPN models. The plant's output signal $\mathbf{y}$ and the detected firing sequence during the evolution.
2: **Output** Control action u during the evolution. Updated markings μ_k and the set of possible marking indexes E_k during the evolution.
3: *Synthesis:*
4: Let $\breve{t}_a$ be the only enabled transition at the specification initial marking $\breve{\mathbf{M}}_0$ whose firing leads to $\breve{\mathbf{M}}_1$.
5: Compute the set of potential plant's initial markings μ_0, by using the Algorithm 5.1.
6: Compute a plant's marking $\mathbf{M}_1$ that is reachable from any marking of μ_0 and that $\boldsymbol{\varphi}\mathbf{M}_1 = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_1$.
7: Compute the controllers $H^i(\mathbf{M}, \breve{\mathbf{M}}_1, \breve{t}_a)$ for the possible initial markings μ_0, by using the Algorithm 5.2.

Operation:
Initialize the plant marking and output index: $k = 0$. Thus, the set of possible initial states at the plant is $\mu_k = \mu_0$ and the current plant's output is $\mathbf{y}_k$. Initialize the set of possible marking indexes as $E_k = \{1, 2, ..., |\mu_0|\}$.
Wait until the firing of $\breve{t}_a$.
Update the specification marking as: $\breve{\mathbf{M}}_1 = \breve{\mathbf{M}}_0 + \breve{\mathbf{C}}(\bullet, a)$.
while the observed output $\mathbf{y}_k \neq \boldsymbol{\varphi}\breve{\mathbf{M}}_1$ **do**
  Compute $\boldsymbol{\Omega}_{\mu_k}$ and $F(\mu_k, E_k)$
  if $F(\mu_k, E_k) = \emptyset$ **then**
    STOP the algorithm
  else
    Select a symbol from $F(\mu_k, E_k)$, let i be the associated index to such symbol.
    Indicate the selected symbol
$$u = H^i(\mathbf{M}_k^i, \breve{\mathbf{M}}_1, \breve{t}_a)$$
    if a firing is not detected in the plant **then**
      Eliminate i from E_k, i.e., $E_k = E_k \setminus \{i\}$
    else
      Denote as t_j the fired transition
      Initialize $\mu_{k+1} = \emptyset$. Initialize $E_{k+1} = E_k$.
      for each $\mathbf{M}_k^s \in \mu_k$ **do**
        if t_j is not enabled at $\mathbf{M}_k^s$ **then**
          Eliminate s from E_{k+1}, i.e.,
$$E_{k+1} = E_{k+1} \setminus \{s\}$$
          Update the marking as: $\mathbf{M}_{k+1}^s = \mathbf{0}$
        else
          Update the marking:
          $\mathbf{M}_{k+1}^s = \mathbf{M}_k^s + \mathbf{C}(\bullet, j)$
        end if
        $\mu_{k+1} = \mu_{k+1} \cup \{\mathbf{M}_{k+1}^s\}$
      end for
      Update the index: $k = k + 1$
    end if
  end if
end while

Remark 1. The control scheme guarantees that $\mu_k[E_k]$ includes the true marking and the fired sequence σ_k is a prefix of the corresponding sequence σ_0^j, which leads the plant from the true initial marking $\mathbf{M}_0^j$ to $\mathbf{M}_1$. However, in some scenarios, the Algorithm 5.3 may block the plant (when $F(\mu_k, E_k) = \emptyset$), without reaching $\mathbf{M}_1$.

5.5 Example

Consider the IPN model presented in the fig. 2, with initial marking $\mathbf{M}_0 = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0]^T$ and the specification model of the fig. 3.

Let us apply the synthesis procedure of the Algorithm 5.3. First, at the initial marking the output vector $\mathbf{y} = [1\ 0\ 0\ 1]^T$ is detected (symbols A and D). Thus, the Algorithm 5.1 computes the possible initial markings

$$\mathbf{M}_0^1 = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0]^T$$
$$\mathbf{M}_0^2 = [0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0]^T$$

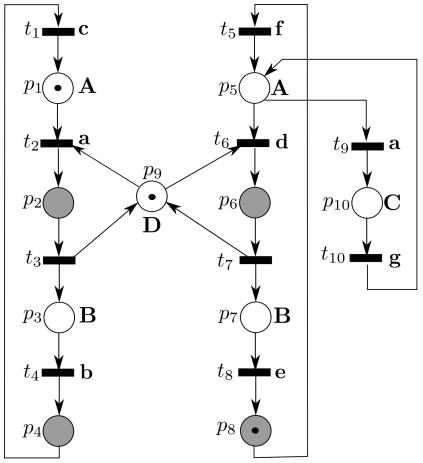

Fig. 2. PN plant, representing two identical processes described by places p_1, p_2, p_3, p_4 and p_5, p_6, p_7, p_8, respectively, and one shared resource, described by p_2, p_6, p_9. The second process can be down for repairing at the idle position p_5. For the production perspective, there is no distinction between the two processes, thus both have the same output symbols. Grey and white places represent nonmeasurable and measurable places, respectively.

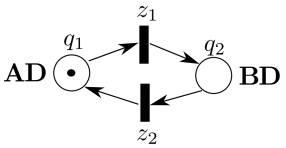

Fig. 3. PN specification. In a first state, a process is required to be idle (A) while the resource is idle (D) as well. In a second state, a process is required to finish (B) while the resource is idle.

Next, considering that z_1 is the only enabled transition at the specification, leading the system from $\breve{\mathbf{M}}_0$ to $\breve{\mathbf{M}}_1 = [0\ 1]^T$, a marking $\mathbf{M}_1$ is computed such that $\boldsymbol{\varphi}\mathbf{M}_1 = \breve{\boldsymbol{\varphi}}\breve{\mathbf{M}}_1$. In particular, consider

$$\mathbf{M}_1 = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0]^T$$

Next, a controller is designed for each possible initial marking in order to drive the system to $\mathbf{M}_1$. In particular, the following controllable sequences are obtained (all the transitions were intentionally involved):

$$\sigma_{z_1}^1 = t_2t_3t_4t_1t_2t_3t_5t_9t_{10}t_6t_7t_8$$
$$\sigma_{z_2}^2 = t_6t_7t_1t_2t_3t_4t_1t_2t_3t_8t_5t_9t_{10}t_6t_7t_8$$

The next step is to apply the Algorithm 4.1 to compute the controllers H^1 and H^2 that impose $\sigma_{z_1}^1$ and $\sigma_{z_1}^2$, respectively. In this way, the synthesis procedure is completed (the functions H^1 and H^2 are not written here due to lack of space).

Now, let us consider the operation procedure of the Algorithm 5.3. At the beginning, $\mu_0 = \{\mathbf{M}_0^1, \mathbf{M}_0^2\}$, $\breve{\mathbf{M}} = \breve{\mathbf{M}}_0$ and $E_0 = \{1, 2\}$. After the firing of z_1 at the specification, the marking specification becomes $\breve{\mathbf{M}}_1$, leading to the specification's output BD.

At μ_0, H^1 indicates $\lambda(t_2) = a$ and disables $\lambda(t_5) = f$, since t_5 is also enabled at $\mathbf{M}_0^1$. On the other hand, H^2 indicates $\lambda(t_6) = d$ and disables $\lambda(t_1) = c$ and $\lambda(t_9) = a$, since both t_1 and t_9 are enabled at $\mathbf{M}_0^2$. Thus, the common events matrix is

$$\mathbf{\Omega}_{\mu_0} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{1}$$

The element 1 in the matrix means that the event indicated by the controller H^1 is disabled by controller H^2 (the event a, related to t_2 and t_9). On the contrary, the event indicated by the controller H^2 is not disabled by the controller H^1, since the second row of the matrix is null. Therefore, $F(\mu_0, E_0) = \{2\}$. Consequently, the index 2 is selected and the symbol $H^2(\mathbf{M}_0^2, \breve{\mathbf{M}}_1, z_1) = \lambda(t_6) = d$ is indicated, i.e., the control action of the second controller is indicated. Nevertheless, since the true marking of the plant is $\mathbf{M}_0$, at which no transition with symbol d is enabled, then no transition fires. Thus, in accordance to the line 19 of the Algorithm 5.3, the index 2 is eliminated from E_0. Thus, $E_0 = \{1\}$ and $\mu_0[E_0] = \{\mathbf{M}_0^1\}$, which is actually the true plant's marking. Consequently, the common events matrix restricted to E_k will be [0] for further steps and thus the only active controller will be H^1, that will indicate the symbols to drive the plant from $\mathbf{M}_0^1$ to $\mathbf{M}_1$.

Now, let us consider again the operation procedure of the Algorithm 5.3, but in this case assuming that the true plant's initial marking is $\mathbf{M}_0 = \mathbf{M}_0^2$. As in the previous case, the common events matrix is (1) as well. Thus, $2 \in E_0$ is selected and the symbol $H^2(\mathbf{M}_0^2, \breve{\mathbf{M}}_1, z_1) = \lambda(t_6) = d$ is indicated. In this case, the transition t_6 fires at the plant, leading to the marking $\mathbf{M}' = [0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0]^T$. Since t_6 was not enabled from $\mathbf{M}_0^1$, then index 1 is removed from E_1. Thus, $E_1 = \{2\}$ and $\mu_1[E_1] = \{\mathbf{M}_1^2\}$, where $\mathbf{M}_1^2 = \mathbf{M}'$, i.e., the algorithm has found the true plant's marking. Consequently, the common events matrix restricted to E_k will be [0] for further steps and thus the only active controller will be H^2, which will indicate the symbols to drive the plant to $\mathbf{M}_1$.

6. CONCLUSIONS AND FUTURE WORK

In this work, the regulation control problem for PNs under partial information has been addressed. For this, the control problem is split in two stages: in the first stage, the plant is driven to a known marking $\mathbf{M}_1$, the second stage is reduced to the full-information regulation case. The first stage control involves three steps: first, the set of possible initial markings is computed, second, a controller is synthesized for each possible initial marking, third, an algorithm is provided to select one control action from one controller and to discard possible initial markings based on the input/output observations. This scheme prevents the plant from firing unplanned sequences, however, the controller may block the plant.

It is left as future work to provide recovery mechanisms when the controller blocks the plant. Moreover, the case in which the plant initial output does not agree with the specification output will be considered.

REFERENCES

Basile, F., Cordone, R., and Piroddi, L. (2013). Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models. *Automatica*, 49(11), 3432–3439.

Chen, Y., Li, Z., Khalgui, M., and Mosbahi, O. (2011). Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems. *IEEE Transactions on Automation Science and Engineering*, 8(2), 374–393.

David, R. and Alla, H. (2010). *Discrete, continuous, and hybrid Petri nets*. Springer.

Giua, A., DiCesare, F., and Silva, M. (1992). Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *IEEE Int. Conf. on Systems, Man Cybernetics*, 974 – 979.

Giua, A. and Silva, M. (2018). Petri nets and automatic control: A historical perspective. *Annual Reviews in Control*, 45, 223–239.

Guevara-Lozano, D., Vázquez, C.R., Antonio, and Ramírez-Treviño, A. (2019). Towards decentralized tracking control for Petri nets. In *In Emerging Technologies and Factory Automation*.

Li, Z., Wu, N., and Zhou, M. (2012). Deadlock control for automated manufacturing systems based on Petri nets. *IEEE Transactions on Systems, Man and Cybernetics Part C:Applications and Reviews*, 42(4), –.

Moody, J. and Antsaklis, P. (1998). *Supervisory control of Discrete Event Systems using Petri nets*. Kluwer Academic Publishers.

Ramadge, P. and Wonham, W. (1982). Supervision of discrete event processes. In *In IEEE International Conference on Decisions and Control*, 1228–1229.

Ramadge, P. and Wonham, W. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1), 206–230.

Ramírez-Trevino, A., Rivera-Rangle, I., and López-Mellado, E. (2003). Observability od discrete event systems modeled by interpreted Petri nets. *IEEE Transactions on Robotics and Automation*, 19(4), 557–565.

Sánchez-Blanco, J., Ramírez-Treviño, A., and Santoyo, A. (2004). Regulation control in interpreted Petri nets using trace equivalence. In *In IEEE International Conference on Systems, Man and Cybernetics*, 1843–1848.

Santoyo, A., Jiménez-Ochoa, I., and Ramírez-Trevino, A. (2001). A complete cycle for controller design in discrete event systems. In *In IEEE International Conference on Systems, Man and Cybernetics*, 2688–2693.

Silva, M. (1993). *Introducing Petri nets. In Practice of Petri nets in manufacturing*. Chapman & Hall.

Vázquez, C.R., Gómez-Castellanos, J.A., and Ramírez-Treviño, A. (2018). Tracking control of electropneumatic systems based on Petri nets. In *International Conference on Informatics in Control, Automation and Robotics*.

Wonham, W., Cai, K., and Rudie, K. (2018). Supervisory control of discrete-event systems: A brief history. *Annual Reviews in Control*, 45, 250–256.